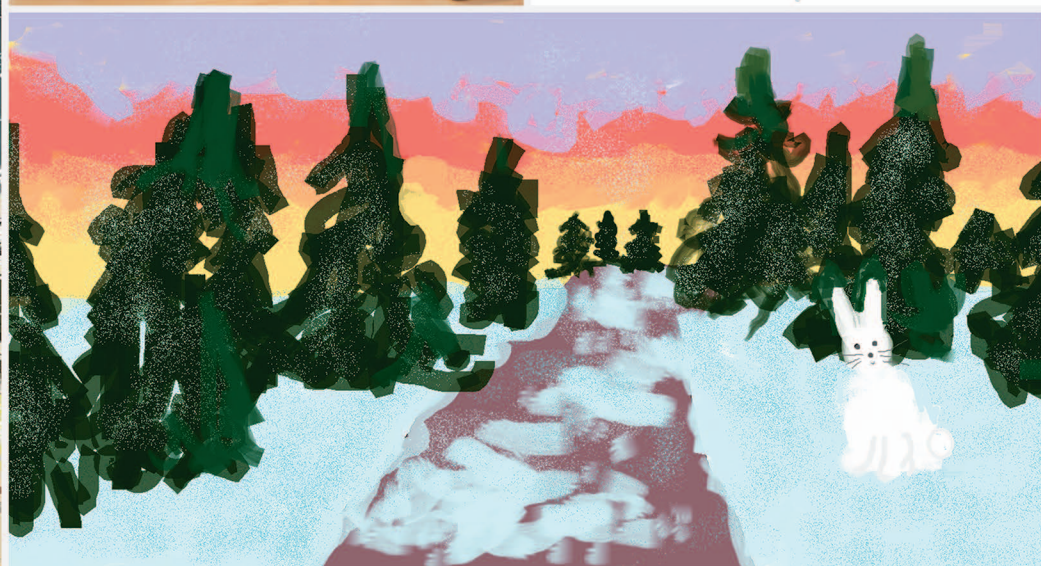
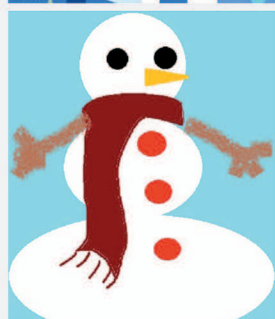
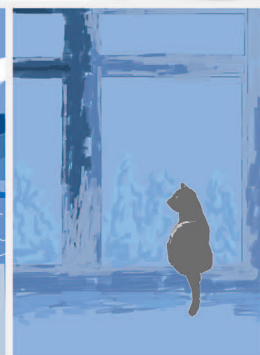
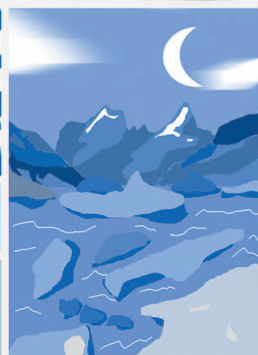


ИНФОРМАТИКА В ШКОЛЕ

№ 1'2025

ISSN 2221-1993

www.infojournal.ru





1С:ПЛАНОВОЕ ПИТАНИЕ



ДИЕТОЛОГ

Бракераж
Составление меню
Корректировка меню
Накопительная ведомость
Разработка рациона питания



КЛАДОВЩИК

Учет прихода-расхода продуктов
Остатки продуктов
Партионный учет
Учет сроков хранения
Расчет заказа продуктов



БУХГАЛТЕР

Учет продуктов питания
Расчет фактической стоимости питания
Ведение разделенного движения
продуктов по источникам
финансирования



ЗАВЕДУЮЩИЙ СТОЛОВОЙ

Бракераж готовых блюд
Акты проработки норм отхода
при холодной обработке
Картотека блюд с нормами
закладки продуктов



Фирма «1С»
123056, г. Москва, а/я 64, ул. Селезневская, д. 21
Тел.: (495) 737-92-57
Факс: (495) 681-44-07
cko@1c.ru, solutions.1c.ru/plan-meal



ООО «Агентство Капитан»
125481, г. Москва, ул. Свободы, д. 71/2, оф. 387
Тел.: (495) 782-25-57, факс: (499) 497-72-21
1cap@1cap.ru, 1cp@1cp.ru (техподдержка)
1cap.ru, 1cp.ru

ИНФОРМАТИКА В ШКОЛЕ

ИЗДАЕТСЯ С СЕНТЯБРЯ 2002 ГОДА

Том 24 № 1 февраль 2025

Содержание

КОНКУРС ИНФО-2024

Итоги XXI Всероссийского конкурса научно-практических работ ИНФО-2024	6
Корнилов П. А., Федотова Н. П. Задачи с математическим содержанием на олимпиадах и конкурсах по информатике	10
Шамсутдинова Т. М. Возможности табличных процессоров по развитию аналитической компетенции обучающихся (на примере олимпиадных задач)	18
Климина Н. В. Воксельное моделирование на базе среды моделирования MagicaVoxel.....	26
Громова С. Ф., Скоробогатов Н. С. «Вокруг света за 34 часа». Виртуальный геокешинг как средство формирования метапредметных компетенций обучающихся	39

МЕТОДИЧЕСКАЯ КОПИЛКА

Еремин Е. А. Многослойная модель компьютера как ориентир при отборе материала для курса информатики.....	53
Чундерова К. А., Васева Е. С. Применение интерактивных методов обучения для развития логического мышления школьников при обучении теме «Элементы алгебры логики»	64

ИТОГОВАЯ АТТЕСТАЦИЯ ПО ИНФОРМАТИКЕ

Поляков К. Ю. Кластеризация в задачах ЕГЭ по информатике.....	69
Шамаева Т. В. Использование электронных таблиц в решении заданий КЕГЭ по информатике	77

КОНКУРСЫ

Итоги XXVIII Всероссийского конкурса цифровых изображений и фотографий журнала «Информатика в школе» ФОТО 1-2025	85
--	----

Свидетельство о регистрации
средства массовой информации
ПИ № 77-12068 от 11 марта 2002 г.

Издатель ООО «Образование и Информатика»
119261, Россия, г. Москва, Ленинский пр-т, д. 82/2, комн. 6
Телефон: +7 (495) 140-19-86
E-mail: readinfo@infojournal.ru
Сайт издательства: <http://infojournal.ru/>
Почтовый адрес: 119270, Россия, г. Москва, а/я 15

Подписано в печать 28.02.25.
Формат 60×90¹/₈. Усл. печ. л. 11,0
Тираж 2000 экз. Заказ № 7691.
Отпечатано в типографии «ЛАЙДЕР ПРИНТ»
142104, Россия, Московская область, г. Подольск,
ул. Свердлова, д. 26, тел.: +7 (495) 212-91-99,
+7 (926) 204-49-31, e-mail: info@book-expert.ru

© «Образование и Информатика», 2025



ISSN 2221-1993

Журнал входит в Перечень российских рецензируемых научных изданий ВАК, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученых степеней доктора и кандидата наук

Контакты

Главный редактор
dr.l@infojournal.ru

Редакция
readinfo@infojournal.ru

Отдел распространения
info@infojournal.ru

Телефон:
+7 (495) 140-19-86

Почтовый адрес
119270, Россия, г. Москва,
а/я 15

Сайт журнала
<http://school.infojournal.ru>

ОБЪЕДИНЕННАЯ РЕДАКЦИЯ ИНФО

**Главный редактор журнала
«Информатика и образование»**
ГРИГОРЬЕВ Сергей Георгиевич

**Главный редактор журнала
«Информатика в школе»**
ЛЕОНОВ Александр Георгиевич

Директор издательства
РЫБАКОВ Даниил Сергеевич

Научный редактор
ДЕРГАЧЕВА Лариса Михайловна

Ведущий редактор
КИРИЧЕНКО Ирина Борисовна

Корректор
ШАРАПКОВА Людмила Михайловна

Верстка
ФЕДОТОВ Дмитрий Викторович

Дизайн
ГЛАВНИЦКИЙ Евгений Николаевич

Отдел распространения и рекламы
КУЗНЕЦОВА Елена Александровна

НАУЧНО-ПРАКТИЧЕСКИЙ ЖУРНАЛ «ИНФОРМАТИКА В ШКОЛЕ»

УЧРЕДИТЕЛЬ:

ИЗДАТЕЛЬСТВО
«ОБРАЗОВАНИЕ И ИНФОРМАТИКА»

ГЛАВНЫЙ РЕДАКТОР

ЛЕОНОВ Александр Георгиевич

доктор пед. наук, канд. физ.-мат. наук, доцент, ведущий научный сотрудник лаборатории вычислительных методов, механико-математический факультет, Московский государственный университет имени М. В. Ломоносова; зав. кафедрой дополнительного профессионального образования, Федеральный научный центр Научно-исследовательский институт системных исследований Национального исследовательского центра «Курчатовский институт» (г. Москва, Россия)

РЕДАКЦИОННАЯ КОЛЛЕГИЯ

БАРАКИНА Татьяна Вячеславовна

канд. пед. наук, доцент, доцент кафедры предметных технологий начального и дошкольного образования, факультет начального, дошкольного и специального образования, Омский государственный педагогический университет (г. Омск, Россия)

БЕШЕНКОВ Сергей Александрович

доктор пед. наук, профессор, профессор кафедры теории и методики спорта и физического воспитания, Институт спортивных технологий и физического воспитания, Московский государственный университет спорта и туризма (г. Москва, Россия)

БОСОВА Анна Юрьевна

заместитель главного редактора журнала «Информатика в школе», ст. преподаватель кафедры математики и информатики в начальной школе, факультет начального образования, Институт детства, Московский педагогический государственный университет (г. Москва, Россия)

ВОЕВОДИН Владимир Валентинович

чл.-корр. РАН, доктор физ.-мат. наук, профессор, лауреат премии Правительства РФ в области образования, директор Научно-исследовательского вычислительного центра, Московский государственный университет имени М. В. Ломоносова (г. Москва, Россия)

ДЕРГАЧЕВА Лариса Михайловна

канд. пед. наук, доцент, научный редактор издательства «Образование и Информатика» (г. Москва, Россия)

ЗАВОДЧИКОВА Надежда Ивановна

канд. пед. наук, доцент, доцент кафедры теории и методики обучения информатике, физико-математический факультет, Ярославский государственный педагогический университет им. К. Д. Ушинского (г. Ярославль, Россия)

ЗАСЛАВСКАЯ Ольга Юрьевна

доктор пед. наук, профессор, профессор департамента информатизации образования, Институт цифрового образования, Московский городской педагогический университет (г. Москва, Россия)

ЗУБРИЛИН Андрей Анатольевич

канд. филос. наук, доцент, доцент кафедры физики, информационных технологий и методик обучения, физико-математический факультет, Мордовский государственный педагогический университет имени М. Е. Евсевьева (г. Саранск, Россия)

КИРИЧЕНКО Ирина Борисовна

ведущий редактор издательства «Образование и Информатика» (г. Москва, Россия)

ЛЕВЧЕНКО Ирина Витальевна

доктор пед. наук, профессор, профессор кафедры теории и методики обучения математике и информатике, Институт математики и информатики, Московский педагогический государственный университет (г. Москва, Россия)

ПОЛЯКОВ Константин Юрьевич

доктор тех. наук, профессор кафедры судовой автоматики и измерений, факультет корабельной энергетики и автоматики, Санкт-Петербургский государственный морской технический университет (Санкт-Петербург, Россия)

РЫБАКОВ Даниил Сергеевич

канд. пед. наук, доцент, директор издательства «Образование и Информатика» (г. Москва, Россия)

ТРОИЦКАЯ Ольга Николаевна

канд. пед. наук, доцент, доцент кафедры экспериментальной математики и информатизации образования, Высшая школа информационных технологий и автоматизированных систем, Северный (Арктический) федеральный университет имени М. В. Ломоносова (г. Архангельск, Россия)

ФЕДОРОВА Галина Аркадьевна

доктор пед. наук, профессор кафедры информатики и методики обучения информатике, факультет математики, информатики, физики и технологии, Омский государственный педагогический университет (г. Омск, Россия)

ФЕДОСОВ Александр Юрьевич

доктор пед. наук, доцент, профессор кафедры информационных технологий, искусственного интеллекта и общественно-социальных технологий цифрового общества, факультет политических и социальных технологий, Российский государственный социальный университет (г. Москва, Россия)

ЧЕРНЕЦКАЯ Татьяна Александровна

канд. пед. наук, руководитель методического направления, отдел образовательных программ, ООО «1С-Публишинг» (г. Москва, Россия)

INFORMATICS IN SCHOOL

PUBLISHED SINCE AUGUST 1986 ИЗДАЕТСЯ С СЕПТЕМБЕР 2002

Volume 24 No 1 February 2025

Table of Contents

INFO-2024 CONTEST

The results of the 21th All-Russian contest of scientific and practical works INFO-2024	6
P. A. Kornilov, N. P. Fedotova. Tasks with mathematical content in Olympiads and competitions in informatics	10
T. M. Shamsutdinova. Capabilities of spreadsheet processors for developing analytical competency of students (on the example of Olympiad tasks)	18
N. V. Klimina. Voxel modeling based on the MagicaVoxel modeling environment	26
S. F. Gromova, N. S. Skorobogatov. "Around the world in 34 hours". Virtual geocaching as a means of forming students' meta-subject competencies	39

TEACHING PRACTICES

E. A. Eremin. Multilayer model of computer as a reference point during selecting material for the informatics course	53
K. A. Chunderova, E. S. Vaseva. Using interactive teaching methods for the development of logical thinking of schoolchildren when teaching the theme "Elements of algebra of logic"	64

FINAL CERTIFICATION IN INFORMATICS

K. Yu. Polyakov. Clustering in tasks of the Unified State Exam in informatics	69
T. V. Shamaeva. The use of spreadsheets in solving tasks from the computer based Unified State Exam in informatics	77

CONTESTS

The results of the 28th All-Russian contest of digital images and photographs of the "Informatics in School" journal PHOTO 1-2025	85
---	----

Certificate of Registration
of Mass Media PI No 77-12068
dated March 11, 2002

Publisher: LLC "Education and Informatics"
119261, Russia, Moscow, Leninsky prospect, 82/2, room 6
Phone: +7 (495) 140-19-86
E-mail: readinfo@infojournal.ru
Publisher's website: <https://infojournal.ru/>
Postal address: 119270, Russia, Moscow, PO Box 15

Signed for printing: 28.02.25.
Format 60×90¹/₈. Cond. printed sheets 11,0
Circulation 2000 copies. Order No 7691.
Printed at the printing office of "LIDER PRINT"
142104, Russia, Moscow region, Podolsk,
st. Sverdlova, 26, tel.: +7 (495) 212-91-99,
+7 (926) 204-49-31, e-mail: info@book-expert.ru

© Education and Informatics, 2025



ISSN 2221-1993

The journal is included in the List of Russian peer-reviewed scientific publications of the Higher Attestation Commission, in which the main scientific results of dissertations should be published for the degrees of Doctor of Sciences and Candidate of Sciences

Contacts

Editor-in-chief
dr.l@infojournal.ru

Editorial team
readinfo@infojournal.ru

Distribution
and Advertising Department
info@infojournal.ru

Phone
+7 (495) 140-19-86

Postal address
119270, Russia, Moscow,
PO Box 15

Journal website
<http://school.infojournal.ru>

EDITORIAL TEAM

**Editor-in-Chief of the
"Informatics and Education" journal**

Sergey G. GRIGORIEV

**Editor-in-Chief of the
"Informatics in School" journal**

Aleksandr G. LEONOV

Director of Publishing House

Daniil S. RYBAKOV

Science Editor

Larisa M. DERGACHEVA

Senior Editor

Irina B. KIRICHENKO

Proofreader

Lyudmila M. SHARAPKOVA

Layout

Dmitry V. FEDOTOV

Design

Eugene N. GLAVNICKY

**Distribution and Advertising
Department**

Elena A. KUZNETSOVA

SCHOLARLY JOURNAL «INFORMATICS IN SCHOOL»

FOUNDER:

PUBLISHING HOUSE
«EDUCATION AND INFORMATICS»

EDITOR-IN-CHIEF

Aleksandr G. LEONOV

Dr. Sci. (Edu.), Cand. Sci. (Phys.-Math.), Docent, Leading Researcher at the Laboratory of Computational Methods, Faculty of Mechanics and Mathematics, Lomonosov Moscow State University;
Head of the Department of Additional Professional Education, Scientific Research Institute for Systems Analysis of the National Research Centre "Kurchatov Institute" (Moscow, Russia)

EDITORIAL BOARD

Tatyana V. BARAKINA

Cand. Sci. (Edu.), Docent, Associate Professor at the Department of Subject Technologies of Primary and Preschool Education, Faculty of Primary, Preschool and Special Education, Omsk State Pedagogical University (Omsk, Russia)

Sergey A. BESHENKOV

Dr. Sci. (Edu.), Professor, Professor at the Department of Theory and Methodology of Sports and Physical Education, Institute of Sports Technologies and Physical Education, Moscow State University of Sport and Tourism (Moscow, Russia)

Anna Yu. BOSOVA

Deputy Editor-in-Chief of the "Informatics in School" Journal, Senior Lecturer at the Department of Mathematics and Informatics in Primary School, Faculty of Elementary Education, Institute of Childhood, Moscow Pedagogical State University (Moscow, Russia)

Vladimir V. VOEVODIN

Corresponding Member of RAS, Dr. Sci. (Phys.-Math.), Professor, Laureate of the Russian Federation Government Prize in Education, Director of Research Computing Center, Lomonosov Moscow State University (Moscow, Russia)

Larisa M. DERGACHEVA

Cand. Sci. (Edu.), Docent, Science Editor of the "Education and Informatics" Publishing House (Moscow, Russia)

Nadejda I. ZAVODCHIKOVA

Cand. Sci. (Edu.), Docent, Associate Professor at the Department of Theory and Methods of Teaching Informatics, Faculty of Physics and Mathematics, Yaroslavl State Pedagogical University named after K. D. Ushinsky (Yaroslavl, Russia)

Olga Yu. ZASLAVSKAYA

Dr. Sci. (Edu.), Professor, Professor at the Department of Education Informatization, Institute of Digital Education, Moscow City University (Moscow, Russia)

Andrej A. ZUBRILIN

Cand. Sci. (Philos.), Docent, Associate Professor at the Department of Physics, Information Technologies and Teaching Methods, Faculty of Physics and Mathematics, Mordovian State Pedagogical University named after M. E. Evseev (Saransk, Russia)

Irina B. KIRICHENKO

Senior Editor of the "Education and Informatics" Publishing House (Moscow, Russia)

Irina V. LEVCHENKO

Dr. Sci. (Edu.), Professor, Professor at the Department of Theory and Methods of Teaching Mathematics and Informatics, Institute of Mathematics and Informatics, Moscow Pedagogical State University (Moscow, Russia)

Konstantin Yu. POLYAKOV

Dr. Sci. (Eng.), Professor at the Department of Ship Automation and Measurements, Faculty of Ship Power Engineering and Automation, State Marine Technical University (Saint Petersburg, Russia)

Daniil S. RYBAKOV

Cand. Sci. (Edu.), Docent, Director of the "Education and Informatics" Publishing House (Moscow, Russia)

Olga N. TROITSKAYA

Cand. Sci. (Edu.), Docent, Associate Professor at the Department of Experimental Mathematics and Informatization of Education, Higher School of Information Technologies and Automated Systems, Northern (Arctic) Federal University named after M. V. Lomonosov (Arkhangelsk, Russia)

Galina A. FEDOROVA

Dr. Sci. (Edu.), Docent, Professor at the Department of Informatics and Methods of Teaching Informatics, Faculty of Mathematics, Informatics, Physics and Technology, Omsk State Pedagogical University (Omsk, Russia)

Aleksandr Yu. FEDOSOV

Dr. Sci. (Edu.), Docent, Professor at the Department of Information Technologies, Artificial Intelligence and Public and Social Technologies of the Digital Society, Faculty of Political and Social Technologies, Russian State Social University (Moscow, Russia)

Tatyana A. CHERNETSKAYA

Cand. Sci. (Edu.), Head of Methodological Direction, Department of Educational Programs, 1C-Publishing LLC (Moscow, Russia)

ВСЕРОССИЙСКИЙ КОНКУРС «РОБОТОТЕХНИКА В ИНФОРМАТИКЕ»

Уважаемые коллеги!

Издательство «Образование и Информатика», редакция журнала «Информатика в школе»
объявляют о проведении
Всероссийского конкурса «Робототехника в информатике»

Конкурс проводится по двум номинациям:

- 1. Номинация для педагогов: Изучение алгоритмизации и программирования на основе робототехники.**
- 2. Номинация для учащихся: Программируем роботов — изучаем информатику.**

Оргкомитет конкурса

Руководит конкурсом Организационный комитет (Оргкомитет), состоящий из представителей Российской академии образования, ведущих методистов, членов редакционных коллегий журналов «Информатика и образование» и «Информатика в школе», сотрудников объединенной редакции журналов.

Цели и задачи конкурса

1. Выявление и поддержка талантливых педагогов.
2. Включение педагогов в деятельность по разработке нового содержания образования, новых образовательных технологий и методик обучения.
3. Выявление и распространение новых образовательных технологий, способствующих развитию интереса школьников к информатике и информационным технологиям.
4. Создание информационно-образовательного пространства на страницах журнала «Информатика в школе» по обмену и распространению опыта преподавания информатики, а также использования в образовательном процессе информационно-коммуникационных технологий.
5. Развитие интереса школьников к информатике и другим учебным дисциплинам, к информационным технологиям.
6. Творческое развитие школьников, повышение их социальной активности, создание условий для самореализации.
7. Повышение информационной культуры и информационно-коммуникационной компетентности всех участников образовательного процесса — учащихся, педагогов, родителей.

Работы на конкурс принимаются с 1 апреля по 31 мая 2025 года включительно. Работы, присланные позже 31 мая 2025 года, к участию в конкурсе допускаться не будут. Подача работ производится только через заполнение формы заявки на сайте ИНФО (необходима предварительная регистрация на сайте или авторизация для зарегистрированных пользователей).

Итоги конкурса будут подведены в № 3-2025 (июнь) журнала «Информатика в школе», а также опубликованы на сайте издательства «Образование и Информатика».

Лучшие работы будут опубликованы в журнале «Информатика в школе».

Победители конкурса получают:

- диплом от издательства «Образование и Информатика»;
- печатный экземпляр журнала «Информатика в школе» № 3-2025 (июнь), в котором будут опубликованы итоги конкурса;
- авторский печатный экземпляр журнала «Информатика в школе» с опубликованной работой.

**Подробную информацию
о требованиях к оформлению конкурсной работы и конкурсной заявки,
а также всю дополнительную информацию о конкурсе
вы можете найти на сайте издательства «Образование и Информатика»:
<http://infojournal.ru/competition/robot-2025/>
а также получить в редакции ИНФО
по адресу: readinfo@infojournal.ru
и по телефону: (495) 140-19-86**



ИТОГИ XXI ВСЕРОССИЙСКОГО КОНКУРСА НАУЧНО-ПРАКТИЧЕСКИХ РАБОТ ИНФО-2024

Уважаемые коллеги!

В октябре 2024 года издательство «Образование и Информатика» объявило конкурс научно-практических работ по методике обучения информатике и информатизации образования ИНФО-2024, посвященный 40-летию школьной информатики.

Было организовано жюри конкурса, в которое вошли представители Российской академии образования, ведущие методисты, члены редакционных коллегий журналов «Информатика и образование» и «Информатика в школе», сотрудники объединенной редакции ИНФО.

В конкурсе приняли участие как работники образования — учителя, преподаватели вузов, работники учреждений дошкольного образования, педагоги системы дополнительного образования, методисты, — так и студенты педвузов из разных регионов Российской Федерации, а также из стран ближнего зарубежья.

Представляем лауреатов (1-е место) и дипломантов (2-е место) конкурса ИНФО-2024.

Все победители представлены в алфавитном порядке.

НОМИНАЦИЯ 1

«40 ЛЕТ ШКОЛЬНОЙ ИНФОРМАТИКЕ: СОВРЕМЕННЫЙ УРОК ИНФОРМАТИКИ»

По мнению жюри, в данной номинации не было представлено работ, достойных 1-го места.

Дипломант конкурса (2-е место)

Букина Елена Юрьевна,

учитель информатики, средняя общеобразовательная школа № 1, г. Зеленокумск, Ставропольский край, Россия

НОМИНАЦИЯ 2

«40 ЛЕТ ШКОЛЬНОЙ ИНФОРМАТИКЕ: ОЛИМПИАДЫ, КОНКУРСЫ, ПРОЕКТЫ ПО ИНФОРМАТИКЕ»

Лауреаты конкурса (1-е место)

Корнилов Петр Анатольевич,

доцент кафедры теории и методики обучения информатике, физико-математический факультет, Ярославский государственный педагогический университет им. К. Д. Ушинского, г. Ярославль, Россия

Федотова Наталья Петровна,

доцент кафедры компьютерной безопасности и математических методов обработки информации, математический факультет, Ярославский государственный университет им. П. Г. Демидова, г. Ярославль, Россия

Шамсутдинова Татьяна Михайловна,

доцент кафедры цифровых технологий и прикладной информатики, экономический факультет, Башкирский государственный аграрный университет, г. Уфа, Республика Башкортостан, Россия

Дипломанты конкурса (2-е место)

Лаптева Татьяна Дмитриевна,

преподаватель кафедры высшей математики и методики обучения математике, математический факультет, Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия

Латышева Любовь Павловна,

доцент кафедры высшей математики и методики обучения математике, математический факультет, Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия

Мусихина Ирина Васильевна,

старший преподаватель кафедры высшей математики и методики обучения математике, математический факультет, Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия

Олехов Алексей Андреевич,

старший преподаватель кафедры высшей математики и методики обучения математике, математический факультет, Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия

Скорнякова Анна Юрьевна,

декан математического факультета, доцент кафедры высшей математики и методики обучения математике, Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия

Черемных Елена Леонидовна,

заведующий кафедрой высшей математики и методики обучения математике, математический факультет, Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия

НОМИНАЦИЯ 3**«40 ЛЕТ ШКОЛЬНОЙ ИНФОРМАТИКЕ: ДОПОЛНИТЕЛЬНОЕ ОБРАЗОВАНИЕ
В ОБЛАСТИ ИНФОРМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»****Лауреат конкурса (1-е место)**

Климина Наталья Владимировна,

учитель информатики, средняя общеобразовательная школа № 6 имени Героя Советского Союза В. Н. Банцеккина, г.о. Сызрань, Самарская область, Россия

Дипломанты конкурса (2-е место)

Семёнова Ольга Геннадьевна,

старший преподаватель кафедры теории и методики обучения информатике, физико-математический факультет, Ярославский государственный педагогический университет им. К. Д. Ушинского, г. Ярославль, Россия

Сергеева Анастасия Олеговна,

студентка 5-го курса кафедры теории и методики обучения информатике, физико-математический факультет, Ярославский государственный педагогический университет им. К. Д. Ушинского, г. Ярославль, Россия

НОМИНАЦИЯ 4**«40 ЛЕТ ШКОЛЬНОЙ ИНФОРМАТИКЕ: ИННОВАЦИИ В МЕТОДАХ И СРЕДСТВАХ ОБУЧЕНИЯ»****Лауреаты конкурса (1-е место)**

Громова Светлана Фаутовна,

доцент кафедры высшей математики и информатики, социально-педагогический факультет, Сургутский государственный педагогический университет, г. Сургут, Ханты-Мансийский автономный округ — Югра, Тюменская область, Россия

Скоробогатов Никита Сергеевич,

студент 1-го курса магистратуры, кафедра высшей математики и информатики, социально-педагогический факультет, Сургутский государственный педагогический университет, г. Сургут, Ханты-Мансийский автономный округ — Югра, Тюменская область, Россия

Дипломанты конкурса (2-е место)

Рибко Ольга Васильевна,

аспирант кафедры прикладной математики и информатики, физико-математический факультет, Институт цифровых технологий, Марийский государственный университет, г. Йошкар-Ола, Республика Марий Эл, Россия

Токтарова Вера Ивановна,

первый проректор — проректор по стратегическому развитию, профессор кафедры прикладной математики и информатики, физико-математический факультет, Институт цифровых технологий, Марийский государственный университет, г. Йошкар-Ола, Республика Марий Эл, Россия

Францкевич Александр Александрович,

декан физико-математического факультета, Белорусский государственный педагогический университет имени Максима Танка, г. Минск, Беларусь

Шалесная Анастасия Сергеевна,

аспирант кафедры информатики и методики преподавания информатики, физико-математический факультет, Белорусский государственный педагогический университет имени Максима Танка, г. Минск, Беларусь

НОМИНАЦИЯ 5

«40 ЛЕТ ШКОЛЬНОЙ ИНФОРМАТИКЕ: ПОДГОТОВКА ПЕДАГОГИЧЕСКИХ КАДРОВ ДЛЯ СОВРЕМЕННОЙ ЦИФРОВОЙ ШКОЛЫ»

Лауреаты конкурса (1-е место)

Бычкова Дарья Дмитриевна,

доцент кафедры вычислительной математики и информационных технологий, физико-математический факультет, Государственный университет просвещения, г. Москва, Россия

Проданец Алина Васильевна,

магистрант направления подготовки 44.04.01 «Педагогическое образование», физико-математический факультет, Государственный университет просвещения, г. Москва, Россия

Дипломант конкурса (2-е место)

Иманова Ольга Анатольевна,

доцент кафедры информационных технологий обучения и непрерывного образования, Институт педагогики, психологии и социологии, Сибирский федеральный университет, г. Красноярск, Россия

НОМИНАЦИЯ 6

«40 ЛЕТ ШКОЛЬНОЙ ИНФОРМАТИКЕ: ИНФОРМАТИЗАЦИЯ УПРАВЛЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ОРГАНИЗАЦИЕЙ»

По мнению жюри, в данной номинации не было представлено работ, достойных 1-го места.

Дипломант конкурса (2-е место)

Васева Елена Сергеевна,

доцент кафедры информационных технологий и физико-математического образования, факультет естествознания, математики и информатики, Нижнетагильский государственный социально-педагогический институт (филиал) Российского государственного профессионально-педагогического университета, г. Нижний Тагил, Свердловская область, Россия

Все представленные выше лауреаты (1-е место) и дипломанты (2-е место) конкурса ИНФО-2024 будут награждены дипломами соответствующего достоинства от издательства «Образование и Информатика». Их работы будут опубликованы в журналах «Информатика и образование» и «Информатика в школе» (в соответствии с рекомендациями членов жюри конкурса в зависимости от тематики и содержания работы).

В качестве приза победителя конкурса ИНФО-2024 получают подписку в печатном виде на 2025 год на журналы «Информатика и образование» и «Информатика в школе».

Также по результатам конкурса отмечены жюри и рекомендованы к публикации в журналах «Информатика и образование» и «Информатика в школе» работы следующих авторов:

Баракина Татьяна Вячеславовна,

доцент кафедры предметных технологий начального и дошкольного образования, факультет начального, дошкольного и специального образования, Омский государственный педагогический университет, г. Омск, Россия

Быкова Ирина Альбертовна,

старший преподаватель кафедры теории и методики обучения информатике, физико-математический факультет, Ярославский государственный педагогический университет им. К. Д. Ушинского, г. Ярославль, Россия

Житкова Полина Александровна,

студентка 5-го курса кафедры теории и методики обучения информатике, физико-математический факультет, Ярославский государственный педагогический университет им. К. Д. Ушинского, г. Ярославль, Россия

Воронов Роман Борисович,

учитель информатики, Удельнинская гимназия, п. Удельная, г.о. Раменский, Московская область;
методист, Методический центр «Раменский дом учителя», г.о. Раменский, Московская область, Россия

Гончарова Татьяна Семеновна,

педагог дополнительного образования, Донецкий Республиканский Дворец детского и юношеского творчества,
г. Донецк, Донецкая Народная Республика, Россия

Зубрилин Андрей Анатольевич,

доцент кафедры физики, информационных технологий и методик обучения, физико-математический
факультет, Мордовский государственный педагогический университет им. М. Е. Евсевьева, г. Саранск,
Республика Мордовия, Россия

Салий Надежда Сергеевна,

студентка 4-го курса, физико-математический факультет, Мордовский государственный педагогический
университет им. М. Е. Евсевьева, г. Саранск, Республика Мордовия, Россия

Иванова Ангелина Валерьевна,

доцент кафедры высшей математики и информатики, социально-педагогический факультет,
Сургутский государственный педагогический университет, г. Сургут, Ханты-Мансийский автономный
округ — Югра, Тюменская область, Россия

Хакимуллина Диана Рифгатовна,

студентка 5-го курса кафедры высшей математики и информатики, социально-педагогический факультет,
Сургутский государственный педагогический университет, г. Сургут, Ханты-Мансийский автономный
округ — Югра, Тюменская область, Россия

Курганова Наталья Александровна,

доцент кафедры прикладной информатики и математики, факультет математики, информатики, физики
и технологии, Омский государственный педагогический университет, г. Омск, Россия

Латышева Любовь Павловна,

доцент кафедры высшей математики и методики обучения математике, математический факультет,
Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия

Скорнякова Анна Юрьевна,

декан математического факультета, доцент кафедры высшей математики и методики обучения
математике, Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия

Смирнов Евгений Иванович,

профессор кафедры высшей математики и методики обучения математике, математический факультет,
Пермский государственный гуманитарно-педагогический университет, г. Пермь;
заведующий кафедрой математического анализа, теории и методики обучения математике,
физико-математический факультет, Ярославский государственный педагогический университет
им. К. Д. Ушинского, г. Ярославль, Россия

Черемных Елена Леонидовна,

заведующий кафедрой высшей математики и методики обучения математике, математический факультет,
Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия

Лобанов Алексей Александрович,

учитель информатики, директор, ЧОУ «Школа «Таурас», Санкт-Петербург, Россия

Лобанова Татьяна Юрьевна,

учитель информатики, Ангарский лицей № 1, г. Ангарск, Иркутская область, Россия

Плаксин Михаил Александрович,

доцент кафедры информационных технологий в бизнесе, факультет социально-экономических
и компьютерных наук, Пермский филиал Национального исследовательского университета
«Высшая школа экономики», г. Пермь;

доцент кафедры математического обеспечения вычислительных систем, Пермский государственный
национальный исследовательский университет, г. Пермь, Россия

Участники конкурса, чьи работы рекомендованы к публикации, получают сертификат об участии в конкурсе и публикации вместе с авторским экземпляром журнала, в котором будет опубликована работа.

Остальные конкурсанты могут получить сертификат об участии, который будет подготовлен по индивидуальному запросу.

Следите за информацией о новых конкурсах в журналах «Информатика и образование» и «Информатика в школе», а также на сайте ИНФО: <https://infojournal.ru/>

DOI: 10.32517/2221-1993-2025-24-1-10-17

**П. А. Корнилов***победитель конкурса ИНФО-2024 в номинации**«40 лет школьной информатике: Олимпиады, конкурсы, проекты по информатике»,**Ярославский государственный педагогический университет им. К. Д. Ушинского, г. Ярославль, Россия***Н. П. Федотова***победитель конкурса ИНФО-2024 в номинации**«40 лет школьной информатике: Олимпиады, конкурсы, проекты по информатике»,**Ярославский государственный университет им. П. Г. Демидова, г. Ярославль, Россия*

ЗАДАЧИ С МАТЕМАТИЧЕСКИМ СОДЕРЖАНИЕМ НА ОЛИМПИАДАХ И КОНКУРСАХ ПО ИНФОРМАТИКЕ

Аннотация

Математические понятия и алгоритмы присутствуют в базовом и углубленном курсах информатики основной и средней школы, в задачах единого государственного экзамена и задачах олимпиад и конкурсов по программированию. Содержательные линии математики и информатики во многом пересекаются, главным образом в разделах дискретной математики: математическая логика, теория графов, комбинаторика, системы счисления, модульная арифметика и др. В статье приводятся примеры математических задач, которые оказываются полезными при изучении различных разделов курса информатики. Рассматривается использование полуинвариантов и инвариантов при решении математических задач и верификации программ. Приведены примеры задач на нетрадиционные системы счисления, которые используются при создании аппаратного и программного обеспечения персональных компьютеров. Обсуждаются вопросы арифметики остатков, которые связаны с современными стандартами шифрования информации. Большинство задач, включенных в данную статью, рассчитаны на хорошо подготовленных и мотивированных школьников. Они предлагались авторами на различных олимпиадах и конкурсах по информатике для школьников Ярославской области.

Ключевые слова: математические основы информатики, инварианты, верификация программ, арифметика остатков, системы счисления.

Информация об авторах

Корнилов Петр Анатольевич, канд. физ.-мат. наук, доцент, доцент кафедры теории и методики обучения информатике, физико-математический факультет, Ярославский государственный педагогический университет им. К. Д. Ушинского, г. Ярославль, Россия; *e-mail:* kornilovpa@yandex.ru

Федотова Наталья Петровна, канд. физ.-мат. наук, доцент кафедры компьютерной безопасности и математических методов обработки информации, математический факультет, Ярославский государственный университет им. П. Г. Демидова, г. Ярославль, Россия; *e-mail:* natatulik@yandex.ru

P. A. Kornilov

Yaroslavl State Pedagogical University named after K. D. Ushinsky, Yaroslavl, Russia

N. P. Fedotova

Yaroslavl State University named after P. G. Demidov, Yaroslavl, Russia

TASKS WITH MATHEMATICAL CONTENT IN OLYMPIADS AND COMPETITIONS IN INFORMATICS

Abstract

Mathematical concepts and algorithms are present in the basic and advanced informatics courses of primary and secondary schools, in the tasks of the Unified State Exam and tasks of Olympiads and competitions in programming. The content lines of mathematics and informatics overlap in many ways, mainly in the sections of discrete mathematics: mathematical logic, graph theory, combinatorics, number systems, modular arithmetic, etc. The article provides examples of mathematical tasks that are useful when studying various sections of the informatics course. The use of semi-invariants and invariants in solving mathematical tasks and verifying programs is considered. Examples of tasks on non-traditional number systems that are used in creating hardware and software for personal computers are given. The issues of remainder arithmetic related to modern information encryption standards are discussed. Most of the tasks included in the article are created for well-prepared and motivated students. They were proposed by the authors at various Olympiads and competitions in informatics for schoolchildren in the Yaroslavl region.

Keywords: mathematical foundations of informatics, invariants, program verification, remainder arithmetic, number systems.

Information about the authors

Petr A. Kornilov, Candidate of Sciences (Physics and Mathematics), Docent, Associate Professor at the Department of Theory and Methods of Teaching Informatics, Faculty of Physics and Mathematics, Yaroslavl State Pedagogical University named after K. D. Ushinsky, Yaroslavl, Russia; *e-mail:* kornilovpa@yandex.ru

Natalia P. Fedotova, Candidate of Sciences (Physics and Mathematics), Associate Professor at the Department of Computer Security and Mathematical Methods of Data Processing, The Mathematics Faculty, Yaroslavl State University named after P.G. Demidov, Yaroslavl, Russia; *e-mail:* natatulik@yandex.ru

Введение

Информатика вводилась в школы СССР как предмет, где основное содержание составляло изучение программирования. В настоящее время, по мнению некоторых исследователей, главной содержательной линией информатики являются теоретические (математические) основы информатики [1, 8, 14]. Математические понятия и алгоритмы присутствуют в базовом и углубленном курсах информатики основной и средней школы, в задачах единого государственного экзамена и задачах олимпиад и конкурсов по программированию. Содержательные линии математики и информатики во многом пересекаются, главным образом в разделах дискретной математики: математическая логика, теория графов, комбинаторика, системы счисления, модульная арифметика и др. Изучение содержания перечисленных разделов может проходить на разных уровнях, но наиболее полно реализуется при подготовке одаренных и мотивированных школьников для успешного участия в олимпиадах и конкурсах по информатике.

В данной статье рассматривается несколько разделов, в которых математика и информатика переплетаются, поддерживают и развивают друг друга. Большинство примеров и задач, рассматриваемых в работе, взято из различных олимпиад и конкурсов по информатике для школьников, которые авторы проводили в течение последних лет.

1. Инварианты и верификация программ

При решении некоторого класса математических задач применяются полуинварианты и инварианты. **Полуинвариант** — это некоторая характеристика системы в целом, которая монотонно изменяется в ходе изучаемых процессов [16]. Полуинварианты помогают доказывать конечность процессов, обосновывают невозможность их заикливания. **Инварианты** — это характеристики системы, которые сохраняются в ходе преобразований системы. С их помощью часто делают вывод о невозможности перехода системы из начального состояния в конечное, а иногда применяют для доказательства корректности преобразований системы.

В информатике полуинварианты и инварианты используют при верификации программ. Например, полуинварианты применяют для доказательства конечности работы циклов с пост- или предусловием `while` и `repeat`, а инварианты применяют для доказательства корректности работы программ.

Рассмотрим некоторые примеры. Первую задачу наш авторский коллектив давал на областную олимпиаду по информатике 1993/1994 учебного года в Ярославской области [7].

Задача 1.1.

В лототроне N шаров, помеченных натуральными числами. Каждую секунду из лототрона вываливаются два шара. Если они помечены одинаковыми числами, то один выбрасывается, а второй возвращается обратно без изменений. Если числа разные, то большее из них заменяется на разность и оба шара возвращаются в лототрон.

1) Докажите конечность алгоритма.

2) Определите, чем все закончится и зависит ли результат от случая.

Решение.

В данном случае ответ на первый вопрос достаточно простой. В качестве полуинварианта можно выбрать сумму всех чисел, написанных на шарах в лототроне. Она монотонно убывает, остается целой и не может стать отрицательной. Поэтому описываемый процесс закончится. Понятно, что в этот момент в лототроне останется один шар.

Чтобы ответить на второй вопрос, можно сначала посмотреть на ситуацию с двумя шарами. Тогда процесс из неопределенного превращается в хорошо известный алгоритм Евклида нахождения наибольшего общего делителя (НОД) двух чисел. Это позволяет выдвинуть гипотезу, что и для любого количества чисел в конце на оставшемся шарике будет написан НОД всех чисел, бывших там изначально. Обоснование этой гипотезы производится точно так же, как и обоснование самого алгоритма Евклида. Для этого рассматриваем один шаг процесса и замечаем, что при переходе от пары (a, b) к паре $(a - b, b)$ общие делители чисел не могут как пропасть, так и появиться. При удалении же дубликата числа из набора НОД всех чисел тем более не может измениться.

Задача 1.2.

Определите, что делает данная программа. Ответ обоснуйте.

```
алг Х(вещ а, у, нат n)
  арг а, n
  рез у
  нач нат k, вещ t, b
  b := 1; t := a; k := n
  нц пока k > 0
    если mod(k, 2) = 0
      то t := t * t; k := div(k, 2)
      иначе b := b * t; k := k - 1
    все
  кц
  у := b
кон
```

Решение.

Для того чтобы появилась гипотеза о том, что делает эта программа, можно пошагово смоделировать ее работу при значениях аргументов, например, $n = 10$ и $a = 2$. Выпишем значения переменных b, t, k , которые меняются в цикле:

№ шага	b	t	k
0	1	2	10
1	1	4	5
2	4	4	4
3	4	16	2
4	4	256	1
5	1024	256	0

Видим, что ответом будет число 1024, которое сразу же помогает выдвинуть гипотезу, что данная программа вычисляет значение a в степени n .

Однако доказывать, что программа всегда работает корректно, какими-то обычными рассуждениями весьма затруднительно. А при использовании инвариантов доказательство будет убедительным. В данном случае, внимательно посмотрев на текст программы или на результаты ее исполнения на нашем примере, мы можем заметить, что неизменной в ходе выполнения цикла остается следующая комбинация изменяющихся параметров: $b \cdot t^k$. Действительно, на каждой итерации цикла в случае четного числа k имеем:

$$b \cdot t^k = b \cdot (t^2)^{k/2},$$

а в случае нечетного k :

$$b \cdot t^k = (b \cdot t) \cdot t^{k-1}.$$

Остается заметить, что в начале исполнения программы эта величина равна $b \cdot t^k = 1 \cdot a^n = a^n$, а в конце работы программы эта же величина равна $b \cdot t^k = y \cdot t^0 = y$. Тем самым, $y = a^n$.

Данный алгоритм имеет различные практические приложения, носит название «**алгоритм быстрого возведения в степень**» и применяется не только при возведении чисел в очень большие степени (что требуется в шифровании), но и в ряде других ситуаций. Например, при вычислении произведения двух чисел по очень большому модулю его заменяют на выполнение небольшого количества операций сложения чисел по данному модулю. Также при вычислении далеких членов рекуррентных последовательностей, задаваемых линейными соотношениями (простейший пример — числа Фибоначчи), можно свести задачу к возведению в большую степень некоторой матрицы и воспользоваться для этого данным алгоритмом.

Задача 1.3.

Определите, что делает данная программа. Ответ обоснуйте.

```

алг П(нат а, b, c)
  арг а, b
  рез с
  нач нат d, e
    d := b; c := 0
    нц пока a > 0
      e := mod(a, 2)
      a := div(a, 2)
      c := c + d * e
      d := d * 2
    кц
  кон

```

Решение.

Поступим аналогично решению предыдущей задачи. Сначала рассмотрим работу алгоритма на примере $a = 12$, $b = 7$:

№ шага	e	a	c	d
0	0	12	0	7
1	0	6	0	14
2	0	3	0	28
3	1	1	28	56
4	1	0	84	112

Возникает вполне логичная гипотеза, что этот алгоритм вычисляет произведение двух чисел. Действительно, в Древней Руси именно так вычисляли произведение натуральных чисел до появления алгоритмов умножения столбиком. Сущность старинного русского способа умножения состоит в том, что умножение любых двух чисел сводилось к ряду последовательных делений одного числа пополам (последовательное раздвоение) при одновременном удвоении другого числа. Например, если в произведении $24 \cdot 5$ множимое 24 уменьшить в два раза (раздвоить), а множитель увеличить в два раза (удвоить), т. е. взять произведение $12 \cdot 10$, то произведение остается равным числу 120.

Данный пример тоже относится к алгоритмам, имеющим практическое применение. В некоторых компиляторах он положен в основу умножения натуральных чисел, поскольку операции умножения и целочисленного деления на 2 — это просто сдвиг двоичной записи числа влево или вправо на 1 бит, которые выполняются чрезвычайно быстро.

Мы предоставляем читателю самостоятельно догадаться по записи алгоритма или по рассмотренному примеру, какая именно комбинация меняющихся в цикле параметров является инвариантом, и с помощью него обосновать корректность работы программы.

Рассмотрим задачу, которую авторы предлагали на областном командном чемпионате школьников по программированию в 2022/2023 учебном году.

Задача 1.4.

С парой чисел (x, y) многократно проводят следующую операцию:

$$\begin{cases} x = x \frac{\sqrt{3}}{2} + \frac{y}{2}; \\ y = y \frac{\sqrt{3}}{2} - \frac{x}{2}. \end{cases}$$

Требуется выяснить, какими будут два числа после проведения K операций (исходные значения x, y по модулю не превосходят 1000, K принимает значения от 1 до 10^9).

Решение.

Есть несколько подходов к решению данной задачи. Самый простой из них — просто записать данное преобразование в цикле от 1 до K . Но, ввиду достаточно больших ограничений в задаче на значение K , это решение набирало небольшое количество баллов. В условии задачи был приведен пример, в котором показывалось, что из пары чисел $(7, 13)$ после проведения 2022 операций получится пара $(-7, -13)$. Это была подсказка о том, что решение задачи носит математический характер, хотя можно было, используя данную подсказку, довести решение до полного уже и без знания математики. На самом деле, можно заметить, что инвариантом задачи является сумма квадратов чисел. Если после этого интерпретировать задачу геометрически, то можно было понять, что описанное преобразование — это просто поворот вектора. Если участник был знаком с матричной формулой поворота вектора или просто посмотрел выполнение одного шага на примерах, то он понимал, что за один шаг происходит поворот вектора на 30 градусов,

после чего задача становилась элементарной. Если участник не мог интерпретировать задачу геометрически, но был хорошо знаком с алгеброй, то он мог понять, что один шаг представляет собой умножение вектора на одну и ту же матрицу. В этом случае задача сводилась к возведению данной матрицы в большую степень K , для чего можно воспользоваться алгоритмом быстрого возведения в степень, который мы обсуждали в задаче 1.2.

2. Системы счисления

Системы счисления изучаются и в математике, и в информатике. В курсе информатики основное внимание уделяется изучению двоичной системы счисления и родственных ей восьмеричной и шестнадцатеричной систем счисления. Это объясняется тем, что в современных компьютерах вся информация представляется в виде двоичного кода. Такой выбор во многом объясняется физическими принципами хранения и обработки информации, например, есть сигнал — нет сигнала, участок намагничен — не намагничен и т. п. Между тем двоичная система счисления не является самой выгодной в смысле производительности и скорости вычислений.

Для иллюстрации приведем пример задачи, которая является частным случаем так называемой задачи Баше—Менделеева о гирях.

Задача 2.1.

Придумайте минимальный набор гирь, с помощью которого можно на чашечных весах взвесить любое целое число граммов лекарств от 1 до 40 граммов.

Решение.

Один из первых способов решения дает ответ 6 гирь, имея в виду представление чисел от 1 до 40 в двоичной системе счисления, для чего потребуются гири весом 1, 2, 4, 8, 16 и 32 грамма.

На самом деле этот ответ не является оптимальным. Здесь есть красивая нестандартная идея, что гири можно класть на разные чашки весов — и туда, где нет груза, и туда, где он находится. Это идея троично-симметричной системы счисления. В данном примере оказывается достаточным набора всего лишь из 4 гирек весом 1, 3, 9 и 27 граммов, что несложно проверяется непосредственно.

Еще одним преимуществом троично-симметричной системы счисления [11] является то, что при ее применении не возникает проблем с представлением в компьютере отрицательных чисел, для чего в современных компьютерах применяется дополнительный код [4, с. 68]. В троично-симметричной системе для представления отрицательного числа нужно всего лишь инвертировать цифры в записи модуля числа. Интересный исторический факт состоит в том, что в конце 1950-х годов в МГУ была разработана, реализована и запущена в серию ЭВМ «Сетунь», архитектура которой была основана именно на более экономной троично-симметричной системе.

Далее рассмотрим пример задачи о необычной системе счисления, идея которой пришла одному из авторов статьи при решении задачи заключительного этапа ВСОШ по математике.

Задача 2.2.

В стране математиков решили ввести в действие позиционную систему счисления с необычным основанием. Это иррациональное число x , которое является большим корнем уравнения $x^2 + x - 7 = 0$. Оказалось, что все натуральные числа можно представить в этой системе счисления, используя только цифры от 0 до 6, например, число 13 представляется в ней как 116. А как в этой системе представляется число 2024?

Решение.

Пример представления числа 13 в этой системе счисления с помощью цифр от 0 до 6, приведенный в условии задачи, является существенной подсказкой к составлению алгоритма перевода натуральных чисел в эту необычную систему счисления. Он показывает, что 7 единиц некоторого разряда можно заменить на одну единицу следующего и одну единицу следующего через один разряда вместе. Это можно увидеть и алгебраически, если данное в условии уравнение домножить на x в любой степени. Тогда имеем:

$$(x^2 + x - 7) \cdot x^k = 0 \quad \text{или} \quad x^{k+2} + x^{k+1} = 7 \cdot x^k.$$

В таком случае, начав с 2024 единиц в младшем разряде и применяя описанное преобразование переноса в старшие разряды поочередно справа налево, мы получим ответ к задаче: 133634121.

Как нам кажется, эта задача хорошо сочетается с подходом к формированию представления чисел в позиционных системах счисления, изложенных в статье Н. И. Заводчиковой [10], и будет полезна для мотивированных школьников.

Еще одной необычной системой счисления, которая применяется в информатике для проведения большого количества вычислений с большими числами, является **система остаточных классов** (СОК). В ней представление натуральных чисел основано на китайской теореме об остатках.

Для представления натуральных чисел в СОК выбирается базис системы — набор взаимно простых чисел $n_1, n_2, \dots, n_k$, а любое число x представляется в ней как набор остатков $x_1, x_2, \dots, x_k$ от деления x на числа базиса системы. При этом, по китайской теореме об остатках, доказательство которой было проведено в трактате Цзинь Цзюшао «Математические рассуждения в 9 главах», датированном 1247 годом, представление любого числа в диапазоне от 0 до произведения всех чисел базиса (не включительно) единственно [12]. Далее, арифметические операции в СОК определяются как операции с остатками в модульной арифметике по отдельности.

К преимуществам СОК надо отнести высокую скорость выполнения арифметических операций, особенно операций умножения. По сравнению с операциями умножения в обычных позиционных системах, где каждую цифру надо умножать на каждую и при этом следить за переносами, здесь каждый остаток умножается только на один такой же остаток у другого числа, причем отсутствие переносов позволяет распараллеливать процесс выполнения любой арифметической операции, что также заметно увеличивает производительность системы.

Но следует отметить и недостатки системы. К ним стоит отнести ограничения на обрабатываемые числа, которые закладываются при выборе базиса системы, неудобство сравнения или деления чисел. В течение последних лет ведется большая работа по устранению этих недостатков. Еще одним недостатком является непростой алгоритм получения самого числа по имеющимся остаткам.

Разберем один из вариантов восстановления числа по известным остаткам.

Задача 2.3.

Базис СОК состоит из четырех чисел: 5, 7, 11 и 13. Остатки от деления числа x на них равны 3, 6, 10 и 10 соответственно. Восстановите число x (в диапазоне от 0 до $5 \cdot 7 \cdot 11 \cdot 13 - 1 = 5004$).

Решение.

Будем восстанавливать число по шагам.

На первом шаге возьмем первый остаток 3.

Далее будем перебирать числа $3 + 5 \cdot k$ (чтобы не портить первый из остатков) и увеличивать k , пока не получим правильный второй остаток. При $k = 0$ остаток от деления на 7 равен 3, при $k = 1$ получаем остаток 1, при $k = 2$ получаем нужный остаток 6, а текущее число 13.

Теперь нам надо сохранять два полученных остатка, поэтому к числу 13 будем прибавлять по $5 \cdot 7 = 35$, пока не получим остаток 10 при делении на 11. У числа 13 остаток 2, у 48 — остаток 4, у 83 — остаток 6, у 118 — остаток 8, а у 153 получаем нужный остаток 10.

Далее надо двигаться с шагом $5 \cdot 7 \cdot 11 = 385$. Но уже при $k = 0$ мы получим нужный остаток, поэтому число 153 и будет ответом к задаче.

3. Арифметика остатков

Данная тема занимает важное место как в теории чисел в математике, так и в информатике. Дело в том, что обычно компьютер, когда работает с целыми числами, отводит под их хранение фиксированный объем памяти. Вследствие этого диапазон чисел, с которыми выполняются действия, ограничен и возникают проблемы с переполнением. Например, тип `byte` в языке Pascal отводит под хранение неотрицательных целых чисел 1 байт, поэтому значения переменных такого типа могут быть только от 0 до 255. Если же сложить 100 и 200, то получится ответ, который не может быть представлен в типе `byte`. В таком случае компиляторы вместо числа 300 будут считать ответом число $300 - 256 = 44$, что связано с физической реализацией операции сложения. Аналогично, при вычислении, например, разности $20 - 40$ ответом будет число $-20 + 256 = 236$, а при вычислении произведения $50 \cdot 41$ ответом будет число $2050 - 256 \cdot 8 = 2050 - 2048 = 2$. Тем самым выполнение арифметических операций с типом `byte` полностью соответствует выполнению операций в арифметике остатков (или вычетов) по модулю 256. В арифметике остатков мы можем привычным образом работать с операциями сложения, вычитания, умножения и возведения в степень. А вот с операцией деления могут возникать проблемы. Важное место, как мы увидим дальше, арифметика остатков занимает и в шифровании информации.

Очень близкой темой по содержанию к данной является тема, связанная с делителями чисел, которая также рассматривается в информатике, и задачи на эту тему даже включаются в ЕГЭ по информатике. Здесь тоже много красивых задач, связанных с основной теоремой арифметики, общим видом делителей, количеством и суммой делителей, признаками делимости и т. п. К сожалению, нам не удастся включить даже некоторые из них в данную статью, но любознательные читатели могут ознакомиться с обзором данной темы в работе [9].

В теории чисел существенную роль в доказательстве нескольких теорем играет следующее несложное утверждение.

Лемма. Пусть p — простое число, a — целое число, которое не делится на p . Тогда среди чисел $a, 2 \cdot a, 3 \cdot a, \dots, (p-1) \cdot a$, рассматриваемых по модулю p , будут представлены все остатки $1, 2, 3, \dots, p-1$ по одному разу.

Доказательство. Предположим, что среди рассматриваемых чисел есть два числа $m \cdot a$ и $n \cdot a$ с одинаковыми остатками. Тогда их разность $(m-n) \cdot a$ должна делиться на p , но ни один из сомножителей не может делиться на p , значит, и их произведение согласно основной теореме арифметики тоже не будет делиться на p . Следовательно, все остатки разные. Но при этом среди них нет остатка 0 по той же причине. Лемма полностью доказана.

Заметим, что требование простоты числа p здесь важно. Для любого составного числа p лемма неверна. Например, при вычислении по модулю 12 мы можем получить $3 \cdot 4 = 2 \cdot 6 = 8 \cdot 6 = 0$, а $5 \cdot 6 = 1 \cdot 6 = 3 \cdot 6 = 6$. Получается, что арифметика вычетов по модулю p «плохая», когда число p составное. Там нельзя сделать вывод, что если произведение равно 0, то один из сомножителей равен 0, нет операции деления. Если же число p простое, то из леммы следует, что у каждого ненулевого элемента есть обратный, так как в каждой строке таблицы умножения присутствуют все числа от 1 до $p-1$. Это позволяет определить операцию деления и работать с остатками так, как мы привыкли работать в обычной арифметике.

Хорошо подготовленным школьникам старших классов можно задать следующий сложный вопрос: а можно ли при каких-нибудь составных числах p так определить операцию умножения на множестве остатков при делении на p , чтобы в каждой строке таблицы умножения были представлены все ненулевые остатки? Этот же вопрос можно поставить перед ними сначала в частном случае, когда $p = 4$. Разумеется, очень сильным ограничением здесь является требование, чтобы мы могли, как обычно, работать с арифметическими выражениями, т. е. чтобы можно было переставлять местами сомножители, выполнять операции умножения нескольких сомножителей в любом порядке, выносить общий множитель в сумме слагаемых за скобки. Ответ здесь, как ни странно на первый взгляд, положительный. Для чисел вида p^k (и только для чисел такого вида), где p — простое число, так сделать можно. В частности, это можно делать для арифметик по модулю $256 = 2^8$, $65536 = 2^{16}$, Возникающие при этом «хорошие» арифметики носят название поля Галуа и положены в основание современных стандартов шифрования информации — американского стандарта DES (Data Encryption Standard) и российского стандарта ГОСТ 28147-89 [2, 15]. Оба они были раз-

работаны и реализованы в конце 70-х годов прошлого столетия, но до сих пор обеспечивают надежную защиту информации.

Выведем из леммы две важные теоремы [5, с. 96; 6, с. 117].

Малая теорема Ферма. Пусть p — простое число, a — целое число, не делящееся на p . Тогда $a^{p-1} = 1$ по модулю p .

Доказательство. Действительно, рассмотрим произведение всех чисел $a, 2 \cdot a, 3 \cdot a, \dots, (p-1) \cdot a$. С одной стороны, оно равно $a^{p-1} \cdot (p-1)!$, а, с другой стороны, по лемме оно равно просто $(p-1)!$. В силу того, что числа p и $(p-1)!$ взаимно простые, мы можем сократить равенство на $(p-1)!$ и получить утверждение теоремы.

Теорема Вильсона. Пусть p — простое число. Тогда $(p-1)! = -1$ по модулю p .

Доказательство. По лемме у каждого элемента есть ровно один обратный к нему элемент, поэтому мы можем разбить все сомножители, кроме 1 и -1 , на пары, каждая из которых дает в произведении единицу. Тогда все произведение равно -1 , что и требовалось доказать.

Малая теорема Ферма позволяет сильно упростить процесс вычисления больших степеней в арифметике остатков, а теорема Вильсона позволяет во многих случаях быстро вычислять факториалы больших чисел в арифметике остатков. Рассмотрим два примера, иллюстрирующих применение этих теорем.

Задача 3.1.

Может ли быть простым число $122^{2025} + 2025^{80}$?

Решение.

Это число имеет в десятичной записи несколько тысяч знаков, поэтому проверить его простоту чрезвычайно сложно. Заметим, правда, что современные методы могут справиться с этой задачей за разумное время. По формулировке задания похоже, что ответ отрицательный и нам надо найти какой-нибудь делитель.

Попробуем для решения этой задачи воспользоваться малой теоремой Ферма, чтобы избавиться от больших степеней. Для этого надо подобрать подходящее простое число. Оно должно быть связано либо с числом 2025, либо с числом 80, чтобы помочь нам. Но ни число 81, ни число 2026 простыми не являются. Тогда попробуем посмотреть на число 41 и остатки от деления слагаемых на него. При вычислении первого слагаемого по модулю 41 мы можем заменить число 122 на 40 или на -1 . Тогда сразу видим, что $(-1)^{2025}$ дает остаток -1 по модулю 41. А ко второму слагаемому мы можем применить малую теорему Ферма и увидеть, что $2025^{80} = (2025^{40})^2 = 1^2 = 1$ по модулю 41. Тогда в сумме получится остаток 0, что означает, что мы нашли делитель 41 у рассматриваемого числа. Ответ: не может.

Задача 3.2.

Найдите остаток от деления $2025!$ на 2027.

Решение.

Число 2027 является простым, поэтому мы можем применить теорему Вильсона. $2026! = -1 = 2026$ по модулю 2027. Но $2026 = 2026! = 2025! \cdot 2026$ по модулю 2027. Поскольку числа 2026 и 2027 взаимно простые, мы можем сократить равенство на 2026 и получить ответ $2025! = 1$ по модулю 2027.

Малая теорема Ферма является довольно неплохим инструментом в теории чисел. Поэтому предпринимались попытки доказать ее аналог для составных чисел. Это удалось Леонарду Эйлеру. Он ввел функцию, которая теперь называется в теории чисел функцией Эйлера $\varphi(n)$ и равна количеству натуральных чисел, которые меньше или равны n и взаимно просты с ним. Далее, он доказал аналог леммы и, соответственно, аналог малой теоремы Ферма, а именно: для любого натурального n и числа a , которое взаимно просто с n , выполнено $a^{\varphi(n)} = 1 \pmod{n}$.

Разберем следующую задачу.

Задача 3.3.

Докажите, что для любого натурального n число $n^{84} - n^4$ делится на 20400.

Решение.

Разложим на множители число $20400 = 16 \cdot 3 \cdot 25 \cdot 17$ и рассмотрим делимость на каждый из них по отдельности. Заметим также, что $n^{84} - n^4 = n^4 \cdot (n^{80} - 1)$. Если n — четное число, то первый множитель делится на 16. В противном случае, учитывая, что $\varphi(16) = 8$, имеем $n^{80} - 1 = (n^8)^{10} - 1 = 1 - 1 = 0$ по модулю 16.

Аналогично, вычисляя по модулю 3, имеем, что либо n делится на 3, либо $n^{80} - 1 = (n^2)^{40} - 1 = 1 - 1 = 0$.

То же самое сделаем по модулю 25, с учетом того, что $\varphi(25) = 20$. Либо n делится на 5, либо $n^{80} - 1 = (n^{20})^4 - 1 = 1 - 1 = 0$ по модулю 25.

И, наконец, по модулю 17 либо n делится на 17, либо $n^{80} - 1 = (n^{16})^5 - 1 = 1 - 1 = 0$.

Поскольку рассматриваемое число делится на каждый из множителей, а они взаимно простые, то оно делится и на их произведение, что нам и требовалось доказать.

Функция и теорема Эйлера играют ключевую роль в алгоритме шифрования с открытым ключом RSA (аббревиатура от фамилий Rivest, Shamir и Adleman).

Система RSA используется для защиты программного обеспечения и в схемах цифровой подписи. Также она используется в открытой системе шифрования PGP и иных системах шифрования (к примеру, DarkCryptTC и формат xdc) в сочетании с симметричными алгоритмами.

Впервые идею шифрования с открытым ключом предложили американские специалисты по криптографии Уитфилд Диффи (Whitfield Diffie) и Мартин Хеллман (Martin Edward Hellman) [17]. Схема, предложенная ими, следующая. Два участника конфиденциальной переписки задумывают свои секретные ключи. Далее первый участник выбирает некоторое сообщение g , простое число p , возводит g в степень своего секретного ключа и передает полученное сообщение A вместе со значениями g и p второму участнику. Тот возводит g в степень своего секретного ключа и передает полученное сообщение B первому участнику. Передача сообщений происходит по открытому каналу связи. Затем первый участник возводит B в степень своего секретного ключа, а второй возводит A в степень своего секретного ключа. Они получают одно и то же число K , поскольку оно является результатом возведения g в степень, равную

произведению двух секретных ключей. Именно число K и является общим секретным ключом этих двух участников. При этом, даже зная величины g, p, A, B , посторонний человек не может вычислить величину K . Причина заключается в том, что возведение числа в большую степень по некоторому модулю считается односторонней функцией. Это означает, что, зная основание и показатель, вычислить результат возведения в степень по известному модулю легко (см. хотя бы задачу 1.2), а, зная результат возведения в степень по известному модулю и основание, определить показатель степени (так называемая задача нахождения дискретного логарифма) практически невозможно за разумное время, если числа достаточно большие.

В алгоритме RSA для шифрования используется операция возведения в степень по модулю большого числа n , которое равно произведению двух больших простых чисел p и q . Для дешифрования надо найти обратное число к открытому ключу по модулю $\varphi(n)$, для чего необходимо уметь вычислять функцию Эйлера от данного большого числа, а для этого необходимо знать разложение числа на простые множители. В настоящее время для шифрования используются простые числа p и q , состоящие примерно из 200–300 цифр каждое. Считается, что, зная произведение двух таких чисел, найти сомножители (задача факторизации большого числа) — это неразрешимая задача за разумное время. В то же время вычислить саму функцию произведения двух чисел можно быстро.

Итак, зная число $n = p \cdot q$ и функцию Эйлера

$$\varphi(n) = (p - 1) \cdot (q - 1),$$

участник выбирает число e , взаимно простое с $\varphi(n)$, и подбирает обратное к нему по модулю $\varphi(n)$ число d , т. е.

$$e \cdot d = 1 \pmod{\varphi(n)}.$$

Обычно это делают с помощью расширенного алгоритма Евклида, но для маленьких чисел мы покажем, как это можно сделать с помощью функции Эйлера: $e^{\varphi(\varphi(n))} = 1 = e^{\varphi(\varphi(n)) - 1} \cdot e$, поэтому можно взять $d = e^{\varphi(\varphi(n)) - 1}$. Далее участник передает открытый ключ (e, n) тому, от кого он хочет получить сообщение, отправитель возводит свое сообщение S в степень e по модулю n и передает результат R . Получатель возводит полученное сообщение R по модулю n в степень d , которую знает только он и, поскольку $e \cdot d = 1 \pmod{\varphi(n)}$, то

$R^d = (S^e)^d = (S)^{e \cdot d} = (S)^{k \cdot \varphi(n) + 1} = (S^{\varphi(n)})^k \cdot S = 1^k \cdot S = S \pmod{n}$, значит, он получает исходное сообщение в первой степени, т. е. оригинал сообщения.

Рассмотрим пример задачи для очень маленьких чисел p и q .

Задача 3.4.

Пусть $p = 5$, $q = 7$, $e = 5$. Найдите закрытый ключ d и расшифруйте пришедшее сообщение 33.

Решение.

Имеем:

$$n = 5 \cdot 7 = 35;$$

$$\varphi(n) = (5 - 1) \cdot (7 - 1) = 24.$$

Тогда закрытый ключ:

$$d = 5^{23} = 5 \cdot 5^{22} = 5 \cdot (5^2)^{11} = 5 \cdot 1 = 5 \pmod{24}.$$

Чтобы восстановить сообщение, надо 33 возвести в степень 5 по модулю 35:

$$33^5 = (-2)^5 = (-32) = 3 \pmod{35}.$$

Ответ: было передано сообщение 3.

Заключение

Изучение теоретических основ информатики невозможно без использования серьезного и разнообразного математического аппарата, это относится в равной степени к школьному и вузовскому курсам информатики. Тем более это относится к олимпиадам и конкурсам по информатике и программированию, в которых участвуют наиболее одаренные и подготовленные школьники.

Как мы видели на многих примерах, задачи из информатики часто имеют аналоги в математике, а многие задачи и методы из математики, как оказывается, имеют прикладное значение в информатике.

Мы желаем нашим читателям встретить еще много красивых и полезных задач, в частности, их можно найти в задачниках [3, 7, 13, 16], а также во многих других источниках.

Список источников

1. Абдуразаков М. М., Лягинова О. Ю., Цветкова О. Н. Информатика, математика и логика в аспекте междисциплинарной и метапредметной образовательной связи // Чебышевский сборник. 2021. Т. 22. Вып. 2. С. 373–388. EDN: RUTLFH. DOI: 10.22405/2226-8383-2021-22-2-373-388.
2. Алферов А. П., Zubov A. Ю., Кузьмин А. С., Черемушкин А. В. Основы криптографии. М.: Гелиос АРБ, 2002. 480 с.
3. Алфутова Н. Б., Устинов А. В. Алгебра и теория чисел. Сборник задач для математических школ. М.: МЦНМО, 2002. 264 с.
4. Андреева Е. В., Босова Л. Л., Фалина И. Н. Математические основы информатики. М.: БИНОМ. Лаборатория знаний, 2012. 312 с. EDN: WPNTBT.
5. Бухштаб А. А. Теория чисел. М.: Просвещение, 1966. 384 с.
6. Виленкин Н. Я. Алгебра и теория чисел. 2-е изд. М.: Просвещение, 1984. 192 с.
7. Волченков С. Г., Корнилов П. А., Белов Ю. А., Дашниц Н. Л., Никулин В. А., Заводчикова Н. И. Ярославские олимпиады по информатике. Сборник задач с решениями. М.: БИНОМ. Лаборатория знаний, 2010. 406 с. EDN: SUGHPX.
8. Журавлев Ю. И. Фундаментально-математический и общекультурный аспекты школьной информатики // Вопросы образования. 2005. № 3. С. 192–200. EDN: ZUYHMG.
9. Заводчиков М. А., Заводчикова Н. И. Элементы теории чисел как математическая основа решения задач на обработку целочисленной информации // Информатика в школе. 2021. № 8. С. 32–36. EDN: HNRMEM. DOI: 10.32517/2221-1993-2021-20-8-32-36.
10. Заводчикова Н. И. Использование графической системы разрядов при изучении темы «Системы счисления» // Информатика в школе. 2023. № 2. С. 55–62. EDN: WDDJTC. DOI: 10.32517/2221-1993-2023-22-2-55-62.
11. Златопольский Д. М. Об уравновешенной троичной системе счисления // Мир информатики. 2017. № 9. С. 1–4. https://infojournal.ru/wp-content/uploads/2020/03/mir_info-2-2017-1.pdf

12. *Нестеренко А. Ю.* Введение в современную криптографию. Теоретико-числовые алгоритмы. М.: Изд-во Московского государственного института электроники и математики (технический университет), 2011. 190 с.

13. *Сгибнев А. И.* Делимость и простые числа. 2-е изд., стереотип. М.: МЦНМО, 2013. 112 с.

14. *Семенов А. Л.* Современный курс математики и информатики в школе // Содержание образования. 2004. № 1. С. 103–118. EDN: ZOQKOM.

15. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования данных. ГОСТ 28147-89. М.: ИПК «Издательство стандартов», 1989. 28 с.

16. *Шень А. Х.* Программирование: теоремы и задачи. М.: МЦНМО, 2017. 320 с.

17. *Diffie W., Hellman M. E.* New Directions in Cryptography // IEEE Transactions on Information Theory. 1976. Vol. 22. Is. 6. P. 644–654.

DOI: 10.32517/2221-1993-2025-24-1-18-25

**Т. М. Шамсутдинова***победитель конкурса ИНФО-2024 в номинации**«40 лет школьной информатике: Олимпиады, конкурсы, проекты по информатике»,**Башкирский государственный аграрный университет, г. Уфа, Республика Башкортостан, Россия*

ВОЗМОЖНОСТИ ТАБЛИЧНЫХ ПРОЦЕССОРОВ ПО РАЗВИТИЮ АНАЛИТИЧЕСКОЙ КОМПЕТЕНЦИИ ОБУЧАЮЩИХСЯ (НА ПРИМЕРЕ ОЛИМПИАДНЫХ ЗАДАЧ)

Аннотация

В статье рассмотрены возможности табличных процессоров в контексте формирования аналитической компетенции обучающихся. Дается определение аналитической компетенции, выделяются ее компоненты (абстрактно-логический, критический, системный, проблемный, образный, алгоритмический, творческий, ассоциативный, рефлексивный). Рассматривается роль олимпиадных задач в образовательном процессе как средства развития аналитического критического мышления, формирования навыков работы с данными и способности к принятию обоснованных решений. Приводятся авторские задачи по теме «Табличные процессоры» для проведения олимпиады как для школьников-старшеклассников, так и для студентов младших курсов вузов. Делается вывод, что формирование аналитического мышления у учащихся является комплексной составной проблемой, включающей в себя повышение абстрактности мышления, развитие логики, формирование творческого подхода к решению задач и др.

Ключевые слова: информатика, анализ данных, аналитическая компетенция, визуализация, моделирование, олимпиада, задачи, табличные процессоры.

Информация об авторе

Шамсутдинова Татьяна Михайловна, канд. физ.-мат. наук, доцент, доцент кафедры цифровых технологий и прикладной информатики, экономический факультет, Башкирский государственный аграрный университет, г. Уфа, Республика Башкортостан, Россия; *e-mail*: tsham@rambler.ru

1. Введение

Современное образование требует от учащихся не только усвоения фактических знаний, но и умения работать с информацией, делать логические выводы и принимать обоснованные решения в условиях стремительного роста объемов данных. В этой связи особую роль играют методы и инструменты, способствующие развитию аналитических навыков, включая умение

структурировать, обрабатывать и интерпретировать информацию.

Один из эффективных инструментов, способствующих формированию у школьников и студентов навыков анализа информации, — олимпиады и интеллектуальные соревнования.

Тема «Табличные процессоры» обладает большим потенциалом для развития аналитических способностей у обучающихся. Решения олимпиадных задач, предлага-

T. M. Shamsutdinova

Bashkir State Agrarian University, Ufa, The Republic of Bashkortostan, Russia

CAPABILITIES OF SPREADSHEET PROCESSORS FOR DEVELOPING ANALYTICAL COMPETENCY OF STUDENTS (ON THE EXAMPLE OF OLYMPIAD TASKS)

Abstract

The article considers the capabilities of spreadsheet processors in the context of developing students' analytical competency. It defines analytical competency and identifies its components (abstract-logical, critical, systemic, problematic, figurative, algorithmic, creative, associative, and reflexive). It considers the role of Olympiad tasks in the educational process as a means of developing analytical critical thinking, developing data handling skills, and the ability to make informed decisions. The author's tasks on the theme "Spreadsheet processors" are provided for conducting the Olympiad for both high school students and junior university students. It is concluded that developing students' analytical thinking is a complex problem that includes increasing the abstractness of thinking, developing logic, developing a creative approach to solving problems, etc.

Keywords: informatics, data analysis, analytical competency, visualization, modeling, Olympiad, tasks, spreadsheets.

Information about the author

Tatiana M. Shamsutdinova, Candidate of Sciences (Physics and Mathematics), Docent, Associate Professor at the Department of Digital Technologies and Applied Informatics, The Faculty of Economics, Bashkir State Agrarian University, Ufa, The Republic of Bashkortostan, Russia; *e-mail*: tsham@rambler.ru

емых участникам для решения в электронных таблицах, предполагают не только простые вычисления или механическое применение формул. Они требуют нестандартного подхода, креативности и глубокого понимания как информатики, так и математической логики. Решение таких задач способствует развитию у учащихся логического и критического мышления, а также способности к моделированию, к анализу и синтезу информации.

Олимпиадные задачи не только способствуют формированию аналитической компетенции школьников, но и формируют у них интерес к учебе, развивают самостоятельность и желание к самосовершенствованию, что делает олимпиады эффективным инструментом в образовательном процессе.

2. Понятие аналитической компетенции и ее компоненты

В статье Е. В. Сергунцовой, О. И. Михоненко, Е. И. Зритневой [8] говорится, что аналитические умения представляют собой систему определенных мыслительных действий, направленных на поиск оценки, систематизацию и обобщение знаний. Л. А. Саенко, Г. Н. Соломатина в работе [7] выделяют следующие мыслительные действия, лежащие в основе аналитических умений: анализ, обобщение, сравнение, классификация, сопоставление, структурирование, моделирование, прогнозирование, рефлексия. В статье А. В. Скрипкиной, Ю. Ю. Дзюбы [9] отмечается, что аналитические умения включают такие мыслительные процедуры, как анализ, синтез, сравнения, обобщения, аналогии, установление причинно-следственных связей и др. В. В. Осипова, Т. П. Бугаева, В. А. Осипов в работе [6] выделяют такие операции аналитического мышления, как структурирование, абстрагирование, классификации, анализ, сравнение, обобщение, определение противоречий, переход от общего к частному и др. В своей статье [11] О. Н. Ярыгин, В. М. Рябова отмечают, что аналитическая деятельность есть процесс взаимодействия интеллектуальных способностей, мотивации, креативности и рефлексии.

В рамках данной статьи будем считать, что *аналитическая компетенция представляет собой совокупность знаний, умений и навыков, позволяющих эффективно анализировать информацию, систематизировать и структурировать данные, распознавать причинно-следственные связи, делать обоснованные выводы и др.*

Обобщая вышесказанное и учитывая результаты работ [1, 10] и др., можем выделить следующие **компоненты аналитической компетенции**:

- *абстрактно-логический*, позволяющий абстрагироваться от несущественных деталей для выявления наиболее значимых признаков, делать логические заключения, находить причинно-следственные связи, выявлять общие признаки и свойства, синтезировать информацию, а также выявлять логические связи между фактами и аргументами, вести последовательные рассуждения;
- *критический*, включающий умение проводить сравнение информации, принимать взвешенное решение при выборе методов анализа и интерпретации данных, находить оптимальное решение;

умение оценивать информацию, выявлять слабые места в аргументации и анализировать различные точки зрения;

- *системный*, позволяющий систематизировать, структурировать информацию, проводить ее классификацию и кластеризацию, выявлять взаимодействия между различными компонентами систем;
- *проблемный*, отвечающий за проблемное мышление и включающий умение формулировать проблемы, определять цели и находить подходы к их достижению, включая сопоставление альтернатив, разработку стратегии и плана выполнения;
- *образный* (включая наглядно-образный и пространственно-образный), помогающий проводить визуализацию данных, оперировать образами понятий и объектов, создавать пространственные образы, осуществлять переход от пространственных образов реальных объектов к их условно-графическим изображениям, представлять информацию в наглядной форме, что помогает лучше понять и интерпретировать данные;
- *алгоритмический*, позволяющий разрабатывать алгоритмы решения типовых задач профессиональной деятельности, представлять алгоритмы как с помощью формализованных алгоритмических языков и систем программирования, так и с помощью команд естественного языка;
- *творческий* — как способность находить новые и оригинальные решения для сложных проблем, включая генерацию идей и альтернативных подходов, синтез инновационных решений;
- *ассоциативный*, помогающий формировать набор ассоциаций, являющихся базой ассоциативного мышления; включающий умение выстраивать ассоциативные связи, прогнозировать результаты; умение делать выводы, находить аналогии, применять уже известные решения к новым ситуациям в решении проблем;
- *рефлексивный*, выступающий как умение анализировать результаты собственной деятельности, проводить анализ своих целей, качеств, достижений; позволяет объективно оценить результаты своей интеллектуальной деятельности, а также наметить векторы профессионального самосовершенствования [9].

Развитие данных компонентов способствует повышению общего уровня аналитического мышления и позволяет эффективно решать сложные задачи как в процессе обучения, так и в сфере самостоятельной профессиональной деятельности.

3. Возможности табличных процессоров по развитию аналитического мышления

Как отмечается в работе Н. Д. Жилиной, Л. Б. Таренко [4], использование развивающих заданий, отражающих структуру и содержание аналитических умений, является одним из условий, способствующих эффективному формированию аналитических компетенций.

Табличные процессоры представляют собой мощные инструменты, помогающие развитию аналитических навыков у обучающихся. Электронные таблицы позволяют

не только обрабатывать и визуализировать данные, но и систематизировать, структурировать информацию, анализировать динамику изменения данных и получать обоснованные выводы, что делает их незаменимым ресурсом в современном образовательном процессе.

3.1. Олимпиадные задачи как метод развития аналитических навыков

Олимпиадные задачи представляют для обучающихся уникальную возможность углубить свои знания в области аналитики, научиться применять их на практике и развивать критическое мышление.

В ходе участия в олимпиадах школьники и студенты сталкиваются с задачами, которые требуют творческого, креативного подхода, умения эффективно решать проблемы моделирования, структурирования и визуализации информации [3, 5]. Табличные процессоры являются при этом одним из интересных, многогранных способов обработки данных, способствуют развитию ключевых аналитических навыков у учащихся [2]. Умение работать со сложными формулами, создавать нестандартные графики и использовать функции анализа данных позволяет обучающимся успешно решать развивающие задачи, а также улучшать логическое, критическое и алгоритмическое мышление.

Таким образом, решение олимпиадных задач с использованием табличных процессоров представляет собой эффективный метод, который может существенно обогатить процесс обучения, способствуя формированию аналитической компетенции у обучающихся.

3.2. Примеры олимпиадных задач по теме «Табличные процессоры»

Олимпиада по теме «Табличные процессоры» предполагает увлекательный формат, позволяющий участникам не только продемонстрировать свои знания

и умения, но и развить критическое мышление, навыки анализа и моделирования, а также научиться эффективно визуализировать информацию.

Далее приведем авторские задачи с общим тематическим сюжетом для проведения олимпиад как для школьников IX—XI классов, так и для студентов младших курсов вузов.

Задачи идут в порядке нарастающей сложности, баллы за решение могут варьироваться в зависимости от уровня подготовки обучающихся и от степени проработанности соответствующих тем на предшествующих уроках.

Задача 1. Графики функций (2–3 балла).

Витя и Даша получили на математике задание построить графики следующих функций: Витя — график функции $y = \sin(x)$; Даша — график функции $y = x^2 - x - 0.5$.

Помогите Вите и Даше. Постройте в Microsoft Excel график, на котором будут одновременно изображены обе функции и две точки их пересечения.

Решение.

Пример возможного решения задачи с соответствующими расчетами представлен на рисунке 1. Основная трудность, возникающая у обучающихся при решении данной задачи, — это отсутствие в исходных данных диапазона и шага изменения переменной x , что стимулирует анализировать условие задачи с использованием математических методов.

Задача 2. Таблица степеней (3 балла).

Выполняя домашнее задание по теме «Двоичная система счисления», Витя и Даша поняли, что не очень хорошо помнят степени числа 2.

Заполните для Вити и Даши в MS Excel таблицу чисел вида 2^N , где N принимает значения от 0 до 29 (рис. 2).

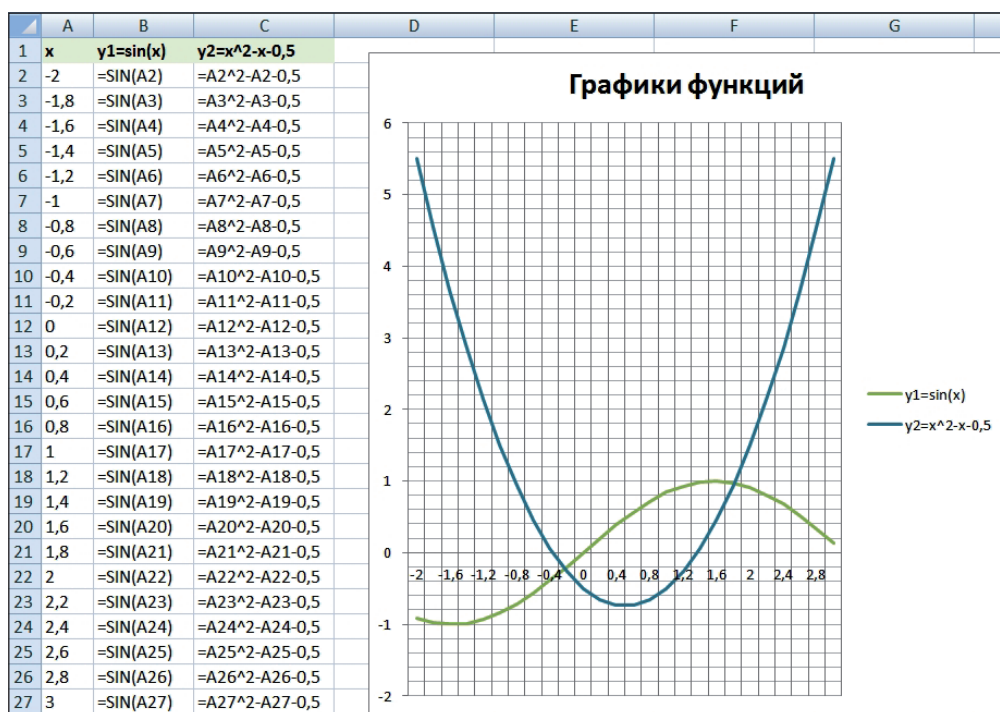


Рис. 1. Решение задачи 1 (в режиме показа формул)

	A	B	C	D	E	F	G	H	I	J	K	L
1			единицы									
2			0	1	2	3	4	5	6	7	8	9
3	Десятки	0	1	2	4							
4		1	1024	2048	4096							
5		2										

Рис. 2. Образец таблицы к задаче 2

	A	B
1	Цвет	Перевод
2	red	красный
3	orange	оранжевый
4	yellow	желтый
5	green	зеленый
6	cyan	голубой
7	blue	синий
8	purple	фиолетовый
9		
10	Введите цвет	purple
11	Перевод	фиолетовый

Рис. 3. Образец оформления задачи 3

В ячейке C3, соответствующей числу 2^0 , должна быть записана формула, которая затем должна быть автозаполнена во все остальные ячейки диапазона C3:L5.

Решение.

В качестве формулы в ячейке C3 для последующего автозаполнения может быть использована формула вида: $=2^{(10*B3+C$2)}$.

Задачи такого типа требуют от учащихся четкого понимания структуры адресации ячеек в контексте имеющейся табличной модели.

Задача 3. Словарь (3–4 балла).

Витя и Даша изучают английский язык и составили в MS Excel небольшой словарь перевода названий цветов *red, orange, yellow, green, cyan, blue, purple* на русский язык.

Английские названия цветов хранятся в ячейках A2:A8, их перевод — в ячейках B2:B8 (рис. 3).

В ячейку B10 вводится любой англоязычный цвет из списка изучаемых слов. Создайте в ячейке B11 формулу, подставляющую русский перевод введенного в ячейку B10 цвета.

Решение.

Несмотря на наличие довольно короткого решения в виде формулы:

$=ВПР(B10;A2:B8;2;0)$,

надо заметить, что функция ВПР относительно редко изучается в курсе школьной информатики, что дает на

олимпиаде эффект новизны и становится новым вызовом для многих учащихся.

Задача 4. Добрые дела (4 балла).

Во время летних каникул Витя, Даша и их друзья по отряду «Добрые дела» отправились в деревню помогать полоть грядки с овощами. Витя прополот 500 м грядок свеклы, 100 м грядок лука и 100 м грядок огурцов. Даша прополот 200 м грядок огурцов и 500 м грядок свеклы. Андрей прополот 200 м грядок лука и 300 м грядок свеклы. Полина прополот 300 м грядок свеклы и 300 м грядок лука.

За прополку школьникам давали памятные значки: один значок — за 100 м грядок свеклы, два значка — за 100 м грядок лука и три значка — за 100 м грядок огурцов.

Постройте таблицу MS Excel, в которую будут введены все исходные данные и будет наглядно рассчитано:

- сколько всего метров грядок овощей прополот каждый из школьников и сколько всего значков он получил;
- сколько метров было прополото по каждому виду овощей и сколько всего значков получено за каждый вид прополки;
- сколько всего значков получил весь отряд за работу;
- какое место занял каждый из школьников в соответствии с полученным числом значков.

Решение.

Пример возможного решения задачи с соответствующими формулами расчетами представлен на рисунке 4.

	A	B	C	D	E	F	G
1		Свекла, м	Лук, м	Огурцы, м	Всего, м	Значки	Место
2	Витя	500	100	100	=СУММ(B2:D2)	=(B\$8*B2+C\$8*C2+D\$8*D2)/100	=РАНГ(F2;F\$2:F\$5)
3	Даша	500		200	=СУММ(B3:D3)	=(B\$8*B3+C\$8*C3+D\$8*D3)/100	=РАНГ(F3;F\$2:F\$5)
4	Андрей	300	200		=СУММ(B4:D4)	=(B\$8*B4+C\$8*C4+D\$8*D4)/100	=РАНГ(F4;F\$2:F\$5)
5	Полина	300	300		=СУММ(B5:D5)	=(B\$8*B5+C\$8*C5+D\$8*D5)/100	=РАНГ(F5;F\$2:F\$5)
6	Всего	=СУММ(B2:B5)	=СУММ(C2:C5)	=СУММ(D2:D5)	=СУММ(E2:E5)	=СУММ(F2:F5)	
7		Норма значков за 100 м					
8		1	2	3			
9	Значки	=B6*B8/100	=C6*C8/100	=D6*D8/100			

Рис. 4. Пример возможного решения задачи 4

Но надо заметить, что данная задача допускает большое количество возможных вариантов моделирования и оформления таблицы, побуждая учащихся мотивированно выбрать оптимальный, с их точки зрения, вариант.

Задача 5. Дежурные (4 балла).

Витя и Даша вызвались создать для своего класса группу в социальной сети.

По нечетным дням дежурным по группе будет Витя, а по четным — Даша.

С использованием функций MS Excel определите, кто будет дежурить в любую введенную дату. Формат ввода даты: дд.мм.год.

Пример оформления решения задачи представлен на рисунке 5. Ячейка B5 заполняется датой вручную, в ячейке B6 содержится расчетная формула.

Решение.

Одним из возможных решений может быть вложение функций вида:

=ЕСЛИ(ЕЧЁТН(ДЕНЬ(B5));"Даша";"Витя"),
требующих наличия навыков работы с логическими функциями и умения обрабатывать даты.

Задача 6. Путешествия (5–6 баллов).

Витя и Даша уже так устали учиться, что решили составить себе график путешествий на будущие летние каникулы (рис. 6).

Рассчитайте продолжительность путешествий (в днях).

Постройте по данной таблице в MS Excel диаграмму по образцу с рисунка 7, используя как основу стандартный линейчатый тип диаграмм.

Построенная диаграмма считается соответствующей образцу при выполнении следующих критериев:

- за основу взят стандартный линейчатый тип диаграмм, доработанный за счет различных параметров настройки построенной диаграммы;

- поездки расположены в требуемом порядке;
- даты находятся по верхней оси;
- дата начала графика совпадает с датой начала первой поездки;
- полосы диаграммы визуально соответствуют продолжительности и периодам поездок в таблице;
- есть требуемое название диаграммы.

Решение.

Данная задача служит для развития навыков по визуализации данных. Для начала необходимо рассчитать продолжительность поездок, используя формулы вида: =C2–B2.

Заметим, что предложенный на рисунке 7 образец диаграммы представляет собой один из вариантов диаграммы Ганта и может быть реализован в MS Excel с помощью следующих шагов:

- построить *Линейчатую диаграмму с накоплением* для полей «Поездка» и «Начало»;
- далее с помощью команды *Выбрать данные.../Добавить данные* добавить ряд «Продолжительность»;
- с помощью команды формата параметров вертикальной оси установить *Обратный порядок категории*;
- далее установить для ряда «Начало» белый (либо бесцветный) цвет заливки;
- просмотреть числовое значение для первой по счету даты начала поездок (ячейка B2 в исходной таблице), для этого изменить в ней тип *Дата* на тип *Числовой*, для даты 22.06.2025 это будет число 45830;
- с помощью команды формата горизонтальной оси установить *фиксированное минимальное значение оси* как 45830;
- оформить построенную диаграмму, дать ей название.

	А	В
1	Даты	Дежурный
2	нечетные	Витя
3	четные	Даша
4		
5	Введите дату	01.01.2025
6	Дежурный	Витя

Рис. 5. Образец оформления задачи 5

	А	В	С	Д
1	Поездка	Начало	Конец	Продолжительность
2	Озеро Байкал	22.06.25	06.07.25	
3	Казань	08.07.25	12.07.25	
4	Каспийское море	15.07.25	25.07.25	
5	Пещера Шульган-Таш	27.07.25	30.07.25	
6	Крым	05.08.25	26.08.25	

Рис. 6. Исходные данные задачи 6

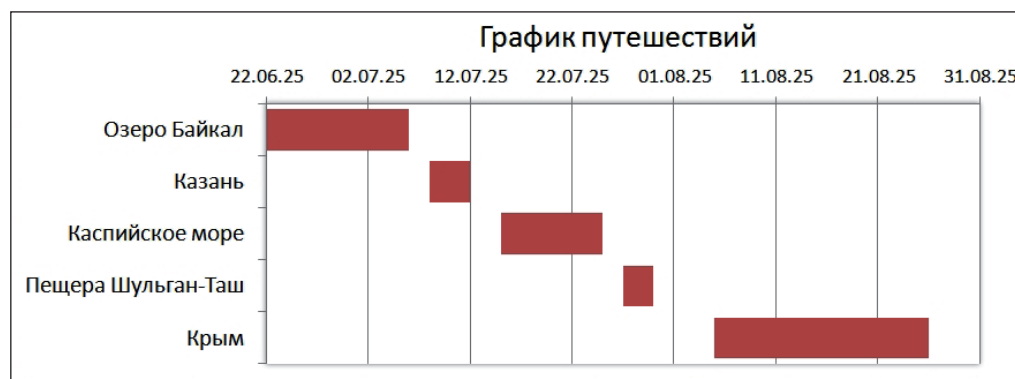


Рис. 7. Образец диаграммы задачи 6

Задача 7. Конкурс проектов (6–7 баллов).

Витя и Даша решили принять участие в конкурсе научных проектов среди школьников и студентов города. По правилам конкурса итоговый балл для каждого участника складывается из суммы баллов за каждый из трех критериев (научная новизна, сложность проекта и его презентация) плюс 5 % от полученной суммы для участников-школьников, при этом итоговый балл округляется до целого числа по математическим правилам.

Участники, набравшие 29 и более баллов, получают диплом 1-й степени, набравшие 28 баллов, — диплом 2-й степени, набравшие 27 баллов, — диплом 3-й степени.

Рассчитайте таблицу результатов конкурса (рис. 8), используя для столбцов «Итоговый балл» и «Степень диплома» формулы с последовательным вложением различных математических и логических функций с возможностью применения приема автозаполнения (т. е. формулы создаются для ячеек G2 и H2, далее автозаполняются на весь столбец).

Для столбца «Степень диплома» при этом реализуйте возможность автоматического проставления цифр 1, 2, 3 для участников, получивших дипломы 1-й, 2-й и 3-й степени соответственно, а также цифры 0 — для остальных участников.

Под таблицей рассчитайте количество дипломов каждой из степеней.

Решение.

Задачи такого типа (на использование сложных вложенных логических, математических и статистических функций) часто используются в составе Федерального интернет-экзамена в сфере профессионального образования (ФЭПО), поэтому навык в их решении может пригодиться не только школьникам в рамках подготовки к ОГЭ и ЕГЭ, но и студентам младших курсов вузов.

Ниже представлены возможные формулы расчета.

Итоговый балл:

=ОКРУГЛ(ЕСЛИ(C2="школа";СУММ(D2:F2)*1,05;СУММ(D2:F2));0)

Степень диплома:

=ЕСЛИ(G2>=29;1;ЕСЛИ(G2=28;2;ЕСЛИ(G2=27;3;0)))

Количество дипломов 1-й, 2-й, 3-й степени соответственно:

=СЧЁТЕСЛИ(H\$2:H\$11;"=1")

=СЧЁТЕСЛИ(H\$2:H\$11;"=2")

=СЧЁТЕСЛИ(H\$2:H\$11;"=3")

Задача 8. Спираль Архимеда (12–15 баллов).

Витя и Даша читали книгу по занимательной математике и узнали, что существует спираль Архимеда с уравнениями точек $x = t \cdot \cos(t)$; $y = t \cdot \sin(t)$, где t меняется от 0 до бесконечности. А еще ребята узнали, что такое кластер из точек, и поспорили, где будет находиться центр кластера из точек данной спирали.

Помогите Вите и Даше построить эту спираль как точечную диаграмму в MS Excel для t от 0 до 10 с шагом 0.1 (общее количество точек — 101). А также найдите центр кластера из построенных 101 точки в следующей системе предположений:

- Под центром кластера будем понимать место на графике, сумма расстояний от которого до всех точек кластера *минимальна*.
- Под расстоянием понимается расстояние Евклида между двумя точками $A(x_1, y_1)$ и $B(x_2, y_2)$ на плоскости, которое вычисляется по формуле:

$$d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

Необходимо использовать надстройку MS Excel *Поиск решения*, применяемую для решения задач оптимизации. Задача оптимизации в данном случае сводится к задаче минимизации суммы расстояний от центра кластера A_0 до всех его точек A_i для всех $i = 1 \dots 101$, т. е. выполняется условие вида:

$$\sum_{i=1}^{101} d_i(A_i, A_0) \rightarrow \min.$$

	A	B	C	D	E	F	G	H
1	№	Имя участника	Место учебы	Новизна	Сложность	Презентация	Итоговый балл	Степень диплома
2	1	Витя	школа	8	9	9		
3	2	Даша	школа	8	8	10		
4	3	Олег	вуз	7	7	8		
5	4	Сергей	вуз	8	8	9		
6	5	Яна	вуз	10	9	9		
7	6	Миша	вуз	10	9	10		
8	7	Артем	вуз	8	9	7		
9	8	Егор	школа	9	9	9		
10	9	Артур	школа	7	8	6		
11	10	Вика	вуз	8	10	8		
12								
13	Количество дипломов 1-й степени							
14	Количество дипломов 2-й степени							
15	Количество дипломов 3-й степени							

Рис. 8. Данные для задачи 7

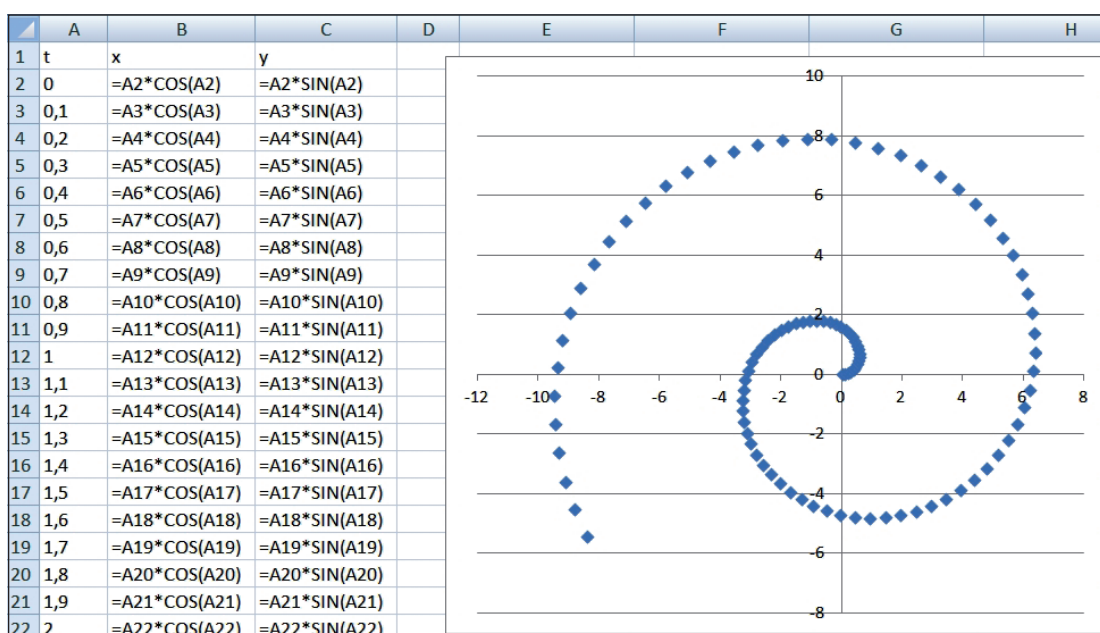


Рис. 9. Построение спирали в задаче 8 (фрагмент рабочего листа в режиме показа формул)

	A	B	C	D	E	F	G	H
1	t	x	y		X0	y0	d	sum
2	0	=A2*COS(A2)	=A2*SIN(A2)		0	0	=CYMMKB(B2-E\$2;C2-F\$2)^0,5	=CYMM(G2:G102)
3	0,1	=A3*COS(A3)	=A3*SIN(A3)				=CYMMKB(B3-E\$2;C3-F\$2)^0,5	
4	0,2	=A4*COS(A4)	=A4*SIN(A4)				=CYMMKB(B4-E\$2;C4-F\$2)^0,5	
5	0,3	=A5*COS(A5)	=A5*SIN(A5)				=CYMMKB(B5-E\$2;C5-F\$2)^0,5	
6	0,4	=A6*COS(A6)	=A6*SIN(A6)				=CYMMKB(B6-E\$2;C6-F\$2)^0,5	
7	0,5	=A7*COS(A7)	=A7*SIN(A7)				=CYMMKB(B7-E\$2;C7-F\$2)^0,5	
8	0,6	=A8*COS(A8)	=A8*SIN(A8)				=CYMMKB(B8-E\$2;C8-F\$2)^0,5	
9	0,7	=A9*COS(A9)	=A9*SIN(A9)				=CYMMKB(B9-E\$2;C9-F\$2)^0,5	

Рис. 10. Подготовка данных для **Поиска решения** в задаче 8 (фрагмент рабочего листа в режиме показа формул)

Решение.

На рисунке 9 представлен возможный вариант построения точечного графика спирали (тип диаграммы — точечная, с использованием данных столбцов B и C).

В качестве начального приближительного значения центра кластера далее установим в E2:F2 произвольную точку (0, 0) и введем требуемые расчетные формулы (рис. 10), служащие для расчета расстояния Евклида между каждой из точек и предполагаемым центром кластера (столбец G), а также формулу для нахождения суммы данных расстояний (ячейка H2).

Далее вызываем надстройку **Поиск решения** (**Данные, Поиск решения**). Заполняем параметры: устанавливаем целевую ячейку H2 как равную **минимальному** значению, изменяя ячейки E2:F2.

Полученные в итоге значения центра кластера: (−0,3795, 0,7122).

Надо заметить, что данная задача создана по мотивам задачи 27 ЕГЭ по информатике (в версии демоварианта 2025 года) и позволит учащимся наглядно визуализировать элементы кластерного анализа.

Разработанные автором задачи 1–7 успешно апробировались на олимпиаде по основам информационных технологий, проводимой Башкирским государственным аграрным университетом для студентов-первокурсни-

ков. Задача 8 рассматривалась на занятиях в ходе лабораторных работ у студентов направления подготовки «Бизнес-информатика».

4. Заключение

Обобщая все вышесказанное, можем сделать следующие выводы:

- аналитическая компетенция представляет собой совокупность знаний, умений и навыков, позволяющих эффективно анализировать информацию, систематизировать и структурировать данные, распознавать причинно-следственные связи, делать обоснованные выводы и др. При этом можно выделить такие компоненты аналитической компетенции, как абстрактно-логический, критический, системный, проблемный, образный, алгоритмический, творческий, ассоциативный, рефлексивный;
- формирование аналитического мышления у учащихся является комплексной составной проблемой, включающей в себя повышение абстрактности мышления, развитие логики, формирование творческого подхода к решению задач и др., причем развитие аналитического мышления не ограничивается лишь уроками информатики и ма-

тематики, это общее направление, которое должно поддерживаться в любых учебных предметах;

- табличные процессоры представляют собой мощные, эффективные инструменты, способствующие развитию аналитических навыков у обучающихся, позволяющие не только обрабатывать и визуализировать данные, но и структурировать информацию, что делает электронные таблицы незаменимым ресурсом в современном образовательном процессе. Решение олимпиадных задач с использованием табличных процессоров представляет собой эффективный метод, который может существенно обогатить процесс обучения, способствуя формированию аналитической компетенции у обучающихся, стимулируя их интерес к информатике.

Список источников

1. *Александрова Н. А., Тимонин А. Н.* Развитие аналитического мышления обучающихся на уроках информатики средствами имитационного моделирования // Информатика в школе. 2022. № 2 (175). С. 18–27. EDN: SLMVSE. DOI: 10.32517/2221-1993-2022-21-2-18-27.
2. *Бочкарева О. В., Скурлатов В. В., Царапкина Ю. М., Миронов А. Г., Литке С. Г.* Методические аспекты обучения решению олимпиадных задач средствами электронных таблиц // Мир науки. Педагогика и психология. 2023. Т. 11. № 3. EDN: ELCQTK.
3. *Голякова В. А.* Содержание программы подготовки команды курсантов к международной олимпиаде по информатике // Педагогический журнал. 2024. Т. 14. № 4-1. С. 473–480. EDN: GNXIBD.
4. *Жилина Н. Д., Таренко Л. Б.* Особенности формирования аналитических умений у будущих специалистов в области информационных технологий // Образовательные технологии и общество. 2016. Т. 19. № 4. С. 364–376. EDN: WWKQLD.
5. *Корнилов П. А., Федотова Н. П.* Олимпиада по информатике и информационным технологиям для абитуриентов ЯГПУ // Информатика в школе. 2022. № 5 (178). С. 39–44. EDN: TTCTDQ. DOI: 10.32517/2221-1993-2022-21-5-39-44.
6. *Осипова В. В., Бугаева Т. П., Осипов В. А.* Развитие аналитического мышления магистрантов в проектной деятельности // Современные наукоемкие технологии. 2022. № 9. С. 174–179. EDN: WTPTPV. DOI: 10.17513/snt.39328.
7. *Саенко Л. А., Соломатина Г. Н.* Аналитические умения студентов: сущностные характеристики, уровни развития, факторы влияния // Научно-педагогическое обозрение. 2021. № 4 (38). С. 68–75. EDN: KKYSMY. DOI: 10.23951/2307-6127-2021-4-68-75.
8. *Сергунцова Е. В., Михоненко О. И., Зритнева Е. И.* Аналитические умения как базовый элемент развития парадоксального и творческого мышления // Проблемы современного педагогического образования. 2023. № 80-2. С. 301–304. EDN: QEEZLY.
9. *Скрипкина А. В., Дзюба Ю. Ю.* Сущностные характеристики развития информационно-аналитической компетентности будущих специалистов социэкономической сферы в процессе обучения в вузе // Вестник Краснодарского государственного института культуры. 2016. № 4 (8). С. 5. EDN: ZDPPKL.
10. *Шамсутдинова Т. М.* Развитие навыков визуализации и визуальной аналитики в курсе информатики // Информатика и образование. 2023. Т. 38. № 3. С. 16–23. EDN: FOBBVX. DOI: 10.32517/0234-0453-2023-38-3-16-23.
11. *Ярыгин О. Н., Рябова В. М.* Неявное знание как компонент компетентности в аналитической деятельности // Балтийский гуманитарный журнал. 2013. № 4 (5). С. 131–134. EDN: RURAAZ.

DOI: 10.32517/2221-1993-2025-24-1-26-38

**Н. В. Климина***победитель конкурса ИНФО-2024 в номинации**«40 лет школьной информатике: Дополнительное образование в области информатики и информационных технологий»,
Средняя общеобразовательная школа № 6 имени Героя Советского Союза В. Н. Банцеева,
г. о. Сызрань, Самарская область, Россия*

ВОКСЕЛЬНОЕ МОДЕЛИРОВАНИЕ НА БАЗЕ СРЕДЫ МОДЕЛИРОВАНИЯ MAGICAVOXEL*

Аннотация

В статье приводится обоснование необходимости знакомства школьников с технологией воксельного моделирования, описывается возможность применения среды воксельного моделирования MagicaVoxel для обучения основам воксельного моделирования школьников на занятиях внеурочной деятельности в школе и на занятиях детских технопарков «Кванториум», приведено описание нескольких авторских практических работ по воксельному моделированию, охватывающих наиболее применяемые инструменты среды моделирования. Методические разработки в виде практических работ будут полезны учителям, преподающим моделирование в школе, для проведения занятий с использованием среды воксельного моделирования MagicaVoxel со II по XI класс. Обучение основам воксельного моделирования на базе данной методической разработки будет способствовать формированию у учащихся навыков трехмерного моделирования вокселями, умения применять информационные технологии для решения творческих задач.

Ключевые слова: моделирование, воксельное моделирование, воксель, шейдер, рендеринг, модель, форма, MagicaVoxel.

Информация об авторе

Климина Наталья Владимировна, учитель информатики, средняя общеобразовательная школа № 6 имени Героя Советского Союза В. Н. Банцеева, г.о. Сызрань, Самарская область, Россия; *e-mail*: natklimina@yandex.ru

1. Введение

Воксельное моделирование — это воскрешение забытого подхода в создании трехмерных объектов и пространств, который основывается на использовании **вокселей** — трехмерных аналогов пикселей. Каждый воксель представляет собой кубическую единицу объема, что позволяет детализировать модели на уровне, недоступном при традиционном полигональном моделировании. Это особенно актуально в таких областях, как архитектура, игры и научные визуализации, где важно передать сложные структуры и формы. Незаслуженно забытым данный перспективный метод представления

моделей оказался из-за небольших аппаратно-вычислительных мощностей компьютеров в 90-х годах прошлого столетия. В играх того времени вокселизировали ландшафты или малые объекты, которые планировалось реалистично разрушить. Воксельная модель требовала много ресурсов памяти, и выгоднее было использовать менее требовательные к ресурсам полигональные модели. В настоящее время возможности компьютерной техники позволяют создавать частично вокселизированные мобильные игры и компьютерные игры. Пять лет назад появилась первая полностью воксельная компьютерная игра, а десять лет назад воксели стали применяться в киноиндустрии. Цифровые художники

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_01-2025/

N. V. Klimina

School 6 named after Hero of the Soviet Union V. N. Bantsekin, Syzran, Samara Region, Russia

VOXEL MODELING BASED ON THE MAGICAVOXEL MODELING ENVIRONMENT

Abstract

The article provides a rationale for introducing schoolchildren to voxel modeling technology, describes the possibility of using the MagicaVoxel modeling environment to teach schoolchildren the basics of voxel modeling during extracurricular activities at school and during classes at the Quantorium children's technology parks, and describes several original practical works on voxel modeling, covering the most commonly used tools of the modeling environment. Methodological developments in the form of practical works will be useful for teachers teaching modeling at school, for conducting classes using the MagicaVoxel voxel modeling environment from 2nd to 11th grade. Teaching the basics of voxel modeling based on this methodological development will help students develop skills in three-dimensional voxel modeling, and the ability to apply information technology to solve creative problems.

Keywords: modeling, voxel modeling, voxel, shader, rendering, model, shape, MagicaVoxel.

Information about the author

Natalia V. Klimina, informatics teacher, School 6 named after Hero of the Soviet Union V. N. Bantsekin, Syzran, Samara Region, Russia; *e-mail*: natklimina@yandex.ru

современности полюбили воксель-арт за приятную эстетичность созданных моделей. Поэтому в последнее время технология воксельной графики стала невероятно популярной у дизайнеров, цифровых художников и просто творческих людей [1, 7–10, 14, 19].

У данной цифровой технологии колоссальное будущее, связанное с виртуальной реальностью. Развитием воксельной графики будут заниматься сегодняшние школьники. Задача учителя — обеспечить вхождение будущих моделлеров в техническое творчество, развивать интеллектуальную одаренность детей, формировать их интерес в сфере перспективного трехмерного моделирования и заинтересованность в профессиях, связанных с решением графических задач. Таким образом, обучение основам воксельного моделирования наряду с обучением полигональному моделированию является необходимым для формирования личности высокотехнологического общества [3, 5, 11, 12].

Для полноценного изучения воксельного моделирования рекомендовано выделение часов внеурочной деятельности или проведение занятий в технопарках «Кванториум».

Среди представителей сред воксельного моделирования наиболее популярным является Open Source редактор MagicaVoxel. Программа портативна и маловесна, но обладает внушительным функционалом [17, 18]. Скачать архив с дистрибутивом можно по ссылке:

<https://github.com/ephtracy/ephtracy.github.io/releases/tag/0.99.7>

2. Интерфейс и инструменты среды воксельного моделирования MagicaVoxel

Интерфейс среды воксельного моделирования MagicaVoxel разбит на пять вертикальных областей, слева направо в порядке следования (рис. 1):

- палитра;
- панель кистей;
- область Мира с областью моделирования;
- панель редактирования;
- панель работы с проектом.

Рекомендуемые основные навыки, которые необходимо формировать у школьников при создании трехмерных воксельных моделей:

- создание модели из отдельных вокселей (использовать воксельный режим);
- создание модели из плоской и объемной групп вокселей (использовать режим «Геометрия» и воксельный режим);
- создание и редактирование модели, используя режим экструдирования и режим box-моделирования;
- заливка модели цветом: заливка отдельных вокселей, отдельных областей вокселей, заливка всей

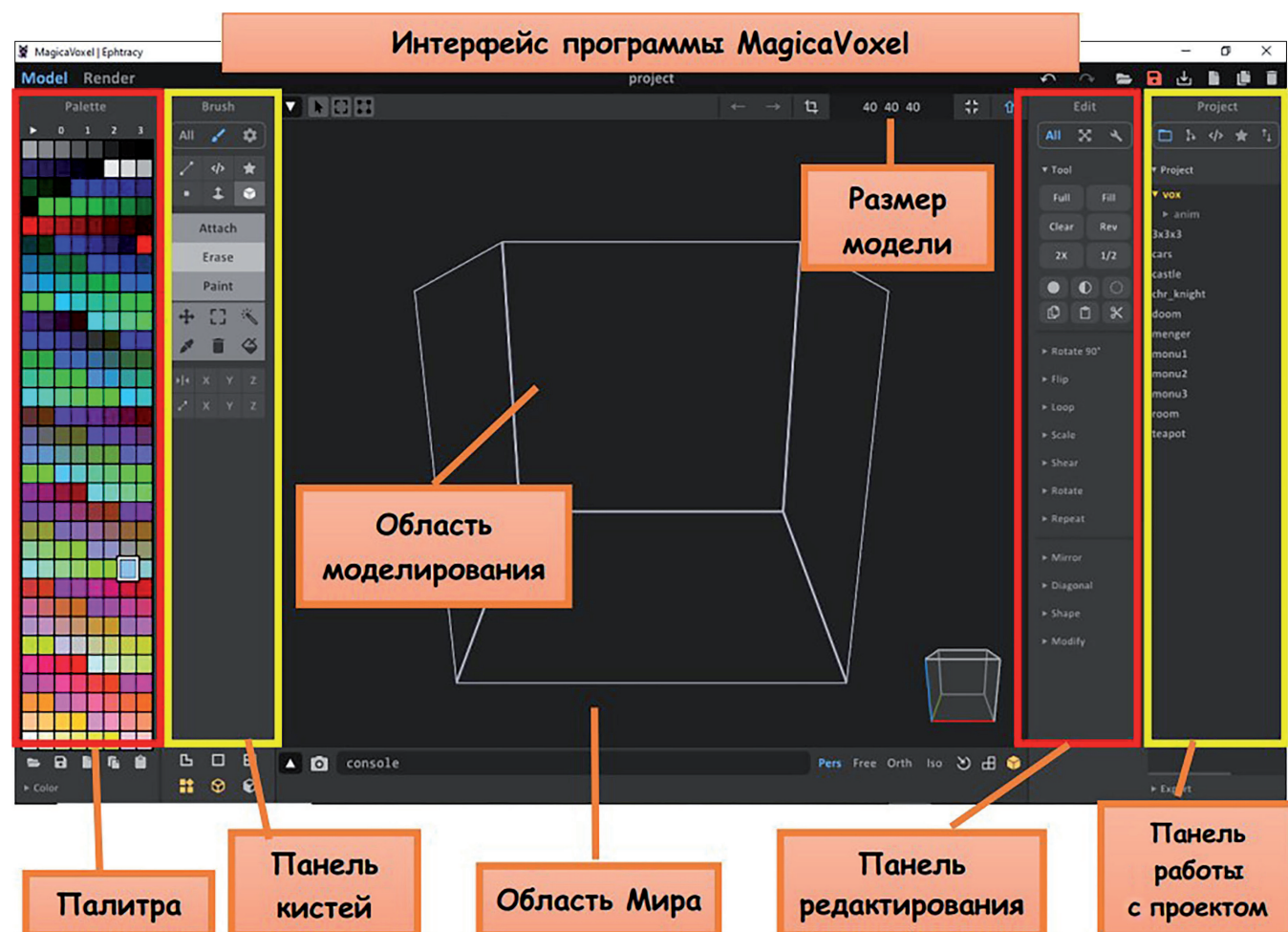


Рис. 1. Интерфейс среды моделирования MagicaVoxel

модели, случайная заливка с использованием нескольких цветов;

- масштабирование моделей;
- отражение модели по осям координат и зеркальное отражение;
- сдвиг модели по осям координат;
- создание моделей с использованием форм;
- наложение материалов и настройка их свойств.

Изучение принципов работы основных инструментов среды моделирования позволит сформировать описанные навыки [1, 14, 18].

Рассмотрим содержание панели кистей.

На рисунке 2 представлены режимы моделирования:

- режим «Геометрия»;
- режим box-моделирования;
- режим экструдирования;
- воксельный режим.

Ниже режимов моделирования идут кисти (рис. 3):

- кисть «Присоединение»;
- кисть «Удаление вокселей»;
- кисть «Раскрашивание».

Ниже кистей расположены режимы «Зеркальная симметрия по осям координат» и «Заполнение вокселями по осям координат» (рис. 4).

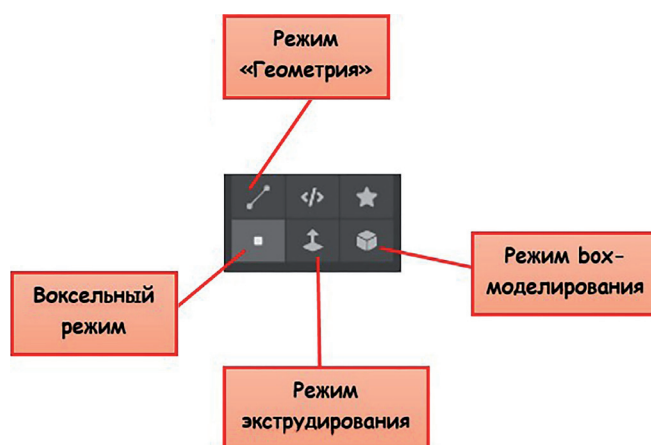


Рис. 2. Режимы моделирования

Для режима «Геометрия» доступна панель свойств. Можно выбрать опцию моделирования линией, квадратом или кругом (рис. 5).

Для воксельного режима доступна панель свойств. Можно выбрать опцию моделирования двумерными или трехмерными группами вокселей, квадратной или круглой формы и число вокселей в группе (рис. 6).

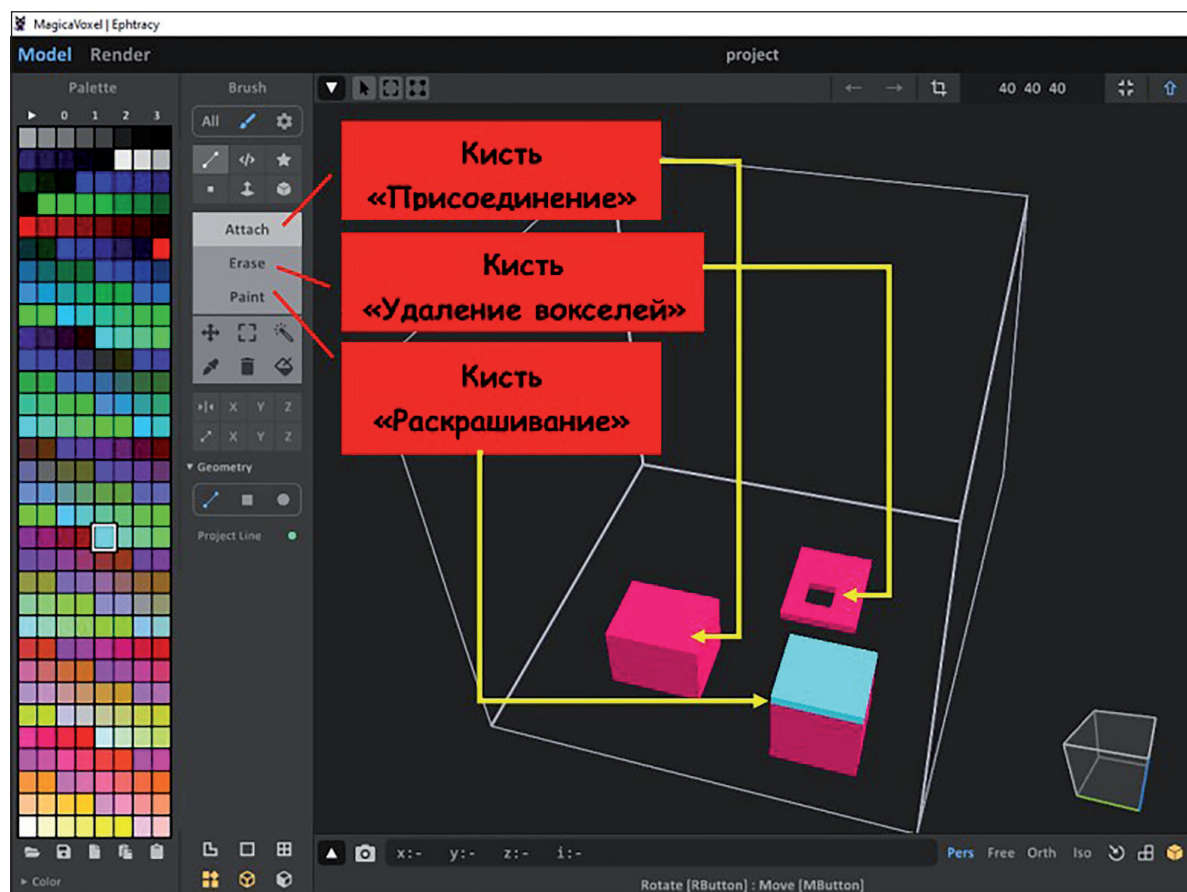


Рис. 3. Кисти



Рис. 4. Режимы зеркальной симметрии и заполнения

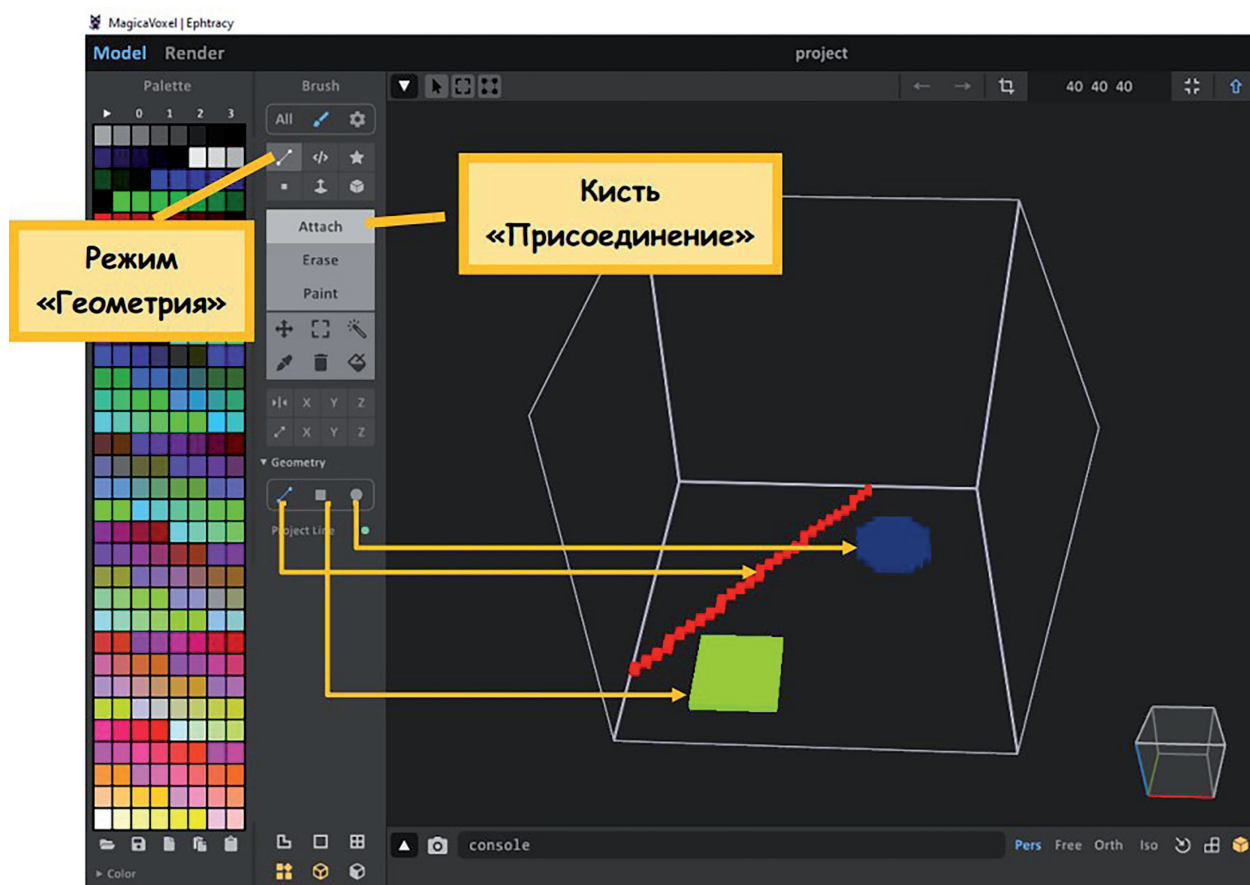


Рис. 5. Панель свойств режима «Геометрия»

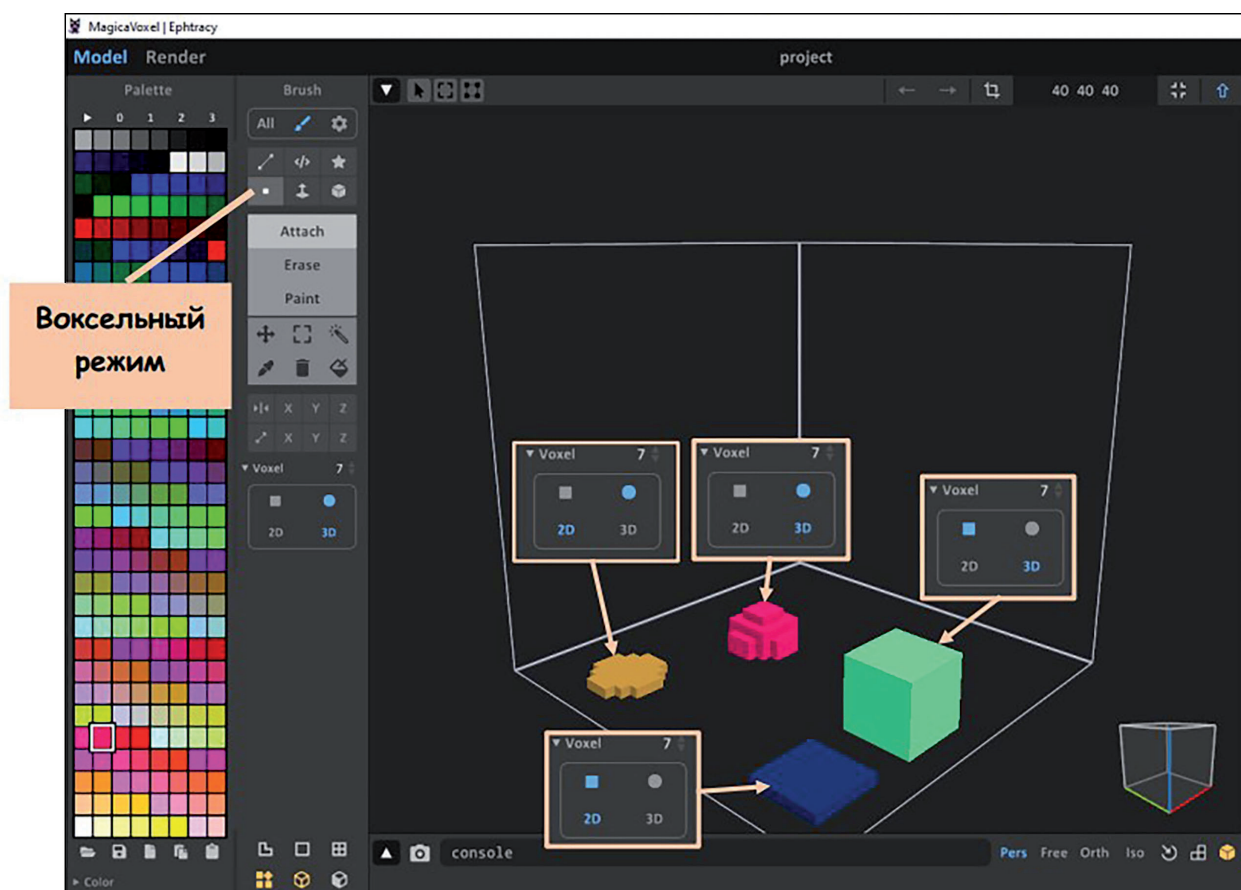


Рис. 6. Панель свойств воксельного режима

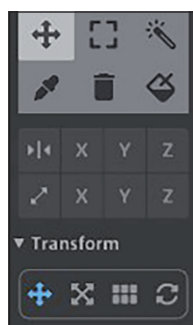


Рис. 7. Инструменты выделения и трансформации

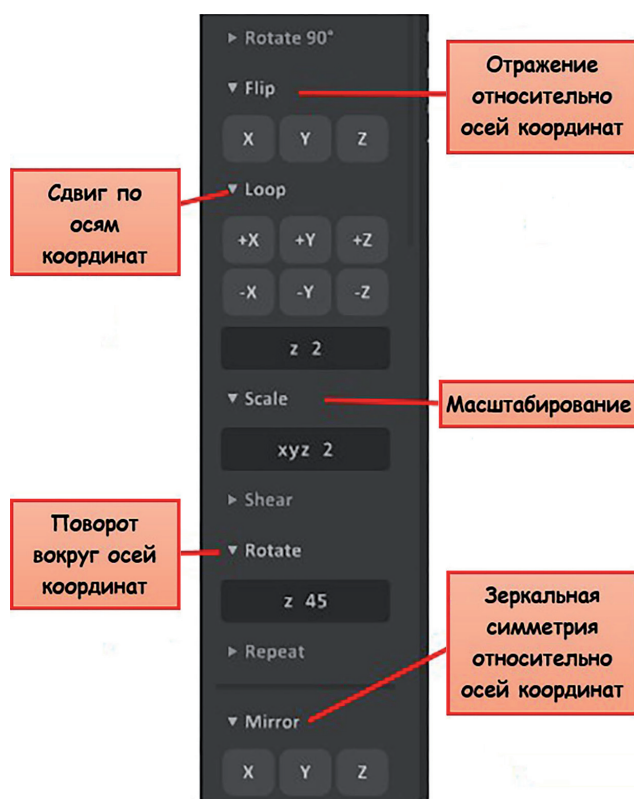
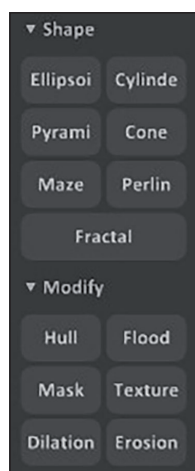


Рис. 8. Инструменты редактирования

Рис. 9. Инструменты редактирования **Формы** и **Модификация**

Для перемещения и выделения модели или ее части представлены инструменты выделения и трансформации. В этой же панели представлены инструменты заливки и удаления выделенной области (рис. 7).

В панели редактирования сверху вниз представлены инструменты редактирования модели (рис. 8).

На той же панели редактирования внизу расположены инструменты **Формы** и **Модификация** (рис. 9).

Еще понадобятся шейдеры, которые представляют собой алгоритмы для автоматизации нескольких действий. Обычно шейдеры для MagicaVoxel пишут на языке программирования C++. Шейдеры можно скачать по ссылке:

<https://github.com/lachlanmcdonald/magicavoxel-shaders/wiki> и в распакованном виде поместить в папку shader [6, 7].

3. Примеры упражнений для отработки навыков моделирования

3.1. «Трасса в пустыне»

Цель: применение режима box-моделирования и экструдирования, кисти заливки вокселей, кисти удаления вокселей и кисти «Присоединение» при создании модели [4, 11, 16].

Ход моделирования:

1. Для очищения зоны моделирования выберем инструмент *Clear* на панели редактирования или нажмем клавишу Del (Delete) на клавиатуре. Выберем цвет песка пустыни в палитре. Создадим область пустыни, используя режим box-моделирования и экструдирования на произвольную высоту и кисть «Присоединение» (рис. 10, 11).
2. Наметим место для трассы, используя режим box-моделирования и кисть удаления вокселей (рис. 12).
3. Заполним подготовленное место для трассы черными вокселями, используя режим box-моделирования и кисть «Присоединение» (рис. 13).
4. Изобразим поверхность пустыни, используя режим box-моделирования и кисть удаления вокселей (рис. 14).
5. Нарисуем разметку на трассе белым цветом, используя режим box-моделирования и кисть заливки вокселей (рис. 15).
6. Отрендерим готовую модель в окне режима Render (рис. 16).

3.2. «Горный водопад»

Цель: применение воксельного режима, режима «Геометрия», материала Glass при создании модели [2, 6, 13, 15].

Ход моделирования:

1. Установим размер области создания модели: 200 256 256. Выберем режим «Геометрия», опцию *круг*. Выберем кисть «Присоединение» и создадим однослойную водную поверхность цветом предполагаемой воды (рис. 17).

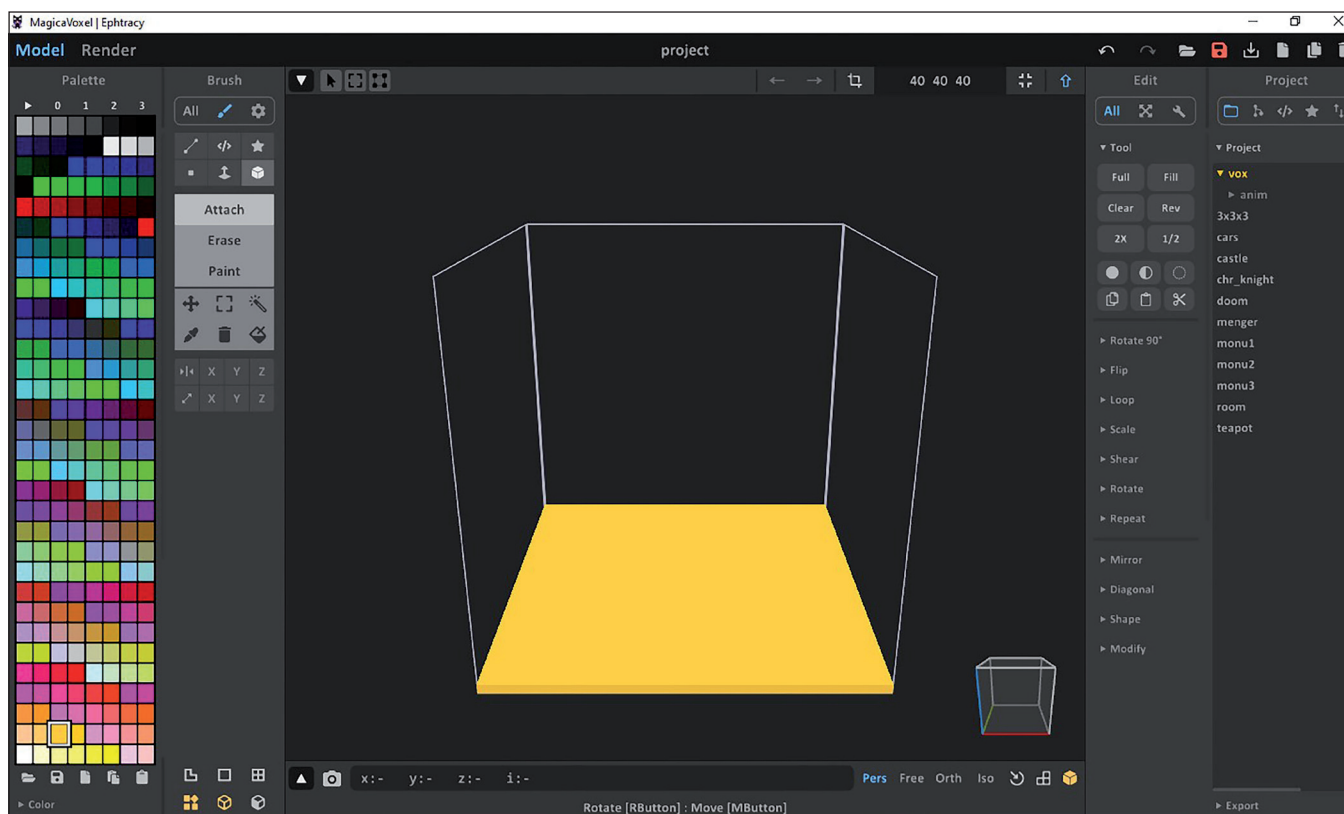


Рис. 10. Решение задачи 3.1. Создание области пустыни.
Применение режима vox-моделирования и кисти «Присоединение» (шаг 1)

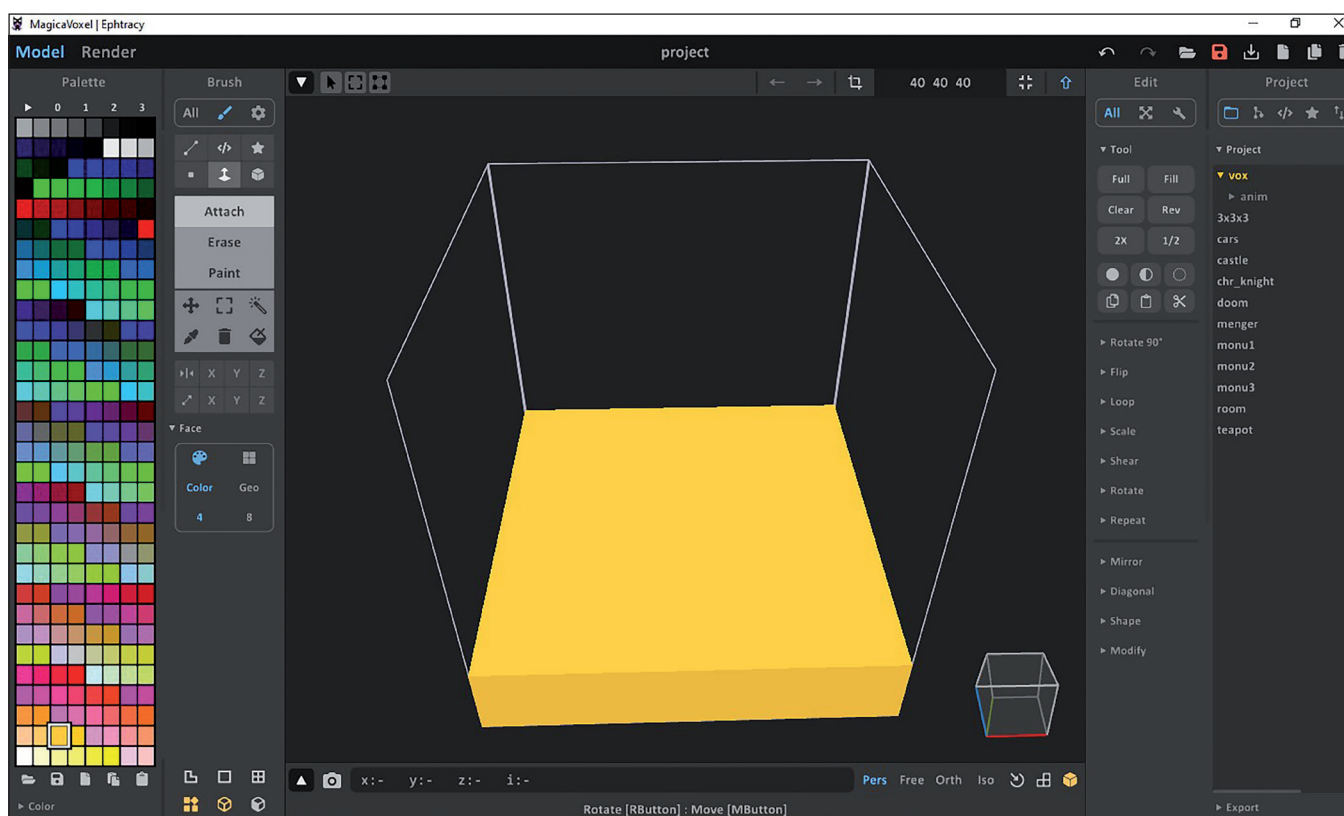


Рис. 11. Решение задачи 3.1. Создание области пустыни.
Применение режима экструдирования и кисти «Присоединение» (шаг 2)

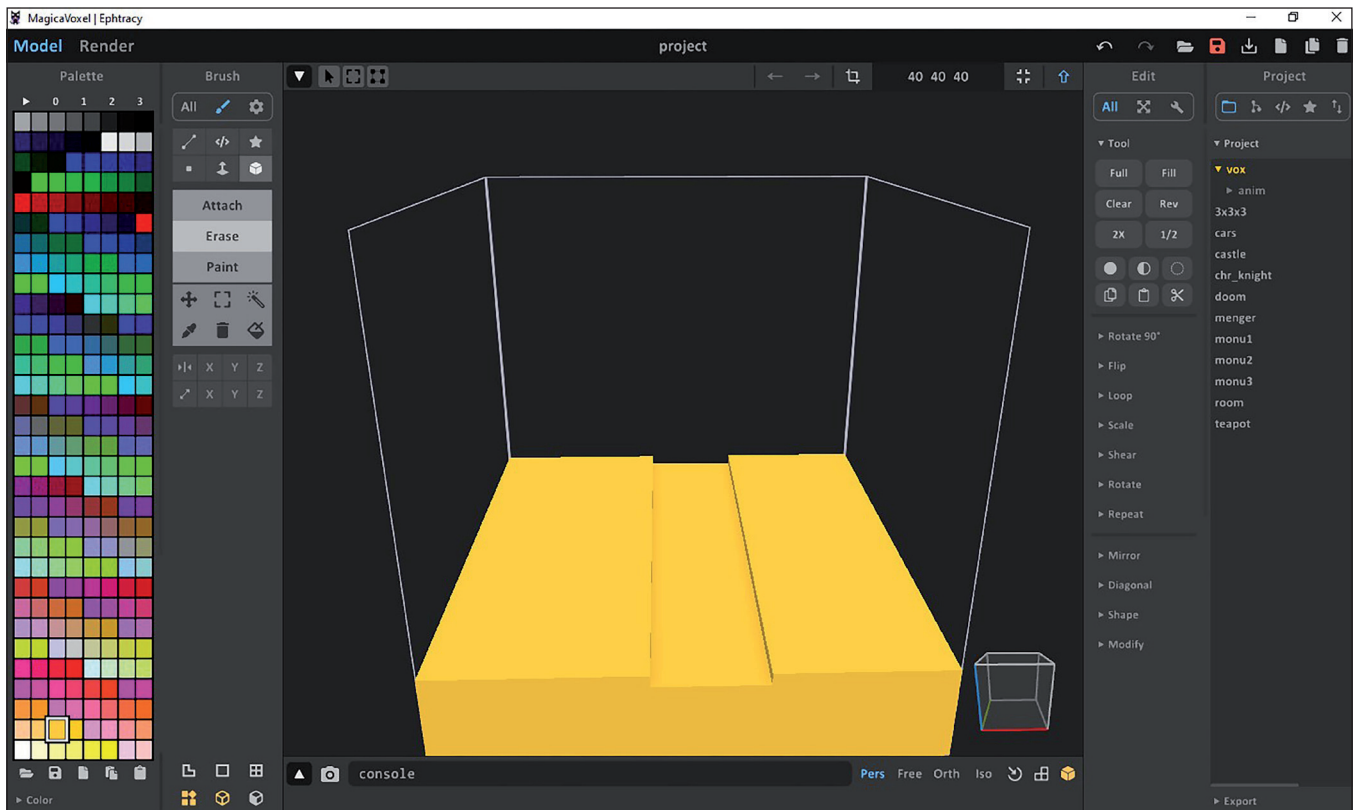


Рис. 12. Решение задачи 3.1. Создания места для трассы.
Применение режима vox-моделирования и кисти удаления вокселей

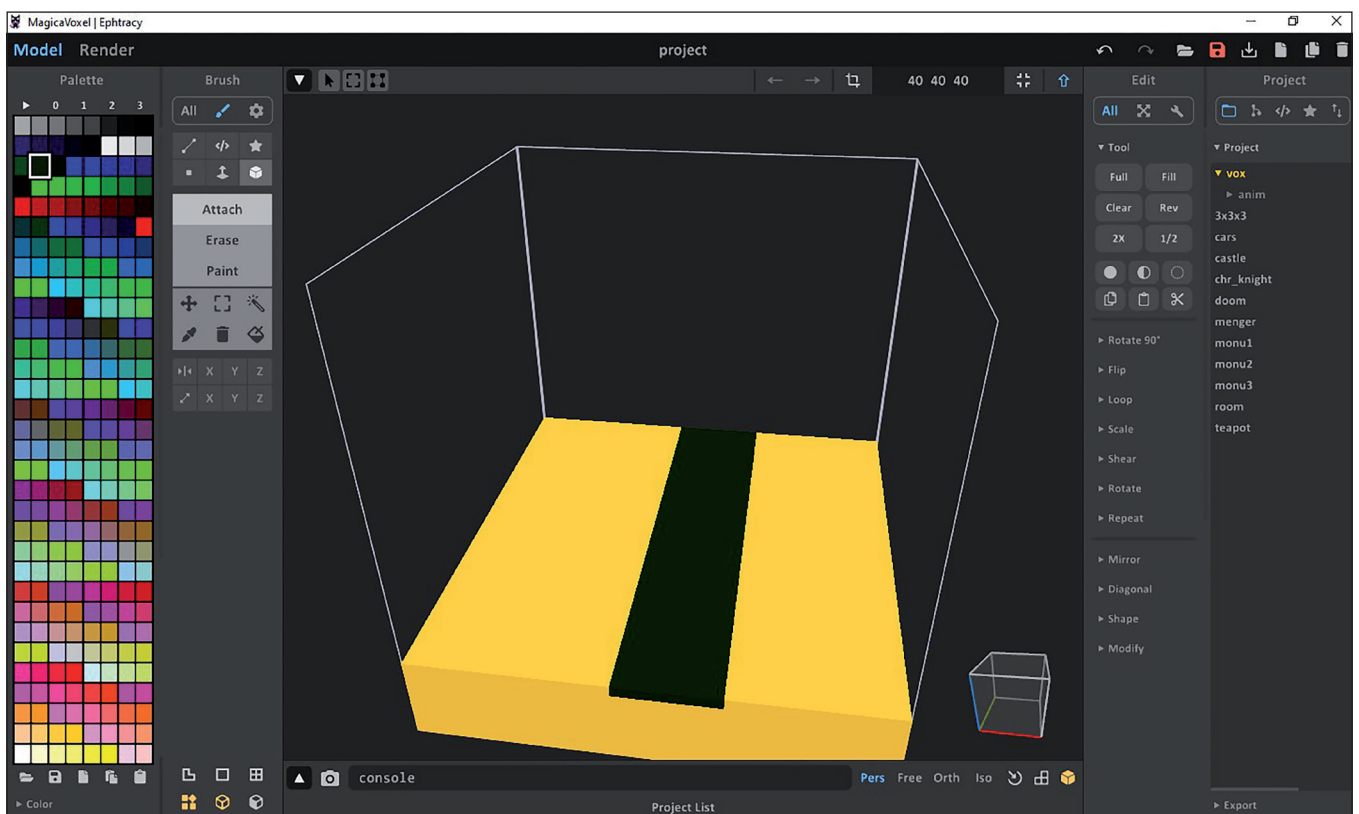


Рис. 13. Решение задачи 3.1. Заполнение места для трассы черными вокселями.
Применение режима vox-моделирования и кисти «Присоединение»

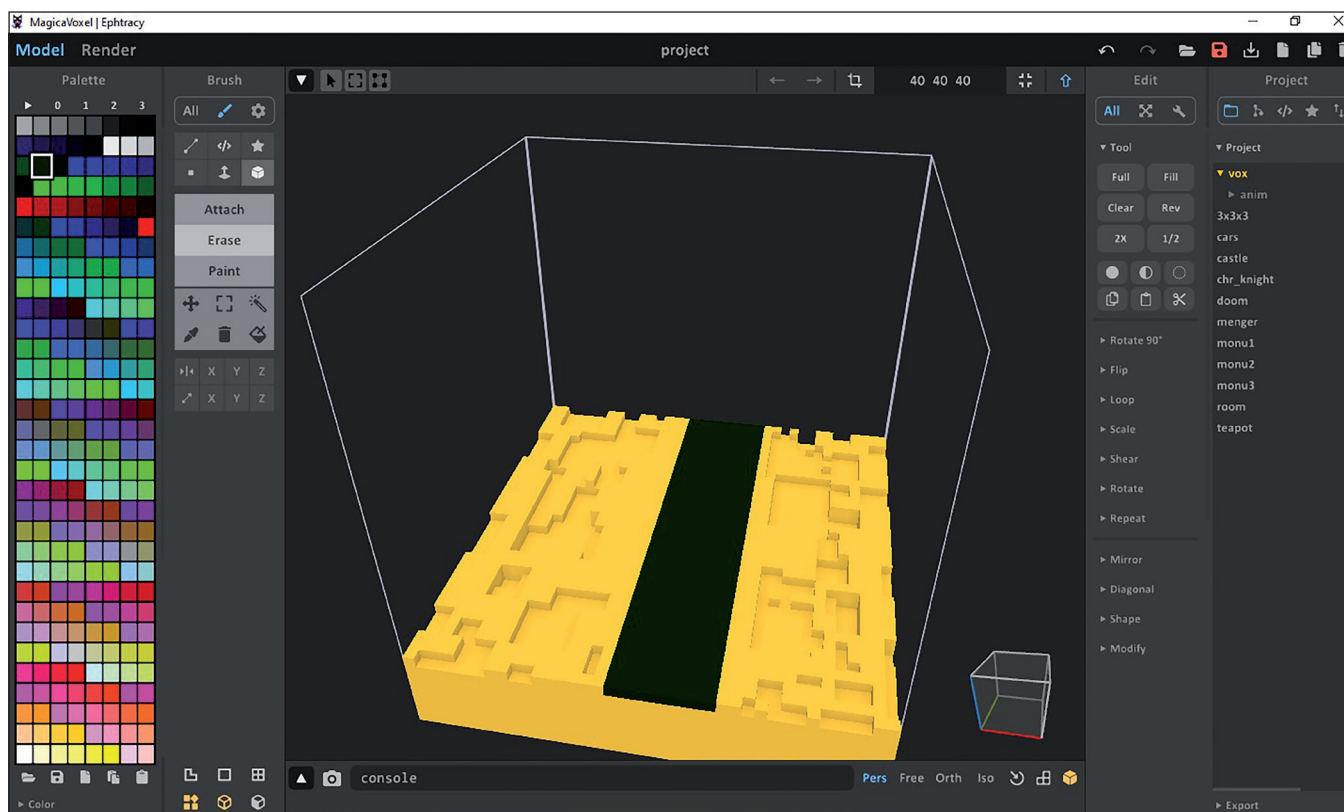


Рис. 14. Решение задачи 3.1. Создание борозд на поверхности пустыни.
Применение режима vox-моделирования и кисти удаления вокселей

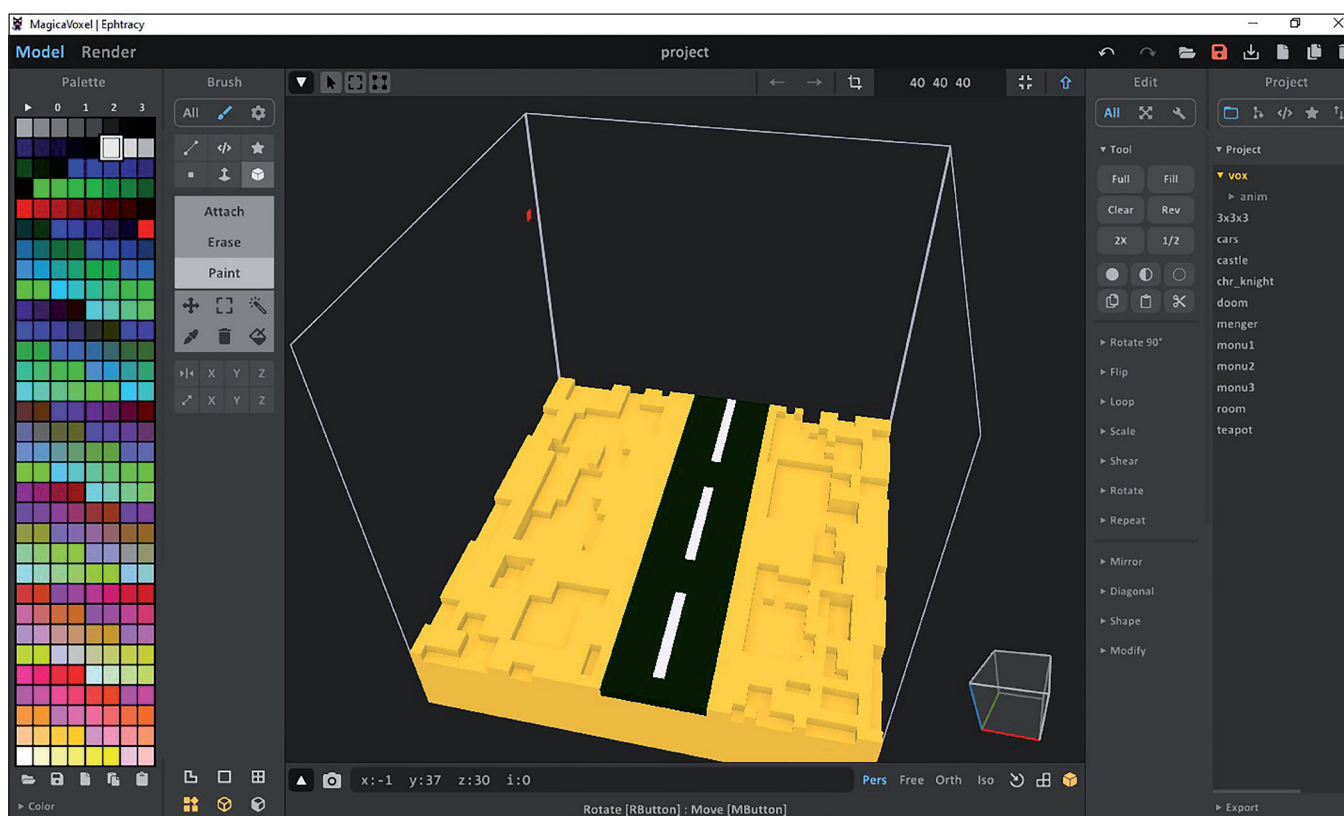


Рис. 15. Решение задачи 3.1. Рисование разметки на трассе.
Применение режима vox-моделирования и кисти заливки вокселей

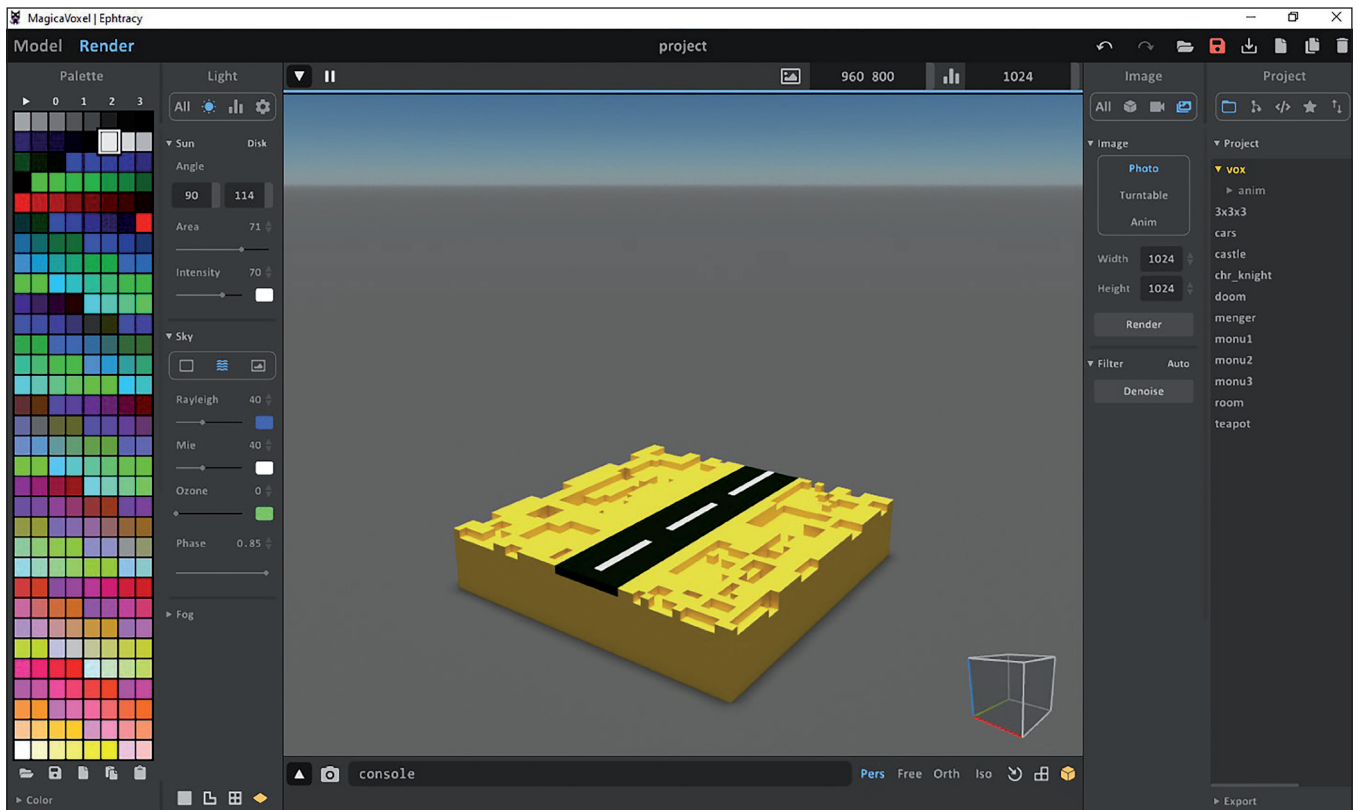
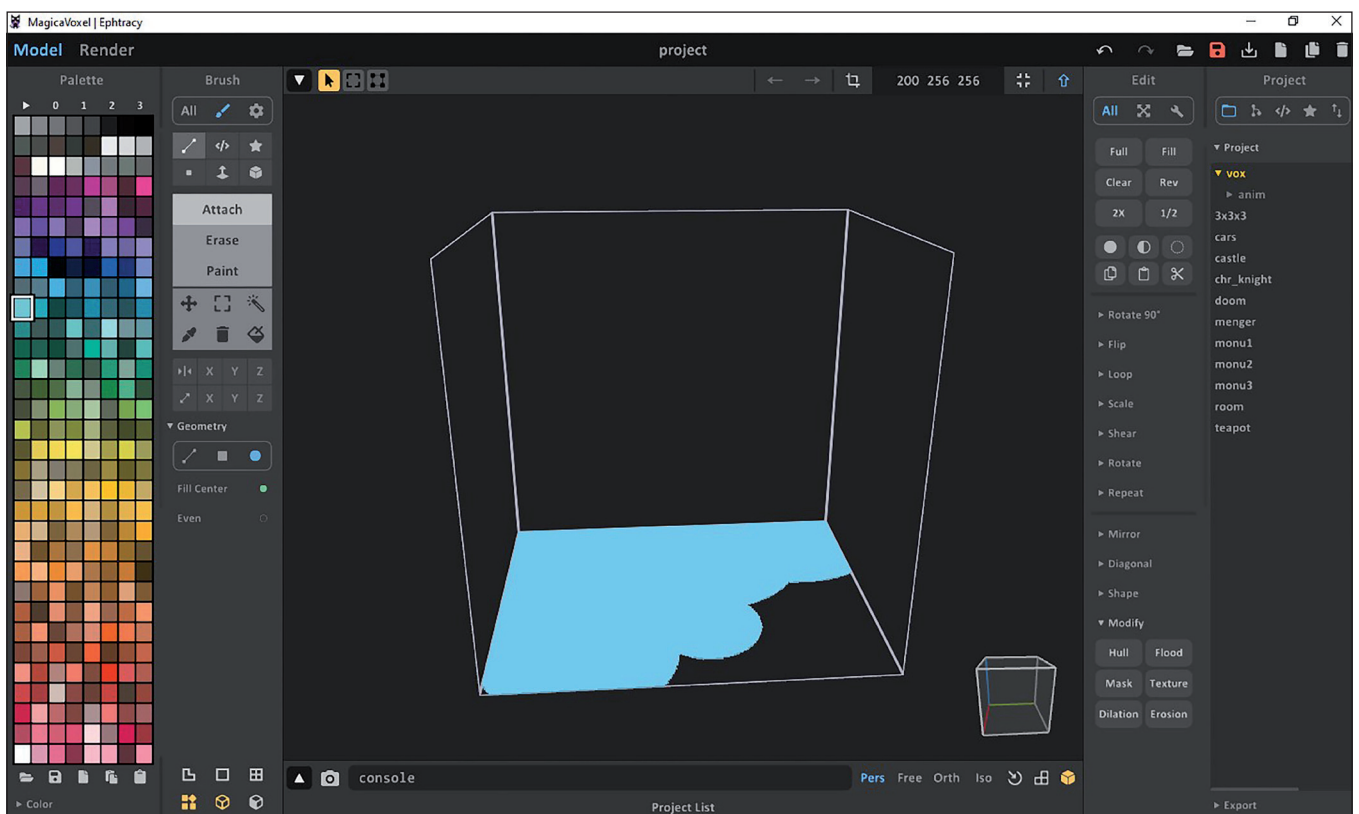


Рис. 16. Решение задачи 3.1. Модель в режиме Render

Рис. 17. Решение задачи 3.2.
Режим «Геометрия». Однослойная водная поверхность

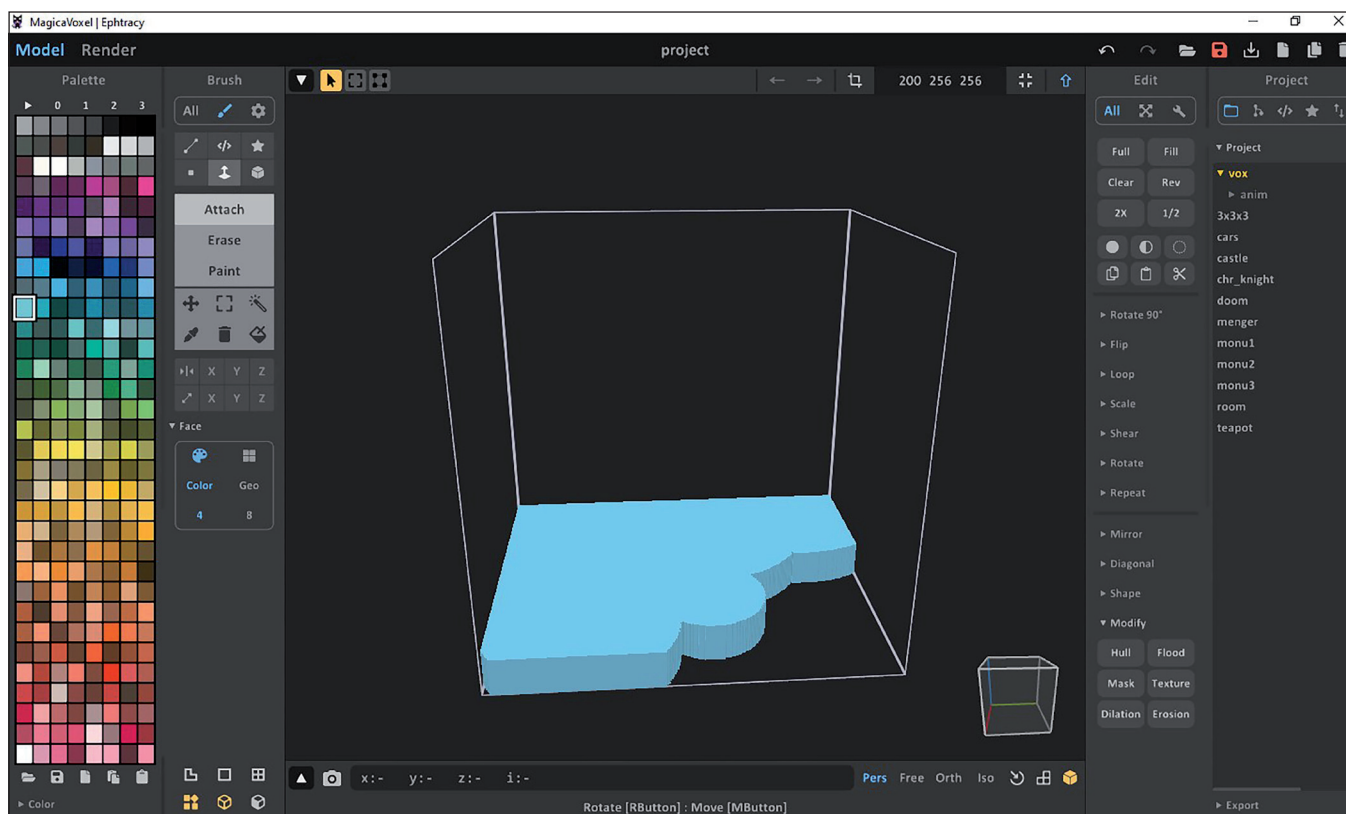


Рис. 18. Решение задачи 3.2. Поднятие уровня воды.
Применение экструдирования

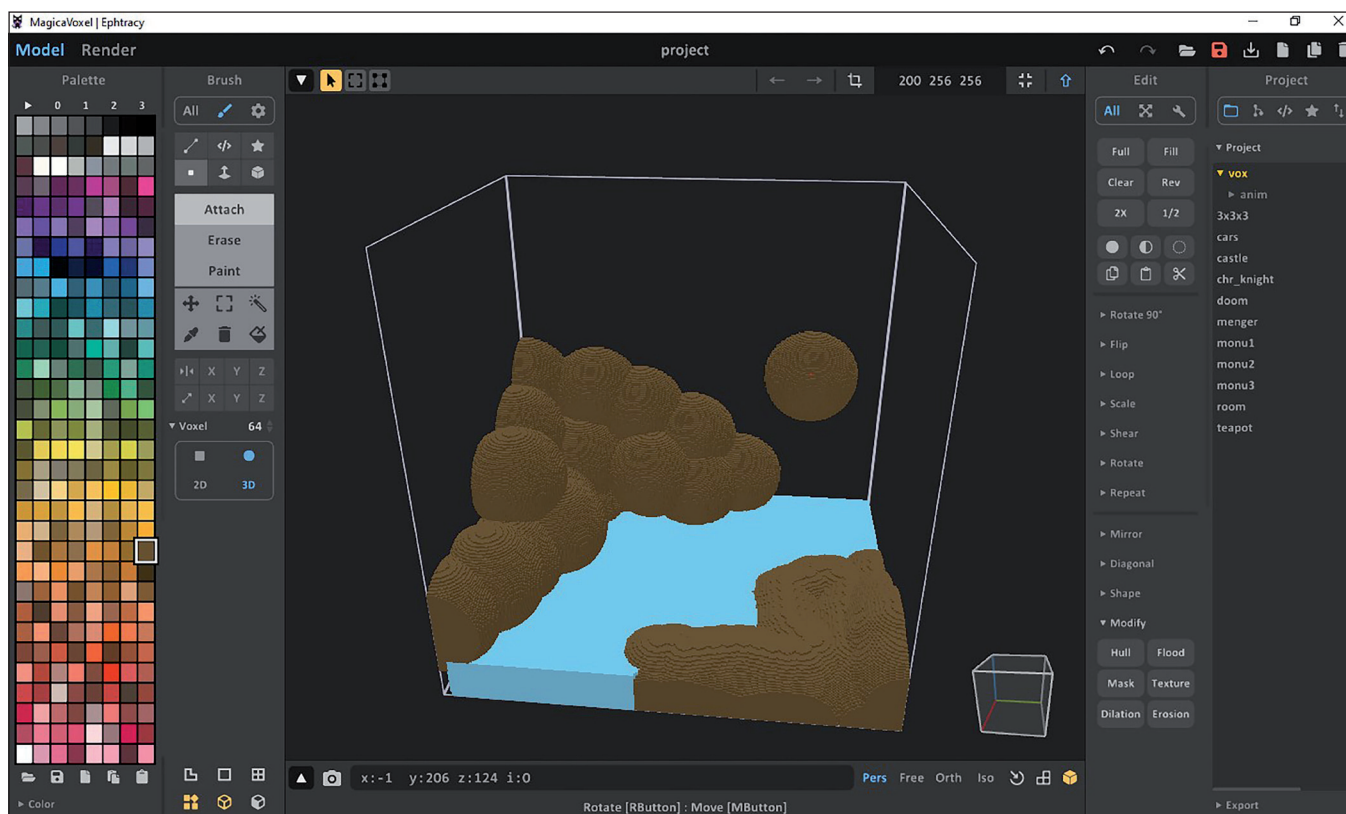


Рис. 19. Решение задачи 3.2. Создание горной поверхности.
Применение воксельного режима с опциями и кисти «Присоединение»

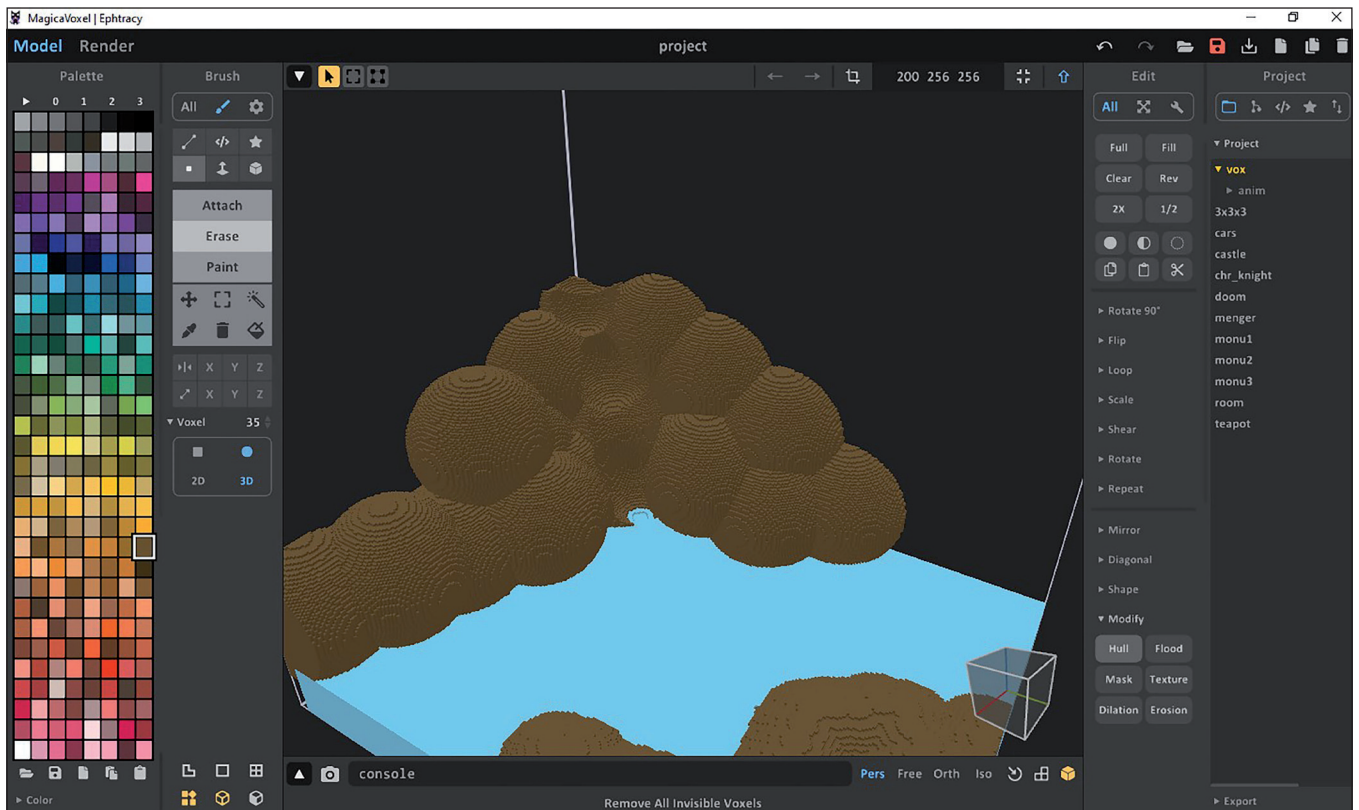


Рис. 20. Решение задачи 3.2. Создание пути воды для водопада.
Применение воксельного режима с опциями и кисти удаления вокселей

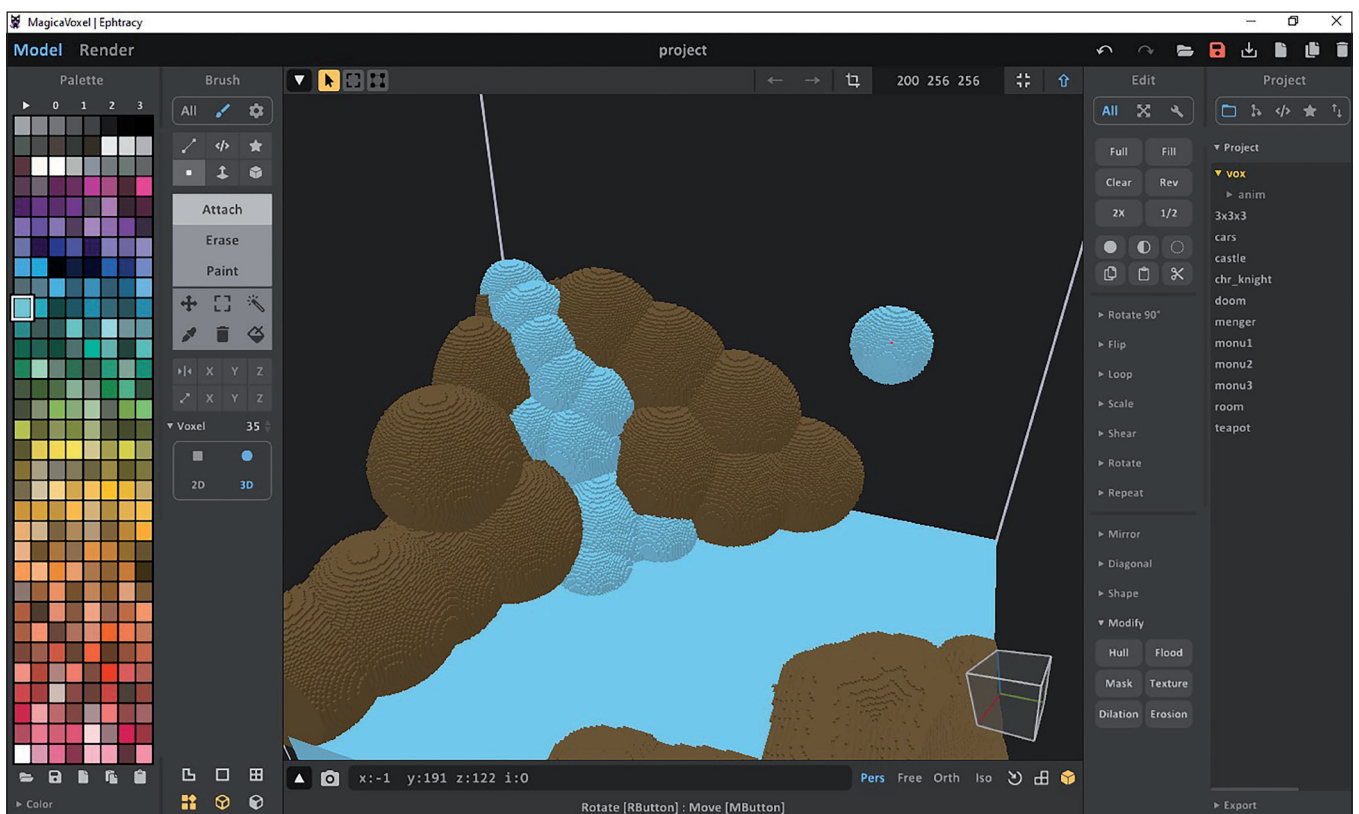


Рис. 21. Решение задачи 3.2. Создание потоков воды для водопада.
Применение воксельного режима с опциями и кисти «Присоединение»

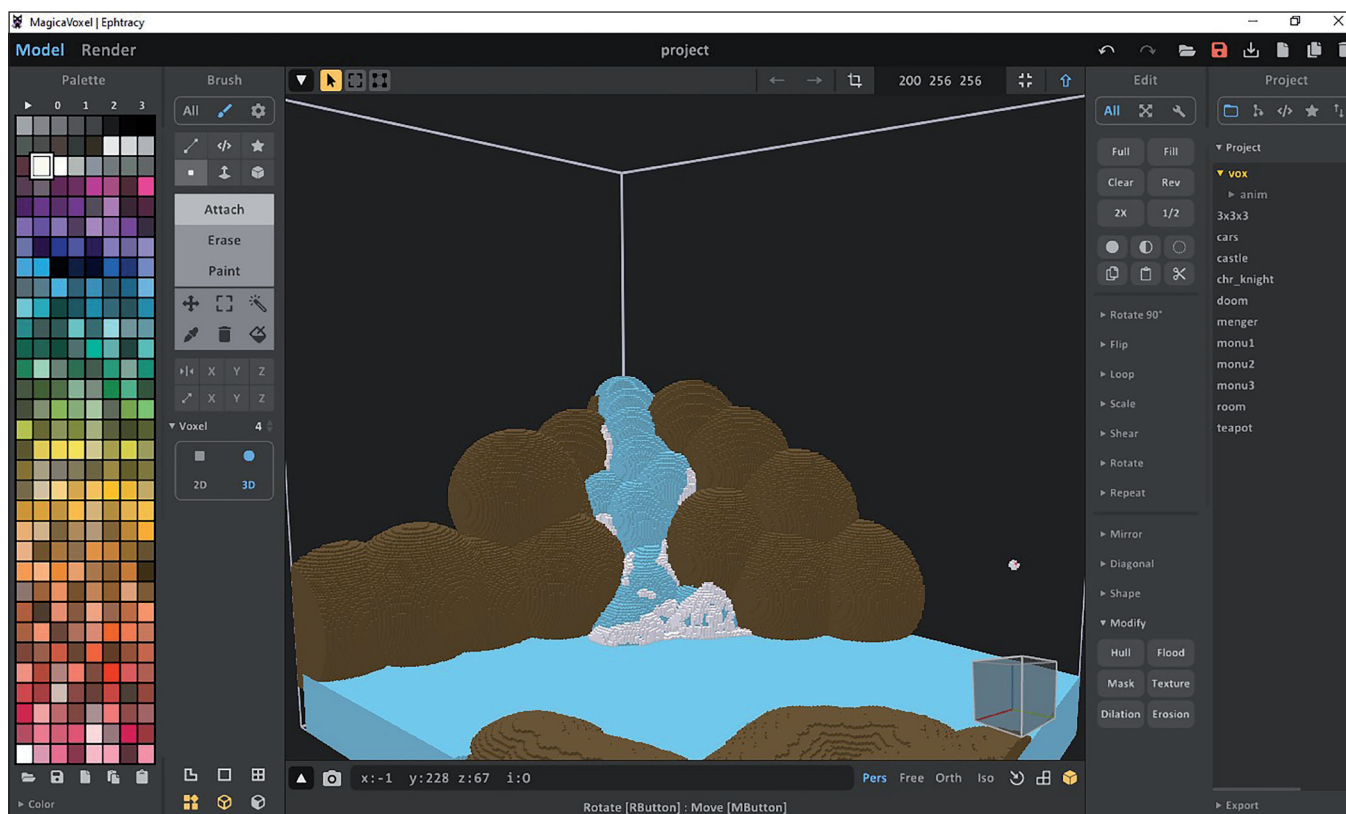


Рис. 22. Решение задачи 3.2. Создание пены белого цвета.
Применение воксельного режима с опциями и кисти «Присоединение»

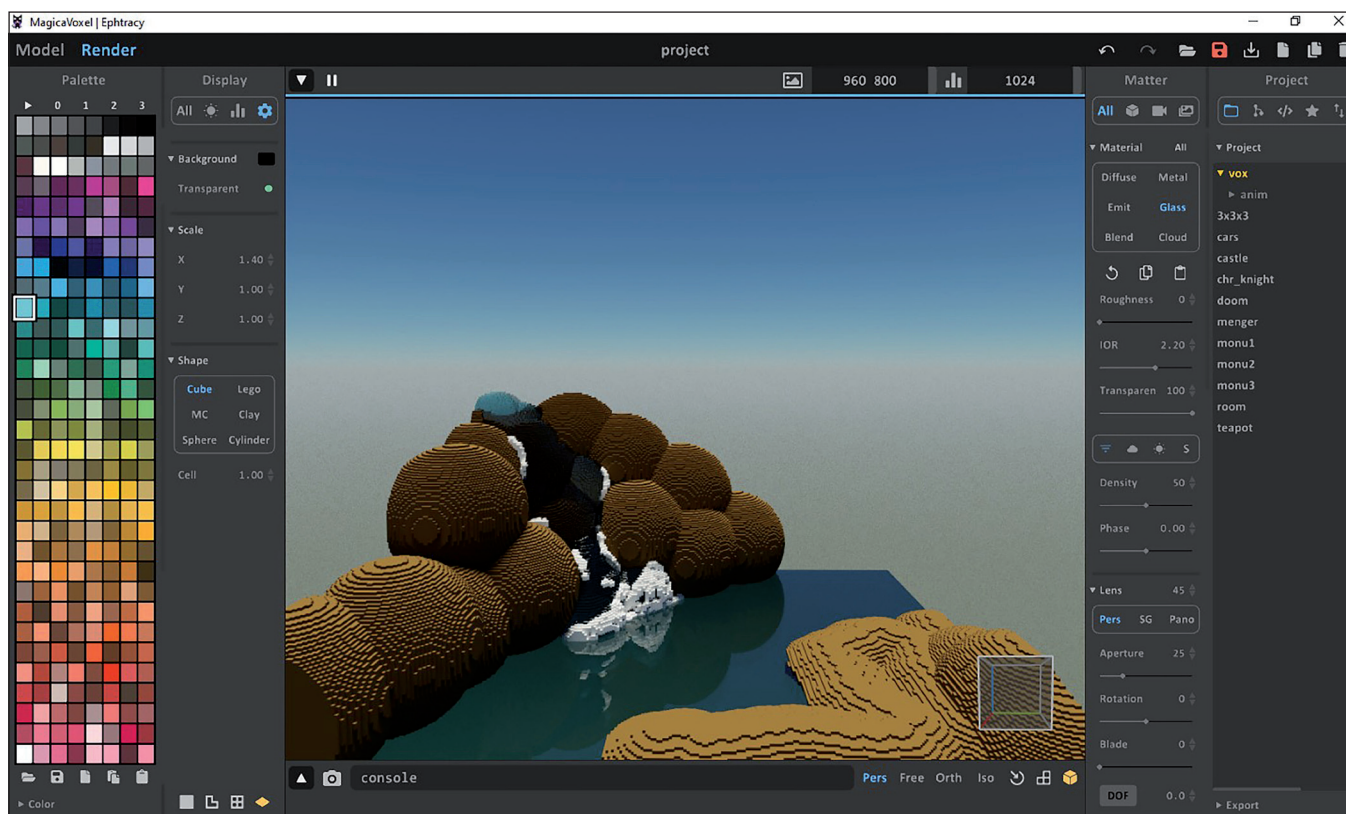


Рис. 23. Решение задачи 3.2. Применение свойств материала Glass в режиме Render

2. С помощью инструмента экструдирования поднимем уровень воды на произвольную высоту (рис. 18).
3. С помощью воксельного режима, опций *Voxel* 64, круг *3D* и кисти «Присоединение» создадим горную поверхность (рис. 19).
4. С помощью воксельного режима, опций *Voxel* 35, круг *3D* и кисти удаления вокселей создадим путь воды для водопада (рис. 20).
5. С помощью воксельного режима, опций *Voxel* 35, круг *3D* и кисти «Присоединение» создадим потоки воды для водопада (рис. 21).
6. Создадим пену белого цвета с помощью воксельного режима, опций *Voxel* 4, круг *3D* и кисти «Присоединение» (рис. 22).
7. Выделим *Волшебной палочкой* воксели воды. В режиме рендеринга в панели *Matter* выберем материал *Glass* и настроим свойства *IOR* и *Transparen* (рис. 23).

Другие варианты упражнений можно скачать на сайте ИНФО:

http://infojournal.ru/journals/school/school_01-2025/

4. Заключение

Трехмерное воксельное моделирование представляет собой творческий процесс создания моделей, которые могут использоваться в играх, в быту, в мультипликации и в других сферах деятельности. Создание воксельных моделей не просто увлекательно для школьников, но и развивает творческое и дизайнерское мышление, пространственное мышление, способствует развитию внимания и формированию навыков конструирования. С помощью среды воксельного моделирования MagicaVoxel создавать трехмерные модели сможет школьник любого возраста. Изучение воксельного моделирования на занятиях внеурочной деятельности или на занятиях технопарков «Кванториум» также будет способствовать профориентации школьников [1, 9].

Список источников

1. Болбаков Р. Г., Цветков В. Я., Мордвинов В. А., Матчин В. Т. Применение иммерсивных технологий в обучении // Информатизация образования и науки. 2023. № 1 (57). С. 19–26. EDN: JGTUZQ.
2. Витиска Н. И., Гуляев Н. А. Метод визуализации трехмерных сцен и объектов воксельной графики для систем имитационного моделирования // Известия ЮФУ. Технические науки. 2015. № 4 (165). С. 77–87. EDN: ТУМОВТ.
3. Воксельные проекторы: как работают трехмерные дисплеи // КОД. Журнал Яндекс Практикума. <https://thecode.media/vokselynye-proektory-kak-rabotayut-tryohmernye-displei/>

4. Григорьев С. Н., Локтев М. А., Толок А. В. Построение воксельных моделей геометрических объектов // Прикладная информатика. 2013. № 4 (46). С. 50–55. EDN: SAZUMJ.
5. Желнов А. С. Применение воксельной графики для распознавания 3D объектов методами искусственного интеллекта // Моделирование нелинейных процессов и систем. Материалы седьмой международной конференции (Москва, 20–23 декабря 2023 года). М.: Янус-К, 2024. С. 79–82. EDN: LQSKVI.
6. Кудж С. А., Цветков В. Я. Моделирование динамических поверхностей // Информация и космос. 2023. № 3. С. 80–85. EDN: PZYHRJ.
7. Лебедева А. В. Реализация методов построения трехмерных моделей человеческого тела и их применение для решения задач радиационной терапии: магистерская диссертация: 016917. СПб., 2017. 41 с. <http://hdl.handle.net/11701/11667>
8. Лоторевич Е. А., Толок А. В. Воксельное графическое 3D-моделирование // Системы проектирования, технологической подготовки производства и управления этапами жизненного цикла промышленного продукта (CAD/CAM/PDM — 2013). Труды 13-й международной конференции (Москва, 15–17 октября 2013 года) / под ред. Е. И. Артамонова. М.: ООО «Аналитик», 2013. С. 205–209. EDN: TABCUR.
9. Мареев Г. О., Алайцев И. К., Ермаков И. Ю., Данилова Т. В., Мантуров А. О. Разработка базовых принципов взаимодействия с воксельной моделью в трехмерном пространстве в хирургическом симуляторе // Бюллетень медицинских интернет-конференций. 2015. Т. 5. № 12. С. 1727–1729. EDN: VKBPIT.
10. Перспективное будущее воксельной графики. <https://lizzia.go.tilda.ws/>
11. Толок А. В. Функционально-воксельный метод в компьютерном моделировании. М.: ООО Издательская фирма «Физико-математическая литература», 2016. 112 с. EDN: WIGUOJ.
12. Тулунов С. В. Применение воксельной графики в 3D моделировании // Информационные и графические технологии в профессиональной и научной деятельности. Сборник статей II Международной научно-практической конференции (Тюмень, 30 ноября 2018 года) / отв. ред. Н. И. Красовская. Тюмень: Тюменский индустриальный университет, 2018. С. 150–152. EDN: YVYNEL.
13. Цветков В. Я., Мордвинов В. А., Матчин В. Т. Технология генерации виртуальных ландшафтов с применением метода вокселизации и волюметрических данных // Образовательные ресурсы и технологии. 2023. № 1 (42). С. 91–99. EDN: XCAFSQ. DOI: 10.21777/2500-2112-2023-1-91-99.
14. Что такое воксели — и в каких программах вы можете создать воксель-арт // RENDER.RU. <https://render.ru/ru/SmirnovSchool/post/22665>
15. Шакаев В. Д., Кравец А. Г. Способы представления воксельного ландшафта при проектировании систем виртуальной реальности // Моделирование, оптимизация и информационные технологии. 2019. Т. 7. № 1 (24). С. 309–327. EDN: ZFBWPZ. DOI: 10.26102/2310-6018/2019.24.1.019.
16. Шедько И. И. Видеоигровые изобразительные стили // Художественная культура. 2021. № 4 (39). С. 382–395. EDN: EGQBOU. DOI: 10.51678/2226-0072-2021-4-382-395.
17. Atomontage. <https://www.atomontage.com/make>
18. MagicaVoxel // ephtracy. <https://ephtracy.github.io/>
19. Voxel (воксельная графика) // CG ITEMS. <https://cgitems.ru/articles/voxel-vokselnaya-grafika/>

DOI: 10.32517/2221-1993-2025-24-1-39-52

**С. Ф. Громова, Н. С. Скоробогатов***победители конкурса ИНФО-2024 в номинации**«40 лет школьной информатике: Инновации в методах и средствах обучения»,**Сургутский государственный педагогический университет, г. Сургут, Ханты-Мансийский автономный округ — Югра, Тюменская область, Россия*

«ВОКРУГ СВЕТА ЗА 34 ЧАСА». ВИРТУАЛЬНЫЙ ГЕОКЕШИНГ КАК СРЕДСТВО ФОРМИРОВАНИЯ МЕТАПРЕДМЕТНЫХ КОМПЕТЕНЦИЙ ОБУЧАЮЩИХСЯ*

Аннотация

В статье рассматриваются вопросы использования технологии виртуального геокешинга во внеурочной деятельности по информатике. Виртуальный геокешинг — это образовательная технология, представляющая собой игру, в ходе которой обучающиеся путешествуют по виртуальному миру, изучают интересные факты о городах и странах, выполняют задания прикладного характера и за правильное решение получают географические координаты новых «кладов». Необычная для педагогической практики технология виртуального геокешинга способствует активизации учебно-познавательной деятельности обучающихся, активно откликающихся на использование новых цифровых инструментов в обучении. В статье представлен авторский курс по внеурочной деятельности «Вокруг света за 34 часа», направленный на формирование метапредметных компетенций учеников IX класса. Представленные методические рекомендации, авторские подходы к организации игры, построению геолокаций, а также предложенные задания могут быть использованы учителями разных предметов при планировании и организации внеурочной деятельности.

Ключевые слова: информатика, информационные технологии, метапредметные компетенции, геокешинг, виртуальный геокешинг, внеурочная деятельность, образование.

Информация об авторах

Громова Светлана Фаутовна, канд. пед. наук, доцент, доцент кафедры высшей математики и информатики, социально-педагогический факультет, Сургутский государственный педагогический университет, г. Сургут, Ханты-Мансийский автономный округ — Югра, Тюменская область, Россия; *e-mail:* grsf@inbox.ru

Скоробогатов Никита Сергеевич, студент 1-го курса магистратуры, направление подготовки 44.04.02 «Психолого-педагогическое образование» (направленность «Цифровизация образования: проектирование, сопровождение, экспертиза»), кафедра высшей математики и информатики, социально-педагогический факультет, Сургутский государственный педагогический университет, г. Сургут, Ханты-Мансийский автономный округ — Югра, Тюменская область, Россия; *e-mail:* sns20013012@mail.ru

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_01-2025/

S. F. Gromova, N. S. Skorobogatov

Surgut State Pedagogical University, Surgut, Khanty-Mansi Autonomous Area — Yugra, Tyumen Region, Russia

"AROUND THE WORLD IN 34 HOURS". VIRTUAL GEOCACHING AS A MEANS OF FORMING STUDENTS' META-SUBJECT COMPETENCIES

Abstract

The article discusses the use of virtual geocaching technology in extracurricular activities in informatics. Virtual geocaching is an educational technology that is a game in which students travel around the virtual world, learn interesting facts about cities and countries, complete applied tasks and receive geographic coordinates of new "treasures" for the correct solution. Unusual for pedagogical practice, the virtual geocaching technology helps to activate the educational and cognitive activities of students who actively respond to the use of new digital tools in learning. The article presents the author's course on extracurricular activities "Around the world in 34 hours" aimed at developing meta-subject competencies of 9th grade students. The presented methodological recommendations, author's approaches to organizing the game, constructing geolocations, as well as the proposed tasks can be used by teachers of different subjects when planning and organizing extracurricular activities.

Keywords: informatics, information technologies, meta-subject competencies, geocaching, virtual geocaching, extracurricular activities, education.

Information about the authors

Svetlana F. Gromova, Candidate of Sciences (Education), Docent, Associate Professor at the Department of Higher Mathematics and Informatics, Faculty of Social and Pedagogical Sciences, Surgut State Pedagogical University, Surgut, Khanty-Mansi Autonomous Area — Yugra, Tyumen Region, Russia; *e-mail:* grsf@inbox.ru

Nikita S. Skorobogatov, a 1st year master's student, the Department of Higher Mathematics and Informatics, Faculty of Social and Pedagogical Sciences, Surgut State Pedagogical University, Surgut, Khanty-Mansi Autonomous Area — Yugra, Tyumen Region, Russia; *e-mail:* sns20013012@mail.ru

1. Введение. Виртуальный геокешинг и его применение в образовательном процессе

Задача современной школы заключается в формировании всесторонне развитой личности, готовой адаптироваться к быстро меняющимся условиям жизни, способной успешно решать разнообразные, в том числе нестандартные задачи. Именно поэтому в федеральных государственных образовательных стандартах нового поколения усилен акцент на развитии метапредметных компетенций школьников в соответствии с запросами современного мира.

Овладение метапредметными компетенциями создает необходимую основу для самостоятельного усвоения обучающимися новых знаний и дальнейшего их применения в различных сферах человеческой жизни. В связи с этим особую значимость приобретает деятельность педагогов, направленная на формирование метапредметных компетенций в процессе всего периода обучения. Развитие у учеников таких качеств, как умение планировать и прогнозировать собственную деятельность, самостоятельно добывать и осваивать новые знания, успешно используя их на практике, становится все более актуальным в условиях бурного развития технологий и постоянно меняющегося информационного пространства [1].

Информатика, как учебный предмет, традиционно считается метапредметной дисциплиной, обладающей высокотехнологичными средствами реализации межпредметных связей, позволяющих эффективно решать задачи из различных областей знания [3].

Использование цифровых технологий в обучении способно сделать учебную информацию более привлекательной для восприятия школьника. Современная школьная практика характеризуется интеграцией информационных и образовательных технологий [2]. Примером такой интеграции является образовательный геокешинг.

Геокешинг — это игровая технология поиска тайников, которая более двадцати лет увлекает людей по всему миру. Суть игры заключается в поиске кладов по координатам, заданным другими пользователями. Использование геокешинга в образовании позволяет сделать обучение не только познавательным, но и интересным. Реализация геокешинга весьма затратна, так как для его проведения требуются достаточно большая территория поиска, обустройство тайников, участие в игре волонтеров, обеспечение безопасности участников и т. д. Поэтому идея геокешинга, в котором в качестве тайников используются знания, а сама игра перемещена в виртуальное пространство, позволяет сделать технологию образовательного геокешинга массовой, раздвигая границы учебного кабинета, школы, города, страны и даже континента.

Применение **виртуального геокешинга**, основанного на использовании дистанционных интернет-технологий, позволяет достигать не только предметных, но, что особенно важно, и метапредметных результатов. Причем эта технология может эффективно использоваться как в урочной, так и во внеурочной деятельности [10].

Н. В. Кудимова так описывает образовательный геокешинг с позиций учителя: «Образовательный

геокешинг для преподавателей — инструмент обучения, включающий в себя критическое мышление, практическое обучение и активное участие, подразумевает “охоту” за сокровищами, имеющими образовательную ценность, на свежем воздухе по географическим координатам, что позволяет повысить мотивацию обучения, а также формировать у учащихся метапредметные компетенции» [7]. Для школьников же, отмечает автор, «образовательный геокешинг — это вариант интересной игры, во время которой ребята получают новые знания, формируются навыки работы в команде, навыки ориентирования на местности, а использование современных информационных технологий позволяет поддерживать высокий интерес к учебному процессу» [там же].

А. В. Рогозина считает, что «образовательный геокешинг — это образовательная технология, позволяющая поднять на качественно новый уровень и наполнить новым практическим содержанием организацию деятельности школьников. Данная технология позволяет сделать процесс обучения действительно актуальным, личностно-значимым, интересным и творческим» [9].

Необходимость формирования метапредметных компетенций является одновременно задачей, выходящей за пределы рамок предметной специализации, и объединяющей силой, способной вооружить современного школьника универсальными инструментами, с помощью которых можно решать любые задачи из любых областей знаний. Следовательно, каждый учебный предмет должен быть ориентирован на формирование метапредметных компетенций у обучающихся.

Среди ученых и педагогов существуют различные подходы к определению понятия «метапредметные компетенции».

Е. А. Яровая понимает метапредметные компетенции «как измеряемые способы действий, навыки и умения, субъектную позицию обучающегося, помогающие ему успешно самореализоваться в определенной области» [11].

Б. А. Крузе, Е. В. Еремеева определяют метапредметные компетенции как «комплексный ресурс личности, который обеспечивает возможность эффективной и разносторонней учебно-исследовательской деятельности, т. е. “готовность и способность” осуществлять метапредметную учебно-исследовательскую деятельность» [6].

М. М. Поташник, М. В. Левит трактуют метапредметные компетенции как «освоенные обучающимися надпредметные, межпредметные умения и универсальные действия, способность построить индивидуальную образовательную траекторию» [8].

Развитие метапредметных компетенций у обучающихся во внеурочной деятельности по информатике с применением информационных образовательных технологий создает необходимые условия для жизни и профессионального развития человека в современном цифровом обществе.

Следует отметить, что использование виртуального геокешинга в урочной деятельности связано с определенными рисками — может оказаться, что в фокусе внимания обучающихся будет прежде всего развлекательная сторона игры или ее технические аспекты (поиск координат с помощью географических онлайн-карт,

использование гаджетов и программного обеспечения), нежели учебная составляющая, да и время урока ограничено. Внеурочная же деятельность предоставляет широкие возможности для реализации технологии виртуального геокешинга, использования содержания большого спектра школьных дисциплин, применения цифровых сервисов, не задействованных в учебном процессе, апробации инновационных педагогических методов и приемов.

Целью настоящей статьи является описание технологии виртуального геокешинга как средства развития метапредметных компетенций школьников на примере курса внеурочной деятельности по информатике для учащихся IX классов.

2. Общая характеристика курса «Виртуальный геокешинг. Вокруг света за 34 часа»

Основная цель современного образования заключается в обеспечении целенаправленного процесса воспитания и обучения, результатом которого является формирование духовно-нравственной, интеллектуально развитой личности, способной решать задачи из различных сфер жизнедеятельности общества. Эффективность реализации данной цели во многом обеспечивается успехом формирования мотивации обучения. Проблема мотивации всегда актуальна для образования. Внеурочная

деятельность по информатике, основной функцией которой является достижение личностных и метапредметных результатов, позволяет успешно решать эту проблему на интересном, значимом для школьников содержании, используя инновационные технологии обучения [4].

Курс внеурочной деятельности «Виртуальный геокешинг. Вокруг света за 34 часа» **предназначен для учеников IX класса.**

В основе программы курса лежит установка на формирование у обучающихся метапредметных компетенций и их применение для решения жизненных задач.

Ссылки на паспорт программы и интерактивную игру — путешествие по странам и континентам — представлены в таблице 1.

Цифровая часть программы оформлена в едином стиле и состоит из семи геолокаций. В иконку соответствующего города встроены постоянные интерактивные элементы, что облегчает визуальное восприятие информации обучающимися (рис. 1). Все изображения взяты из открытых источников.

В процессе игры развивается **единая сюжетная линия**: ученики являются пассажирами самолета «Виртуальный геокешинг», которым управляет российский пилот гражданской авиации, Герой Российской Федерации — Дамир Касимович Юсупов, известный тем, что 15 августа 2019 года спас пассажиров, совершив успешную аварийную посадку вблизи подмосковного города Жуковский (рис. 2).

Таблица 1

Материал	Ссылка	QR-код
Паспорт программы курса внеурочной деятельности «Виртуальный геокешинг. Вокруг света за 34 часа»	https://docs.google.com/document/d/1IRH10UhoeYr6zbfWIMRmxQl0GU5Req5C/edit?usp=sharing&oui d=113107933309778407961&rtpof=true&sd=true	
Виртуальный геокешинг «Вокруг света за 34 часа»	https://view.genially.com/65511d1a96c3480011484537/interactive-content-virtualnyj-geokeshing-vokrug-sveta-za-34-chasa	

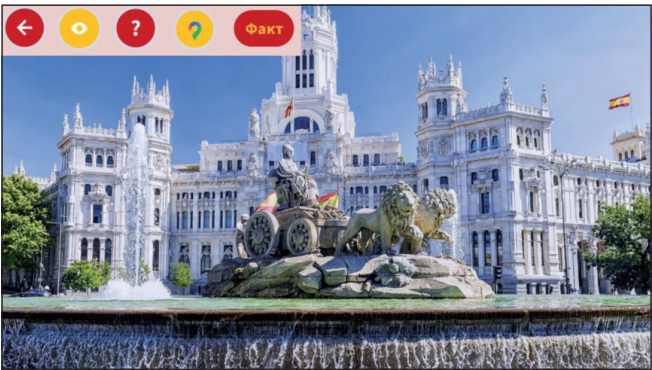


Рис. 1. Оформление цифровой части курса



Рис. 2. Дамир Юсупов

Таблица 2

Материал	Ссылка	QR-код
Демонстрационный вариант геокешинга: «Виртуальный геокешинг «От Волги до Енисея»»	https://view.genially.com/644147b28a76c600129e1af5/presentation-virtualnyj-geokeshing-ot-volgi-do-eniseya	

Каждая из семи геолокаций игры предлагает школьникам задачу по одному из учебных предметов:

- информатика;
- биология;
- физическая культура;
- история;
- литература;
- математика;
- изобразительное искусство.

И каждое из предложенных для выполнения заданий направлено на формирование метапредметных компетенций.

Игра реализована на платформе Genially, предназначенной для создания интерактивных презентаций. Школьники получают ссылку, переход по которой начинает демонстрацию интерактивных слайдов. С помощью программирования интерактивных элементов игра может развиваться по тому сценарию, который выберет сам обучающийся.

Для учителя платформа удобна возможностью доработки уже созданного продукта и редактированием игры в реальном времени. Так, в случае выявления незначительной ошибки, допущенной в процессе разработки игры, учитель может оперативно устранить недочет без изменения ссылки на образовательный ресурс [5]. Уникальностью представляемой программы внеурочной деятельности является не столько ее межпредметный и междисциплинарный характер, сколько необычный формат реализации, который является новым для образовательных организаций.

3. Содержание курса

Курс внеурочной деятельности «Виртуальный геокешинг. Вокруг света за 34 часа» состоит из трех разделов:

- 1) «Пристегните ремни, мы взлетаем!» — 3 ч;
- 2) «Правее, на солнце, вдоль рядов кукурузы» — 29 ч;
- 3) «Наш самолет готов к посадке» — 2 ч.

3.1. Раздел 1

«Пристегните ремни, мы взлетаем!» (3 ч)

Основные понятия:

- геокешинг;
- виртуальный геокешинг;
- правила игры;
- элементы игры;
- координаты;
- сервис Genially.

Методические рекомендации для учителя.

Учитель:

- знакомит школьников с технологией образовательного геокешинга;
- рассказывает об истории его возникновения;
- выделяет отличительные признаки виртуального геокешинга;
- обозначает перед школьниками цель курса, ставит его задачи.

Перед началом путешествия ученики знакомятся с правилами игры и проходят ее демонстрационный вариант, в качестве которого предлагается виртуальный геокешинг «От Волги до Енисея» (табл. 2).

В ходе представления демонстрационного варианта учитель обращает внимание учеников на основной инструмент игры — «навигатор». Для этого необходимо актуализировать знания обучающихся по теме «Поиск географических объектов по карте с помощью координат».

Каждому ученику выдаются логин и пароль от сервиса Genially, необходимые для выполнения проекта.

Рекомендации для обучающихся.

До непосредственного начала игры следует убедиться, что обучающиеся владеют такими понятиями, как «широта», «долгота», «экватор», «стороны света», «нулевой меридиан».

В разделе 1 игры школьники под руководством учителя осваивают демонстрационный вариант игры, знакомятся с ее основными элементами и правилами.

Поиск геолокаций производится с помощью навигатора, который располагается на платформе GoogleMaps (рис. 3).

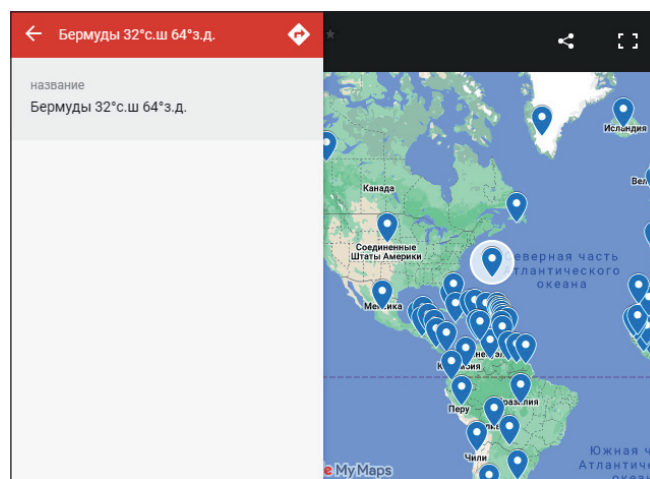


Рис. 3. Навигатор

3.2. Раздел 2

«Правее, на солнце, вдоль рядов кукурузы» (29 ч)

3.2.1. Введение в игру (1 ч)

Основные понятия:

- виртуальный геокешинг;
- игра;
- групповая работа.

Методические рекомендации для учителя.

Компьютерная игра «Виртуальный геокешинг “Вокруг света за 34 часа”» может быть реализована в двух форматах. Первый — индивидуальный. Обучающийся может самостоятельно проходить игру в комфортном для себя режиме. Такой формат будет способствовать процессу формирования метапредметных компетенций по мере продвижения от задания к заданию. Но мы рекомендуем более интересный вариант — групповой. Для этого в самом начале игры следует разделить класс на группы.

Скриншот титульной страницы виртуального геокешинга представлен на рисунке 4.



Рис. 4. Титульная страница игры

В разделе 2 курса обучающиеся выполняют путешествие по семи геолокациям. На каждую геолокацию отводится по четыре часа. Геолокация представляет собой определенную страну мира, в которой путешественники знакомятся с тремя городами.

Посещение каждого из городов направлено на решение одной из следующих задач:

- 1) решение заданий на функциональную грамотность;
- 2) выполнение творческого задания по учебной теме;
- 3) реализация проектного задания.

В разделе обучающиеся самостоятельно выполняют все практические задания и работают над собственным проектом — созданием виртуального города на платформе Genially.

Рекомендации для обучающихся.

На протяжении всего курса школьники выполняют творческое задание — создают собственный виртуальный геокешинг, описывающий достопримечательности одного из городов ХМАО — Югры.

В каждой геолокации игроки должны посетить три города, в которых будут выполнять различные предметные задания, разрабатывать свои творческие упражнения, проходить обучение по работе с платформой Genially

для создания собственного виртуального города. Также они получают возможность познакомиться с достопримечательностями городов, особенностями культуры и образа жизни жителей страны, в которую попали в ходе игры.

Посещение городов осуществляется нажатием кнопки «Аэропорт» (рис. 5).



Рис. 5. Аэропорт

Город 1. В нем обучающиеся выполняют задания, предложенные учителем. Верное выполнение заданий является условием получения координат следующего города.

Город 2. В этом городе необходимо разработать собственное упражнение в соответствии с техническим заданием для того, чтобы встроить его в свой виртуальный город.

Город 3. В ходе его посещения ученики проходят пошаговое обучение по работе с платформой Genially, необходимое для разработки индивидуального проекта.

Во время групповой работы обучающимся следует придерживаться следующих правил:

- 1) Внимательно ознакомиться с материалами заданий.
- 2) Организовать обсуждение для выработки алгоритма решения задания.
- 3) Распределить выполнение задания среди участников группы.
- 4) Обсудить результаты индивидуальной работы в группе для выработки коллективного решения.
- 5) Подвести итоги выполнения группового задания.
- 6) Оформить результаты работы в соответствии с требованиями.

3.2.2. Геолокация 1 (4 ч)

Основные понятия:

- обработка текстовой информации;
- шифрование информации;
- шифр Виженера;
- шифр координат;
- практическая работа;
- групповое взаимодействие;
- текстовый процессор (Microsoft Word).

Раздел открывает первое путешествие, в котором обучающимся предстоит выполнить задания по информатике, используя текстовый процессор (например, Microsoft Word). В практическом задании необходимо зашифровать и расшифровать информацию с помощью шифра Виженера и предложенного «шифра координат».

Таблица 3

Материал	Ссылка	QR-код
Геолокация 1	https://view.genially.com/655b9b2c25f9300010aed276/interactive-content-ssha	
Практическая работа 1.1	https://docs.google.com/document/d/1gLGnvmfnvSpto10UTEsi5jjQRfTrIew87LqnaQe3Wo/edit?usp=drive_link	
Практическая работа 1.2	https://docs.google.com/document/d/1qZsu6c457ePDT5y-zzNdL2Awc_mAmdgx7Ln6qHtJOU0/edit?usp=sharing	



Рис. 6. Титульная страница геолокации 1

Каждую практическую работу предваряет теоретический материал, позволяющий школьникам успешно справиться с заданиями.

В ходе выполнения практических работ учитель осуществляет мониторинг за деятельностью обучающихся, консультирует, оказывает помощь.

Результатом прохождения первого раздела является разработанная обучающимися титульная страница собственного виртуального города. Ссылки на практические работы раздела представлены в таблице 3*.

С предметными и метапредметными задачами раздела можно ознакомиться, нажав интерактивную кнопку на титульной странице геолокации 1 (рис. 6).

В Приложении представлен пример практической работы.

* Также тексты всех практических работ и материалы к ним можно скачать в дополнительных материалах к данной статье на сайте ИНФО: http://infojournal.ru/journals/school/school_01-2025/

3.2.3. Геолокация 2 (4 ч)

Основные понятия:

- электронная таблица;
- относительная ссылка;
- абсолютная ссылка;
- индекс массы тела;
- здоровый образ жизни;
- рацион питания.

При прохождении данного раздела (рис. 7) обучающиеся выполняют задания по биологии, используя возможности табличного процессора (например, MS Excel).

Перед выполнением практической работы учитель может актуализировать знания школьников по теме «Нутриенты» и продемонстрировать возможности табличного процессора MS Excel.

Ссылки на практические работы раздела представлены в таблице 4.



Рис. 7. Титульная страница геолокации 2

Таблица 4

Материал	Ссылка	QR-код
Геолокация 2	https://view.genially.com/656e8c2ed8531700131d733b/interactive-content-yaponiya	
Материалы для практических работ 2.1 и 2.2	https://docs.google.com/spreadsheets/d/10No12qJi3fi4RS0eQ1a0fTIgJU0VQhcm/edit?usp=drive_link&ouid=113107933309778407961&rtpof=true&sd=true	
Практическая работа 2.1	https://docs.google.com/document/d/1DoTCN7mGnvfd0xhqZVHnUGFj-J7U2iruwDxccJbp6qk/edit?usp=sharing	
Практическая работа 2.2	https://docs.google.com/document/d/1ysMxP7eoKLV3b3xl3x0FI2K1LOJ4bmtQm7ZPytO6tW8/edit?usp=sharing	

3.2.4. Геолокация 3 (4 ч)

Основные понятия:

- база данных;
- СУБД;
- объекты базы данных;
- таблицы;
- запросы;
- формы;
- отчеты;

- ключ;
- перекрестный запрос.

В данном разделе (рис. 8) учитель знакомит обучающихся с основными понятиями баз данных, демонстрирует особенности СУБД MS Access, контролирует правильность выполнения заданий, консультирует участников групп.

Ссылки на практические работы раздела представлены в таблице 5.



Рис. 8. Титульная страница геолокации 3

Таблица 5

Материал	Ссылка	QR-код
Геолокация 3	https://view.genially.com/655b8e559db8e20011362c51/interactive-content-braziliya	
Материалы для практических работ 3.1 и 3.2	https://drive.google.com/file/d/1jTcg3kGiVS_pw9Amx7TkM7gOUuXwwMTv/view?usp=drive_link	
Практическая работа 3.1	https://docs.google.com/document/d/19YF-u4EbE1CpyHIAj4cYRS_pR6Q9pOp_5DIscuEw6hs/edit?usp=sharing	
Практическая работа 3.2	https://docs.google.com/document/d/1noLVDfDE7-VGopebtbntcPUQ2hb4zFkkKS_3x0GUkl8/edit?usp=drive_link	

3.2.5. Геолокация 4 (4 ч)

Основные понятия:

- история информатики;
- ученые;
- лента времени;
- TimeGraphics;
- Genially;
- нейросеть.

В данном разделе (рис. 9) учитель демонстрирует школьникам возможности сервиса TimeGraphics для

создания ленты времени по истории информатики, а потом и по истории своих виртуальных городов.

В одном из заданий практической работы предлагается сгенерировать изображения с помощью нейросети. Учитель знакомит обучающихся с правилами безопасности при работе с искусственным интеллектом, а также демонстрирует возможности бесплатных сервисов. Рекомендуются использовать чат-бот Kandinsky в Telegram. Особенность сервиса заключается в том, что можно описать изображение, задать стилистику и уточнить детали картинки, при



Рис. 9. Титульная страница геолокации 4

Таблица 6

Материал	Ссылка	QR-код
Геолокация 4	https://view.genially.com/655b9631e657560011083bb7/interactive-content-egipet	
Практическая работа 4.1	https://docs.google.com/document/d/1OoGc2AJSdT_ZIHjlj_iZ5u-R7H4B90bge9KPZ_r_Oik/edit?usp=sharing	
Практическая работа 4.2	https://docs.google.com/document/d/1K9ZVgoVr1gMnVbFRFCd5xCZ781amGH6uv8Cu-ibcxvg/edit?usp=sharing	

этом сгенерированная картинка будет выглядеть как реальная фотография.

Ссылки на практические работы раздела представлены в таблице 6.

3.2.6. Геолокация 5 (4 ч)

Основные понятия:

- дополненная реальность;
- видеоролик;
- проект;
- AR.Studio.

В этом разделе (рис. 10) учитель знакомит обучающихся с сервисом AR.Studio, позволяющим создавать оживающие фотографии, тесты с дополненной реальностью и многое другое. В практической работе 5.1 школьникам предлагается учебное видео по работе с сервисом.

После прохождения обучения школьники выполняют творческое задание — создают видеоролик.

Ссылки на практические работы раздела представлены в таблице 7.

3.2.7. Геолокация 6 (4 ч)

Основные понятия:

- площадь фигур;
- треугольники;
- четырехугольники;
- биссектриса;
- медиана;
- высота;
- площадь;
- проект;
- сервис GeoGebra.

В данном разделе (рис. 11) учитель знакомит обучающихся с сервисом GeoGebra. В практических работах 6.1 и 6.2 в помощь учителю предлагается ряд готовых заданий и электронное методическое пособие. Необходи-



Рис. 10. Титульная страница геолокации 5

Таблица 7

73,8 мм	Ссылка	QR-код
Геолокация 5	https://view.genially.com/655b9eb2d95886001297246f/interactive-content-ispaniya	
Практическая работа 5.1	https://docs.google.com/document/d/1SNhbLvwcOoHlyyTI-vb1saRCQBD6GMfo0q7wysTVAjK/edit?usp=sharing	
Практическая работа 5.2	https://docs.google.com/document/d/1ynQekyhLSg-B8EeW-M8W22HFOvjR3e9f3nxjqe3mKaA/edit?usp=sharing	



Рис. 11. Титульная страница геолокации 6

Таблица 8

Материал	Ссылка	QR-код
Геолокация 6	https://view.genially.com/655ba2a2e65756001110823a/interactive-content-velikobritaniya	
Материалы для практической работы 6.1	https://drive.google.com/drive/folders/1Z16497F7ZR1vBIMwzgN6_a46u3qh8C8E?usp=sharing	
Практическая работа 6.1	https://docs.google.com/document/d/1LN811kf1itZZT4gbNDgvpdzligwv1fqdsJldpv-1XA0/edit?usp=sharing	
Практическая работа 6.2	https://docs.google.com/document/d/16xEGC16EmIEBXbVAyC8KoEE38dsq-qM5KsEKqBN_bD0/edit?usp=sharing	

димо продемонстрировать обучающимся возможности приложения GeoGebra для решения математических задач, например, для построения геометрических объектов, вычисления площадей фигур, нахождения величин углов.

Ссылки на практические работы раздела представлены в таблице 8.

3.2.8. Геолокация 7 (4 ч)

Основные понятия:

- анимация;

- мастер презентаций Microsoft PowerPoint;
- проект.

В данном разделе (рис. 12) учитель демонстрирует ученикам возможности приложения презентационной графики Microsoft PowerPoint.

В практической работе 7 обучающиеся должны создать архитектурный объект города с помощью геометрических фигур.

Ссылки на практические работы раздела представлены в таблице 9.

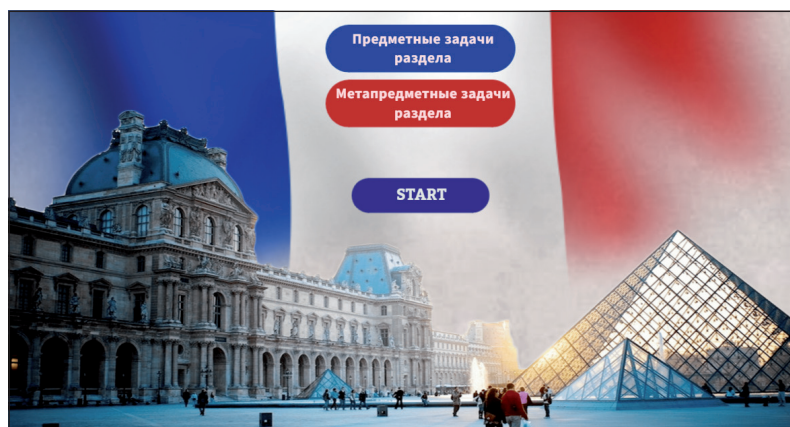


Рис. 12. Титульная страница геолокации 7

Таблица 9

Материал	Ссылка	QR-код
Геолокация 7	https://view.genially.com/655b9c642572d0001125f07c/interactive-content-franciya	
Практическая работа 7.1	https://docs.google.com/document/d/1do8yL2t5pIPtOFVzn1QvoftuVxyLZ6V1VYiUyGlsrEo/edit?usp=sharing	
Материалы для практической работы 7.2	https://docs.google.com/presentation/d/1aeGxCHzR5AsEngfadH9uF7z2QOidURKu/edit?usp=sharing&ouid=113107933309778407961&rtpof=true&sd=true	
Практическая работа 7.2	https://docs.google.com/document/d/1MRGNv02M2W1JEDjOSWf_-t9qcomGSgVXOxd9db2j9mY/edit?usp=sharing	

3.3. Раздел 3

«Наш самолет готов к посадке» (2 ч)

Основные понятия:

- проект;
- подготовка к защите проекта;
- виртуальный город.

Это итоговый раздел курса. Обучающиеся:

- завершают оформление своего виртуального города;
- проверяют работу интерактивных элементов;
- соотносят результат с требованиями технического задания;

- проводят внутреннюю экспертизу проекта.

Перед защитой проектов все группы объединяют созданные виртуальные города в единую интерактивную карту региона, в которой отражены достопримечательности городов, их история, объекты культурного наследия, факты о флоре и фауне и т. д.

В качестве методической поддержки курса предлагается готовый шаблон виртуальной карты на платформе Genially, который поможет обучающимся объединить все разработанные проекты в единую интерактивную игру (рис. 13).

Ссылки на пример итогового проекта представлены в таблице 10.



Рис. 13. Шаблон карты региона

Таблица 10

Материал	Ссылка	QR-код
Пример проекта (карта ХМАО-Югры)	https://view.genially.com/65b79934a6997400149a4f16/interactive-content-virtualnyj-geokeshing-hmao-yugra	

В процессе защиты проекта обучающиеся должны продемонстрировать результаты творческих работ, выполненных по мере прохождения курса, и представить интерактивный плакат, объединивший виртуальные города в единое целое.

4. Заключение

Курс «Виртуальный геокешинг. Вокруг света за 34 часа» стал логическим завершением серии игр для школьников в стиле виртуального геокешинга, которые по достоинству оценили ученики и учителя школ Сургута и Сургутского района. Виртуальное путешествие по странам и континентам в течение 34 учебных занятий было успешно апробировано в естественно-научном лицее г. Сургута, а сама технология виртуального геокешинга вызвала живой интерес у школьников.

Представленные материалы могут быть использованы как полноценный курс внеурочной деятельности по информатике, а также взяты за основу проектирования учебных занятий по различным предметам, внеклассной и внеурочной деятельности.

Список источников

1. *Бороненко Т. А., Федотова В. С.* Школьный курс информатики в эпоху цифровых трансформаций: приоритетное направление — развитие цифровой грамотности // Информатика в школе. 2021. № 4 (167). С. 3–15. EDN: UTBNAL. DOI: 10.32517/2221-1993-2021-20-4-3-15.
2. *Босова Л. Л.* О целях современного образования школьников в области информатики и информационных технологий // Известия Кыргызской академии образования. 2020. № 2. С. 214–219. EDN: BUVVMB.
3. *Босова Л. Л.* Цифровые навыки современного школьника и возможности их формирования в школьном курсе информатики // Информатика в школе. 2020. № 7 (160). С. 5–9. EDN: BPEULP. DOI: 10.32517/2221-1993-2020-19-7-5-9.
4. *Босова Л. Л., Босова А. Ю.* Информатика. 7–9 классы: методическое пособие. 3-е изд., перераб. М.: БИНОМ. Лаборатория знаний, 2019. 512 с. EDN: CIRXGN.
5. *Громова С. Ф.* Виртуальный геокешинг «Достижения Югры» как средство краеведческого воспитания // Актуальные вопросы математического образования: состояние, проблемы и перспективы развития. Сборник статей по материалам VIII Всероссийской научно-практической конференции с международным участием (Сургут, 10–12 апреля 2024 года). Сургут: Сургут. гос. пед. ун-т, 2024. С. 84–89. EDN: EBSVUF.
6. *Крузе Б. А., Еремеева Е. В.* Определение понятия метапредметных компетенций младшего школьника // Современные проблемы науки и образования. 2013. № 6. С. 271. EDN: RVCRTN.
7. *Кудимова Н. В.* Образовательный геокешинг в загородном лагере // Народное образование. 2013. № 3 (1426). С. 126–129. EDN: QAFMAL.

8. *Поташник М. М., Левит М. В.* Как помочь учителю в освоении ФГОС: методическое пособие. М.: Педагогическое общество России, 2014. 320 с.

9. *Рогозина А. В.* Геокешинг как образовательная технология формирования универсальных учебных действий // Инновационные процессы и технологии в современном мире. 2016. № 1 (4). С. 57–59. EDN: XUZTED.

10. *Шабанова М. В., Патронова Н. Н., Безумова О. Л.* «Геометрический Geocaching»: опыт сетевого наставничества // CONTINUUM. Математика. Информатика. Образование. 2022. № 3 (27). С. 82–92. EDN: KPCGXW. DOI: 10.24888/2500-1957-2022-3-82-92.

11. *Яровая Е. А.* О метапредметных компетенциях и их видах // Научные труды SWorld. 2015. Т. 6. № 3 (40). С. 66–71. EDN: UZJTID.

Приложение

ПРАКТИЧЕСКАЯ РАБОТА 1.2 Шифр виртуального геокешинга

Тема: Шифрование информации.

Форма проведения: практическое занятие.

Для того чтобы зашифровать слово, необходимо:

1. Составить таблицу:

1-я строка — исходное слово шифра (МОСКВА);

2-я строка — каждой букве исходного слова ставится в соответствие любая страна, название которой начинается на эту букву (М — Монако, О — Оман и т. д.).

Список всех стран можно посмотреть, перейдя по ссылке:

<https://ru.freeflagicons.com/country>

3-4-я строки — каждому символу исходного слова соответствуют пять позиций (цифр): первые две соответствуют широте, остальные три — долготе (С — 41° с. ш. 100° з. д.).

Обратите внимание! Если долгота страны — двузначное число, то обязательно поставьте перед ним ноль.

Всегда должно быть:

- широта — два символа;
- долгота — три символа.

Например: Эфиопия имеет координаты: 7° с. ш. 39° в. д. Следовательно, чтобы зашифровать букву Э, нужно записать: 07039.

2. С помощью навигатора выбрать любую точку, принадлежащую данной стране:

https://www.google.com/maps/d/edit?mid=1smc-VB16XxSF5f2_ZAp3SLm9MqpHjIM&usp=sharing

Обратите внимание! Для того чтобы записать координаты *северной широты и восточной долготы*, необходимо использовать *положительные числа*. Для записи *южной широты и западной долготы* необходимо использовать *отрицательные числа*.

Например: США имеет координаты: 41° с. ш. 100° з. д. Следовательно, чтобы зашифровать букву С, нужно записать: 41–100.

Обратите внимание! Буквы Ж, Щ, Ъ, Ы, Ь шифруются как пять цифр 9: 99999.

Пример 1.

Зашифруем название города МОСКВА:

1	М		О		С		К		В		А	
2	Монако		Оман		США		Камбоджа		Вьетнам		Австралия	
3	Шир.	Долг.	Шир.	Долг.	Шир.	Долг.	Шир.	Долг.	Шир.	Долг.	Шир.	Долг.
4	43	007	19	056	41	–100	12	105	13	108	–24	127

В результате получаем следующий шифр (числовой ряд):

430071905641–1001210513108–24127

Для того чтобы расшифровать слово, необходимо:

(рассмотрим на конкретном примере, когда задан числовой ряд:

430071905641–1001210513108–24127)

1. Составить таблицу:

1-я строка — разбейте числовой ряд по пять цифр: 43007 19056 41–100 12105 13108 –24127

2-я строка — разбейте каждую «пятерку» на две и три цифры:

43007 = 43 007, 41–100 = 41 –100 и т. д.

3-я строка — запишите широты и долготы:

43 — с. ш., 007 — в. д. и т. д.

Отрицательная широта соответствует *южной широте* (–24 = 24° ю. ш.).

Отрицательная долгота соответствует *западной долготы* (–100 = 100° з. д.).

Обратите внимание! Первые два числовых значения — широта, остальные *три* — долгота.

4-я строка — с помощью навигатора найдите и запишите страну, зная ее координаты:

https://www.google.com/maps/d/edit?mid=1smc-VB16XxSF5f2_ZAp3SLm9MqpHJiM&usp=sharing

5-я строка — запишите первую букву каждой страны.

Пример 2.

Код: 430071905641–1001210513108–24127.

1	43007		19056		41–100		12105		13108		–24127	
2	43	007	19	056	41	100	12	105	13	108	24	127
3	с.ш.	в.д.	с.ш.	в.д.	с.ш.	з.д.	с.ш.	в.д.	с.ш.	в.д.	ю.ш.	в.д.
4	Монако		Оман		США		Камбоджа		Вьетнам		Австралия	
5	М		О		С		К		В		А	

Задание 1.

В таблице представлены коды, которые вам необходимо расшифровать в соответствии с алгоритмом, представленном выше. Номер варианта совпадает с номером вашей группы.

В результате выполнения задания 1 каждая группа получит название города, который станет объектом создания вашего личного виртуального геокешинга. Успехов!

Вариант 1	01103–33–0556109705–0580203206001
Вариант 2	12–08418–07715–09018–063–2201699999
Вариант 3	45015–3701817008–0703599999
Вариант 4	0603014–08619056–2105513–08821–078
Вариант 5	0203261097–2412715046
Вариант 6	22–102240275001039–0031905617008
Вариант 7	–2902812–0695100516–0612402718–0661806304–056
Вариант 8	22–0781905650010–37018–290289999914–061
Вариант 9	06–00918–07712–08410–06119056–21055
Вариант 10	–10–0751905659–12126003–25–06939–003

Задание 2.

Придумайте и опишите свой шифр. Данный шифр будет являться паролем для въезда в ваш будущий виртуальный город.

Ответы к практической работе 1.2

Вариант 1	01103–33–0556109705–0580203206001	Сургут
Вариант 2	12–08418–07715–09018–063–2201699999	Нягань
Вариант 3	45015–3701817008–0703599999	Ханты
Вариант 4	0603014–08619056–2105513–08821–078	Югорск
Вариант 5	0203261097–2412715046	Урай
Вариант 6	22–102240275001039–0031905617008	Мегион
Вариант 7	–2902812–0695100516–0612402718–0661806304–056	Лангепас
Вариант 8	22–0781905650010–37018–290289999914–061	Когалым
Вариант 9	06–00918–07712–08410–06119056–21055	Лянтор
Вариант 10	–10–0751905659–12126003–25–06939–003	Покачи



Е. А. Еремин

Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия

МНОГОСЛОЙНАЯ МОДЕЛЬ КОМПЬЮТЕРА КАК ОРИЕНТИР ПРИ ОТБОРЕ МАТЕРИАЛА ДЛЯ КУРСА ИНФОРМАТИКИ

Аннотация

Многослойная модель компьютера — это представление его в виде последовательности слоев (уровней): от физического кодирования значений 0 и 1 до прикладного программного обеспечения. Для формирования целостной картины знаний каждый слой имеет определенное значение. В статье традиционно выделяемые слои модели сопоставляются с содержанием сложившегося в школе курса информатики. Показано, что в программе углубленного курса X—XI классов можно найти материал, в той или иной степени связанный с каждым из слоев. Тем не менее в свете возникшей острой необходимости в вычислительной технике отечественного производства и ее элементной базе, удельный вес изучения некоторых слоев следует увеличить. На основе проведенного анализа федеральных рабочих программ школьного курса информатики приводятся конкретные рекомендации по возможному совершенствованию содержания предмета. Статья будет полезна учителям информатики, а также школьникам, интересующимся компьютерами и желающим связать с ними будущую специальность.

Ключевые слова: слой, уровень, многослойная модель, компьютер, школьный курс, информатика, целостность, систематичность.

Информация об авторе

Еремин Евгений Александрович, канд. физ.-мат. наук, доцент, доцент кафедры прикладной информатики, информационных систем и технологий, факультет информатики и экономики, Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия; *e-mail*: evg_eremin@mail.ru

Достаточно однажды увидеть всю картину целиком, чтобы все стало на свои места, подобно тому как с последним десятком кусочков складывается большая мозаика-головоломка.

Стивен Кинг

1. Введение

Компьютер — одно из фундаментальных понятий курса информатики. О том, что при его изучении удобно пользоваться представлением в виде последовательно до-

бавляемых слоев, известно давно. На наш взгляд, наиболее удачно для знакомства с указанной концепцией подходят классический учебник Э. С. Таненбаума «Архитектура компьютера» [15], а также недавно переизданная объемная книга С. Л. Харриса, Д. Харриса «Цифровая схемотехника и архитектура компьютера: RISC-V» [24] для подготовки разработчиков современной вычислительной техники.

В данной статье многослойная модель компьютера сопоставляется с содержанием школьного курса ин-

Е. А. Eremin

Perm State Humanitarian Pedagogical University, Perm, Russia

MULTILAYER MODEL OF COMPUTER AS A REFERENCE POINT DURING SELECTING MATERIAL FOR THE INFORMATICS COURSE

Abstract

The multilayer model of computer is its representation as a sequence of layers (levels): from physical encoding of the values 0 and 1 to application software. Each layer has a specific meaning for forming a complete picture of knowledge. In the article the traditionally distinguished layers of the model are compared with the content of the established at the school informatics course. It is shown that in the program of advanced course for grades 10–11 the material that is to one degree or another related to each of the layers can be found. However, in light of the urgent need for domestically produced computing equipment and its element base, the proportion of studying some layers should be increased. Based on the analysis of the federal work programs for the school informatics course, specific recommendations are provided on possible improvement of the discipline content. The article will be useful for teachers, as well as for pupils interested in computers and wishing to connect with them future specialty.

Keywords: layer, level, multilayer model, computer, school course, informatics, integrality, systematicity.

Information about the author

Evgeniy A. Eremin, Candidate of Sciences (Physics and Mathematics), Docent, Associate Professor at the Department of Applied Informatics, Information Systems and Technologies, Faculty of Informatics and Economics, Perm State Humanitarian Pedagogical University, Perm, Russia; *e-mail*: evg_eremin@mail.ru

форматики. Показано, что традиционно выделяемые в модели слои в той или иной степени в сложившемся курсе присутствуют, хотя внимание к некоторым из них недостаточно (аргументация приводится в завершающей части после описания всех слоев).

Опираясь на свой продолжительный педагогический опыт, мы считаем, что значение основополагающих моделей, которые способны придать целостность содержанию учебного курса, недооценивается. В качестве аргументации сошлемся на глубокий анализ в публикации М. Е. Бершадского [2], где обсуждается роль понимания в усвоении материала и механизм формирования понимания. В частности, там показывается, что «ученик понимает новый материал, если в его сознании известные и новые предметы мышления могут быть соединены с помощью известных ученику видов связей» [там же]. Ситуация дополнительно осложняется тем, что, как свидетельствуют эксперименты, описанные в статье [30], интуитивно складывающаяся в голове студента структура знаний подчас носит весьма причудливый и далекий от систематичности характер. Вместо того чтобы взять за основу формирующейся картины наиболее общие понятия, студенты неосознанно берут наиболее важные с их собственной точки зрения; в роли таких «опорных» пунктов часто оказываются, в частности, яркие (но второстепенные) примеры. Авторы работы [30] назвали этот эффект «*paths of lower-cost*», что по смыслу лучше всего соответствует русской фразе «по линии наименьшего сопротивления». Из сказанного становится ясно, что наличие в голове ученика некоторой упорядоченной модели, на которую будет «ложиться» новый материал, существенно повышает систематизацию, а значит, и качество знаний.

Тем, кто получил образование в СССР, здесь наверняка вспомнятся известные слова К. Маркса о преимуществе самого плохого архитектора над наилучшей пчелой: человек «с самого начала отличается тем, что, прежде чем строить ячейку из воска, он уже построил ее в своей голове» [9]. В цитируемом высказывании подчеркивается значение внутренней модели внешнего мира как основы любой человеческой деятельности. Эта старая цитата сейчас неожиданно начинает играть новыми красками: как говорилось на пленарном заседании Всероссийского съезда учителей информатики, наличие собственной внутренней модели мира является главным отличием человеческого разума от современного искусственного интеллекта.

Таким образом, данная статья именно про *значение мировоззренческих основ* при планировании школьного курса информатики в противовес имеющемуся «кое-где у нас порой» подходу, когда изучение сосредоточено на *наборе сведений*, необходимых для решения двух-трех десятков типов контрольных задач, разработанных в целях итоговой оценки знаний по предмету.

Знакомство с описанной моделью будет полезно, в первую очередь, учителям информатики: целостное представление о предмете всегда помогает в преподавании. Кроме того, благодаря тщательному отбору материала и доступному его изложению, статья также может быть рекомендована и школьникам, которые интересуются компьютерами, и особенно тем, кто собирается связать с ними свою будущую профессию.

Замечание о терминологии.

В информатике часто используются два очень близких по значению термина — *уровень* (англ. level, layer) и *слой* (англ. layer, stratum). Если заглянуть в толковые словари русского языка, то для обсуждаемой сейчас темы лучше всего подойдут следующие формулировки:

Уровень — это подразделение чего-нибудь целого, получаемое при его расчленении, а слой — это пласт, который лежит между другими пластами и имеет более или менее четкие границы с ними; то, что образовалось поверх другого, что, возникнув позже, покрывает собой другое.

Видно, что смысл довольно близок. Но все же, по нашему мнению, термин «уровень» немного больше ассоциируется с положением в некоторой разветвленной иерархии объектов, а «слой» — с упорядоченной последовательностью частей определенного назначения и устройства. Именно поэтому модель в названии статьи названа *многослойной*, а не *многоуровневой*, как тоже вполне можно было бы написать.

Данная статья во многом опирается на учебник Э. С. Таненбаума [15]. В его оригинальном английском тексте в предисловии прямо написано: «We may view a computer <...> as a series of *layers* or *levels*, one on top of another», т. е. мы можем рассматривать компьютер как последовательность *слоев* или *уровней*, расположенных один над другим. Так что, следуя классике, оба обсуждаемых термина можно смело использовать как синонимы.

2. Особенности многослойных моделей

Многослойная модель полезна, когда мы имеем дело со сложным объектом. Разобраться во всех его деталях затруднительно (да и нужно ли?), поэтому важно выделить среди них наиболее значимые. При таком подходе, как правило, возникают группы понятий и принципов, которые относятся к некоторой общей области, — это и есть слои или уровни. Каждый слой включает в себя определенные базовые объекты, их функции и закономерности поведения. Каждый новый слой надстраивается над предыдущим и опирается на него. Строго говоря, для понимания работы более высокого слоя не обязательно знать все особенности нижележащего, но для формирования адекватной картины крайне полезно иметь представление о наиболее важных из них. Как писал М. Бен-Ари, «в каждом отдельном курсе вы будете учить определенному уровню абстракции; вы должны представить в явном виде жизнеспособную модель на один уровень ниже, чем тот, который вы учите» [27].

Если же требуется не просто освоить некоторый набор практических навыков, а сформировать общее систематическое представление об объекте, тогда понадобятся (хотя бы самые минимальные) сведения обо всех его уровнях. Здесь мы касаемся предмета постоянных дискуссий о том, насколько нужно понимать работу компьютера, чтобы им грамотно пользоваться. Сошлемся на мнение авторитетного специалиста, британского ученого Б. дю Бюлей: «Даже если не делается попыток представить картину того, что происходит “внутри”, ученики сформируют свою собственную. Без помощи [*учителя*] их картина будет довольно бедной, базирующейся на совпадениях и, может быть, недостаточной, чтобы объ-

яснить большую часть наблюдаемого поведения. Пока все идет хорошо, это, возможно, не имеет значения, но понимание ошибок часто требует знаний вида «машина пыталась сделать это, ... но наткнулась на то-то ... и поэтому не знает, как продолжить»» [29].

При переходе от одного слоя к другому активно используется **принцип абстракции**, когда, не принимая во внимание второстепенные детали, удается свести сложный предмет к более простому для понимания. Как и у любого полезного средства, кроме положительного эффекта часто возникают и побочные. В статье [26] принцип упрощения путем создания многочисленных «защитных слоев» несколько иронично назван «капустным»: «Один из эффектов “кочана” заключается в снижении квалификационных требований к специалистам массовых профессий, причем не только к тем, кто занимается эксплуатацией информационных систем, но даже и к тем, кто ведет массовые направления разработок. Логическим результатом прогресса в таком нетребовательном к уровню образования направлении становится преимущественное положение “монтерских” знаний в сравнении со знаниями академическими. <...> Надо сказать, что ИТ не одиноки в этом отношении, <...> например, нечто подобное можно обнаружить, если сравнить электротехнику с соответствующими разделами физики».

Посмотрим на обсуждаемую тему под другим углом. Добавляя уровни абстракции, мы упрощаем картину и облегчаем ее понимание. С одной стороны, это, бесспорно, хорошо. С другой стороны, при этом мы перестаем видеть и теряем возможность использовать многие полезные свойства нижних слоев. Чтобы нагляднее это себе представить, вспомним языки программирования высокого уровня. Да, их гораздо легче освоить, чем встроенный язык процессора, на высоком уровне легче и быстрее пишется программа. Но зато при этом из-за появления дополнительных слоев падает быстродействие (как тут не вспомнить «медленный» Python), а из-за «удаления» от аппаратной части компьютера мы теряем возможность ею эффективно управлять. Так что какие-то знания о нижних слоях все же могут приносить пользу.

3. Структура многослойной модели компьютера

Перейдем теперь непосредственно к рассмотрению многослойной модели компьютера. Внимательно проанализировав уровни, предложенные в упомянутых выше

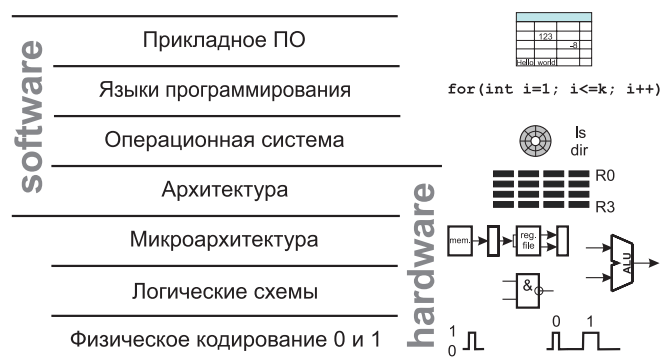


Рис. 1. Многослойная модель компьютера

учебниках [15] и [24], применительно к школьному курсу информатики, мы предлагаем сформировать модель из следующих слоев (рис. 1).

Далее каждый слой модели будет рассмотрен подробнее.

Заметим, что аналогичная многоуровневая модель от физической передачи сигналов до взаимодействия прикладных программ существует и для компьютерных сетей; она называется OSI — Open Systems Interconnection (модель взаимодействия открытых систем) [10].

3.1. Физическое кодирование 0 и 1

Физическое кодирование — это методы преобразования нулей и единиц в физические сигналы (сейчас чаще всего электрические, но можно брать за основу и более современные оптические, а также устаревшие механические или гидравлические методы двоичного представления). Проще говоря, это способ, с помощью которого хранятся или передаются (что, как мы скоро увидим, не совсем одно и то же) нулевое и единичное значения каждого из битов. Очевидно, что данный слой устройства компьютера с мировоззренческой точки зрения необычайно важен, хотя непосредственно на практическую работу пользователя на компьютере почти не влияет.

Самый простой и самый распространенный способ физического кодирования состоит в том, что каждому из двух логических значений (0 или 1) ставится в соответствие определенный уровень напряжения. Важно подчеркнуть, что ради повышения надежности распознавания всегда допустимо некоторое отклонение от стандартного значения (рис. 2).

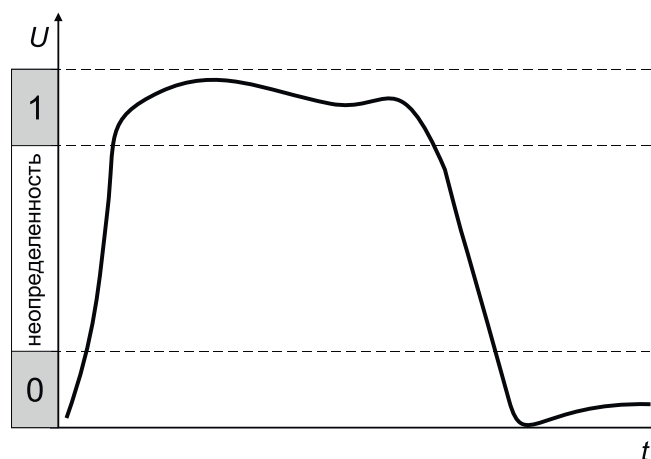


Рис. 2. Соответствие уровней напряжения с логическими 0 и 1

На рисунке 2 сигнал некоторое время попадает в область неопределенности (между 0 и 1). Это так называемый «переходный процесс», влияние которого легко исключается путем синхронизации по времени: считывание уровня включается с некоторой задержкой и выключается через достаточный для распознавания промежуток времени. В результате в момент переключения распознавание просто-напросто «не работает». Практическая реализация описанного способа не вызывает затруднений у инженеров.

Интересно отметить, что величина напряжения (в том числе даже знак!) часто зависит от выбранных ло-

гических микросхем [24]. Для нашего разговора о слоях это не совсем удобно, поскольку логические схемы на данном уровне пока еще не рассматривались.

Обсуждение вопросов физического кодирования также невозможно без разговора об аналоговых и цифровых сигналах. По сути, рассмотренный нами рисунок 2 показывает, как работает распознавание дискретных значений 0 и 1 в «непрерывном», т. е. аналоговом, диапазоне значений напряжения.

А теперь рассмотрим физическое кодирование при обмене данными между компьютерами. Тут появляется дополнительная проблема: если внутри одного компьютера все процессы строго синхронизированы (за этим следит устройство управления), то при передаче сигнала между разными компьютерами все сложнее. Представим себе, что 0 и 1 по-прежнему кодируются низким и высоким напряжением, как на рисунке 2. Очевидно, что здесь невозможно отличить полное отсутствие сигнала от передачи нуля — в обоих случаях напряжение низкое. На помощь приходит импульсное кодирование, при котором 0 и 1 представлены не уровнями напряжения, а переходом от одного уровня к другому. Понятно, что это снимает проблему опознавания отсутствия данных, поскольку в последнем случае изменений вообще нет. Приведем в качестве примера широко известный манчестерский код [17], в котором единица кодируется перепадом от 1 к 0, а ноль, напротив, переходом от 0 к 1. Кроме того, существуют и другие способы кодирования передаваемых данных: например, можно различать 0 и 1 по длительности импульса или по частоте, т. е. по количеству перепадов, зарегистрированных за стандартный промежуток времени [7].

3.2. Логические схемы

С переходом к следующему слою мы отвлекаемся от физических основ кодирования данных и оперируем исключительно абстрактными величинами 0 и 1. Такой подход дает полное право применять к анализу работы логических схем сугубо теоретическую науку под названием «Алгебра логики».

Для обработки двоичных данных используются самые разнообразные логические схемы. Простейшими из них являются *логические элементы*, каждый из которых выполняет элементарную логическую операцию над одним битом.

Выбор базовых элементов для вычислительных устройств является самостоятельной задачей. Это в теории можно взять за основу любые понравившиеся логические элементы. А для практического конструирования необходимо принять во внимание, что эти элементы придется изготавливать в больших количествах. Отсюда, в частности, вытекает, что базовые элементы должны быть максимально простыми, а их разновидностей должно быть как можно меньше.

Классический набор логических элементов — это И, ИЛИ, НЕ. Доказано, что на их основе можно построить любую сколь угодно сложную логическую функцию. Но, как оказывается, совсем не обязательно брать за основу *три* элемента — достаточно *одного*, правда, комбинированного. Таким универсальным базисом обычно служит элемент И-НЕ (отрицание конъюнкции, так называемый *штрих Шеффера*): используя его, несложно построить любой из трех перечисленных выше

элементов [15, 18] и, следовательно, любую логическую функцию. Для электроники это означает, что любую логическую схему можно спроектировать на базе одного типового элемента. Попутно отметим, что схема И-НЕ требует меньше транзисторов, чем И [15].

Аналогичными свойствами обладает операция ИЛИ-НЕ (*стрелка Пирса*).

От традиционного обсуждения логических элементов, таблиц их истинности и практического использования (в частности, для сброса или установки необходимых битов) нетрудно перейти к более сложным логическим схемам, которые применяются в вычислительной технике. В школьном курсе обычно разбирают *триггер* и *однобитный сумматор* [1, 11]. Первый, будучи основой *регистра*, дает некоторое представление об устройстве памяти, а второй — об идеях работы *арифметико-логического устройства* (АЛУ). В качестве «информации к размышлению» для наиболее любознательных читателей призываем попутно задуматься над следующим фактом. Традиционный алгоритм сложения является последовательным, т. е. сначала складываются младшие разряды слагаемых, затем (с учетом возможного переноса) следующие, более старшие и т. д. Можно ли как-то ускорить этот процесс, обеспечив параллельное (одновременное) сложение нескольких разрядов сразу? Если потребуется подсказка, то наберите в поисковой машине запрос «*ускоренный перенос*».

3.3. Микроархитектура и архитектура

Поняв основные идеи, по которым строятся логические схемы компьютера, можно перейти к рассмотрению устройства компьютера в целом. Удобно обсуждать два следующих слоя вместе, поскольку они тесно связаны. К тому же уровень микроархитектуры для школьного курса информатики представляет не самый большой интерес и может считаться некоторым дополнением к важному уровню архитектуры.

Итак, что же такое архитектура? Самый простой, но весьма глубокий ответ можно сформулировать так: *архитектура — это то, как видит hardware программист*. Причем подчеркнем, что речь идет о профессиональном системном программисте, а не о студенте, начинающем изучать язык высокого уровня. Э. С. Таненбаум, предвидя возможные возражения по этому поводу, предлагает для четкости говорить о возможностях компилятора [15], но мы не будем усложнять наше обсуждение.

Для предотвращения путаницы надо обязательно подчеркнуть, что устоявшийся в компьютерной литературе термин «фон-неймановская архитектура» (в честь выдающегося математика Дж. фон Неймана, описавшего совместно с А. Берксом и Г. Голдстейном классические основы построения вычислительных устройств [28]) не вполне соответствует приведенному выше определению. Дело в том, что сформулированные в этом актуальном до сих пор труде фундаментальные принципы слишком общие и допускают различную конкретизацию. Приведем в качестве примера два варианта реализации архитектуры процессора — CISC и RISC [14], которые оба являются фон-неймановскими.

Что же в таком случае образует понятие архитектуры в общепринятом смысле? Прежде всего, это

система команд компьютера (точнее говоря, его процессора). Как образно пишет В. Столлингс, «областью, в которой “встречаются” программист и конструктор компьютеров, является набор машинных команд» [14]. Кроме того, сюда же необходимо добавить ряд связанных деталей, а именно: обрабатываемые данные и их форматы, структура собственной памяти процессора (его регистры), особенности организации ветвлений программ, методы адресации к памяти, способы обмена данными с внешними устройствами и некоторые другие важные вещи. Таким образом, мы видим, что *архитектура есть не что иное, как самое общее, очищенное от второстепенных подробностей, описание устройства компьютера*.

Кроме того, есть еще одно обстоятельство, которое делает архитектуру достойной рассмотрения в образовательном курсе. Дело в том, что архитектура — это *наиболее стабильное и неизменное* описание компьютера. Обоснование данного утверждения достаточно простое: пользователи программного обеспечения (ПО) хотят максимального сохранения работоспособности привычных для них программ на новых моделях компьютеров. А это значит, что разработка ПО должна опираться на какие-то базовые неизменные основы, которые как раз и есть архитектура.

Рассмотрим в качестве примера семейство процессоров Intel. До недавнего времени их совместимость обеспечивалась благодаря соблюдению тщательно документированной 32-битной архитектуры IA-32 (IA — сокращение от Intel Architecture). Все более интенсивное использование возможностей 64-битных процессоров породило новую, 64-битную, архитектуру Intel64 (не путать с IA-64). Переход к новой архитектуре не всегда гладкий: часть очень старых программ перестала работать под новыми 64-разрядными версиями Windows.

Важной частью архитектуры является кодирование команд, т. е. правила их записи в форме набора нулей и единиц. Увидеть хотя бы раз такое представление команд необычайно важно для мировоззрения, поскольку без подобных примеров все утверждения о том, что программа состоит из двоичных кодов, выглядят голословно. К сожалению, в распространенных Intel-совместимых процессорах кодирование устроено достаточно сложно. В то же время в более современной архитектуре RISC-V (которая из всех имеющихся вариантов лучше всего подходит для реализации в отечественной вычислительной технике [23]) разобраться в двоичной записи нескольких простейших инструкций процессора вполне возможно [24]. Материалы по стандарту RISC-V можно найти на официальном сайте (<https://riscv.org/>), а также на сайте российского Альянса RISC-V (<https://riscv-alliance.ru/>).

Как подчеркивал Э. С. Таненбаум [15], нет четкой границы между включением инструкции в систему команд и оформлением ее в виде служебной программы. Прекрасным примером этого может служить наличие или отсутствие в разных моделях процессоров встроенных инструкций целочисленного умножения и деления, а также вещественной арифметики.

Из сказанного выше о сути архитектуры отчетливо видно, что она есть *теоретическое* описание компьютера. Реализовать *на практике* такое обобщенное описание можно различными способами. И тут на

сцену выходит слой *микроархитектуры*, который является связующим звеном между логическими схемами и архитектурой. Здесь описывается, как именно в процессоре расположены и соединены друг с другом его функциональные узлы, необходимые для реализации архитектуры. Очевидно, что у каждой архитектуры может быть много различных микроархитектур, которые обеспечивают разную сложность, производительность и цену. Все они смогут выполнять одни и те же программы, но их внутреннее устройство может очень сильно отличаться.

Например, в неоднократно цитированном ранее учебнике [24] для современной свободно распространяемой архитектуры процессоров RISC-V рассматриваются три варианта реализации: одноктактный, многотактный и конвейерный процессоры. Обсуждение особенностей каждого варианта увело бы нас слишком далеко. Скажем только, что конвейерное (параллельное) выполнение нескольких команд очень широко применяется в современных процессорах.

Подчеркнем, что архитектура и микроархитектура разрабатываются разными специалистами, что важно с точки зрения профориентационных целей.

Итак, мы видели, что слой микроархитектуры является чисто технологическим и для учебного курса не так интересен. Зато слой архитектуры, напротив, очень важен. Мы вернемся к этому вопросу, когда закончим рассмотрение оставшихся слоев модели.

3.4. Операционная система

Теперь представим себе компьютер, у которого реализованы все предыдущие слои. Мы имеем машину, которая уже способна выполнить программу, но только такую, что представлена в виде кодов команд из нулей и единиц. (При некоторой фантазии можно допустить, что программа вводится в виде восьмеричных цифр, как это было в школьных компьютерах УКНЦ, где такой ввод обеспечивался встроенной программой-монитором.)

Попутно предложим читателям любопытный вопрос: а вы не задумывались над тем, что абсолютно верные утверждения «компьютер всегда работает по программе» и «при включении питания в ОЗУ нет никакой программы» как-то плохо согласуются между собой? Разгадка здесь в том, что в любом современном компьютере есть стартовая программа, которая хранится в виде «готового к употреблению» двоичного кода в специальном устройстве памяти — ПЗУ (постоянном запоминающем устройстве). Именно она и начинает выполняться в момент включения компьютера. Стартовая программа умеет немного: протестировать компьютер и определить наличие у него наиболее важных устройств, а затем инициировать процесс загрузки программного обеспечения в ОЗУ. И что же ищется и загружается в первую очередь? Конечно же, операционная система (ОС), которая и образует следующий слой, обеспечивающий работу компьютера.

Таким образом, в нашем рассмотрении появляется первый слой, который не встроен жестко в конструкцию компьютера, а *загружается* в него. Иными словами, у пользователя теперь есть выбор, какую именно операционную систему поставить (а может, и какие несколько ОС).

Можно ли записать ОС в ПЗУ в качестве стартовой программы? Технически при современных возможностях производства микросхем это решаемая задача. Но так не делают, чтобы не потерять гибкость. И дело даже не столько в том, что продавать компьютеры с «защитой» версией Windows 10 или Linux Ubuntu станет сложнее. Гораздо важнее, что существенно усложнится обновление ОС: при покупке, скажем, новейшего принтера вам, скорее всего, придется вызывать мастера для перепрограммирования ПЗУ с операционной системой. Не самая привлекательная перспектива...

Зачем нужна ОС и почему без предустановленной операционной системы практически не продаются компьютеры?

История развития ОС весьма поучительна. В первом поколении вычислительной техники операционных систем не было [16]. Программу загружал человек, а после ее выполнения и выдачи результатов машина останавливалась в ожидании ввода новой задачи. Сначала такой режим был приемлемым, тем более что лампы, на которых тогда были построены ЭВМ, часто выходили из строя, из-за чего требовался ремонт. Но с переходом ко второму поколению машины постепенно стали более надежными, так что люди начали задумываться по поводу долгого простоя дорогостоящего оборудования при переходе от одной задачи к другой. Появился так называемый *пакетный режим*, при котором сразу после завершения вычислений одной задачи, пока печатались результаты, автоматически загружалась следующая. Руководили этим процессом специальные управляющие *программы-мониторы*. Так сложилось, что третье поколение представляло собой огромные машины для коллективного пользования. Для них организация многозадачной работы стала еще более насущной, что и привело к появлению операционных систем, управляющих эффективным прохождением задач через машину. Заметим, что хотя компьютеры четвертого поколения в основном персональные, т. е. ими обычно пользуется только один человек, но они по-прежнему многозадачны: одновременно принимают набираемый текст, параллельно проверяя его синтаксис и стиль, воспроизводят музыку, «закачивают» обновления (часто даже без вашего ведома!) и делают параллельно еще много чего. Итак, *стремление к эффективному выполнению задач* служит главной причиной появления ОС.

Была и еще одна причина появления операционных систем. О ней вспоминают реже, что не снижает ее значимости. Дело в том, что до появления ОС все внешние устройства были достаточно примитивными: устройства ввода с перфокарт, перфолент и магнитных лент имели ограниченные возможности, и управление ими легко программировалось. Ситуация коренным образом изменилась с появлением магнитных дисков, где доступ к данным осуществляется значительно сложнее. Для упрощения ситуации была придумана идеология *файловой системы*, когда достаточно указать символическое имя файла, а система, пользуясь своими служебными данными, найдет и прочтает с диска нужные сектора. Кстати говоря, порядок чтения секторов тоже можно оптимизировать, что способна делать ОС.

Связь ОС с наличием дисковых накопителей наиболее отчетливо видна на примере микро-ЭВМ

(персональных компьютеров), начиная с самой первой операционной системы CP/M [16]. В то время даже существовал термин «дисковая операционная система», фиксирующий факт такой связи.

Помимо названных выше функций реализации многозадачности и поддержки файловой системы операционная система имеет и еще целый ряд важных функций. Чтобы не загромождать изложение, выделим среди них только одну, имеющую непосредственное отношение к обсуждаемой многослойной модели. Речь идет о создании интерфейса между слоем прикладного ПО и аппаратным обеспечением; часто такой интерфейс называют API — Application Program Interface (программный интерфейс приложения). Благодаря ему прикладной программист, которому требуется, скажем, напечатать на бумаге текст, просто вызывает стандартную функцию API. А все необходимые действия по управлению выводом на конкретную разновидность принтера берет на себя ОС.

Таким образом, операционная система может рассматриваться как расширенная (еще одним слоем) машина, управлять которой значительно проще, чем это делается на уровне «железа». «Операционная система превращает уродливое аппаратное обеспечение в красивые абстракции», как несколько иронично гласит подпись под рисунком 1.2 в книге [16].

3.5. Языки программирования

На данном уровне обсуждения мы имеем машину, пригодную для разработки программного обеспечения. Сразу заметим, что возможность сохранения текста программы и результатов ее трансляции в файлы, которая обеспечивается слоем ОС, играет здесь не последнюю роль.

Очевидно, что для разработки новой программы необходима программа-ассистент, которая помогает преобразовать написанный человеком текст на некотором языке в набор нулей и единиц, понятный процессору. Такая служебная программа называется *транслятором*, т. е. переводчиком (с некоторого внешнего языка на «единственно правильный» язык машинного кода из 0 и 1). Существуют две базовые стратегии трансляции — *интерпретация* и *компиляция*; возможны также их комбинации. Вопрос этот многократно описан в литературе, так что сейчас мы не будем на нем останавливаться.

Языков программирования — великое множество. Наиболее простой из них (не случайно его называют *языком низкого уровня*) — **ассемблер**. Он позволяет записать последовательность машинных команд не в виде 0 и 1, а как некоторый более удобный для человека текст. Подчеркнем, что каждая строка ассемблерной программы преобразуется в двоичный код *одной* машинной инструкции.

Как выглядит запись команды на ассемблере? Например, так:

ADD x31, x6, x7

Перед нами команда сложения содержимого регистров x6 и x7, результат которой сохраняется в x31.

Часто ассемблер трактуют как язык, в котором двоичные коды заменяются определенным текстом. При этом говорится, что осмысленное (английское) слово

ADD гораздо легче запомнить, чем двоичное число. Не будем сейчас дискутировать по поводу простоты запоминания, особенно для тех, кто не знает английский (для примера взгляните на пару реальных команд: AUIPC и SLTIU). Подчеркнем только, что главное достоинство ассемблера совсем не в этом — гораздо важнее то, что адреса ячеек памяти можно заменять символическими именами (их называют *метками* или *идентификаторами*). Дело в том, что заранее удачно предугадать конкретные адреса в ОЗУ не так просто; особенно трудно бывает переделать программу в случае, когда была допущена ошибка и для ее устранения требуется переместить часть кода по памяти. Зато ассемблер справляется с этой задачей прекрасно. Он сначала планирует размещение двоичного кода в памяти, а затем составляет таблицу соответствия между метками и адресами памяти. Пользуясь этой таблицей, ассемблер быстро и безошибочно подставляет вместо меток актуальные числовые адреса. Естественно, что в случае любого изменения в тексте программы описанная процедура без труда повторяется заново.

Судите сами, насколько это важно. Если поставить метку в строке, где описывается место памяти для хранения *данных*, то эта метка фактически станет именем *переменной*. Если же помечена ячейка в *программе*, то на это место можно сделать переход, в том числе и условный. В результате получаем средство для организации *развилки, цикла или процедуры*, т. е. любой алгоритмической структуры.

Ассемблер незаменим в двух случаях: во-первых, когда необходимо написать эффективный и быстро работающий машинный код, а во-вторых, если нужен доступ к аппаратной части компьютера. Как вы догадываетесь, речь идет, в первую очередь, о драйверах и системных программах.

Конечно, и недостатки у ассемблера есть. Прежде всего, он *машинно-зависимый*, т. е. для каждого семейства машин свой. (В скобках заметим, что, зная и понимая один ассемблер, перейти к другому не так сложно.) Кроме того, программирование на ассемблере весьма трудоемко.

Поэтому естественным образом возникли **языки высокого уровня**, такие как Pascal, Java, Python и др. В них вводятся (машинно-независимые) правила записи алгоритмических конструкций, таких как ветвление, цикл, процедура и т. д. Компилятор анализирует текст программы и составляет подходящие блоки из машинных инструкций, которые реализуют соответствующие структуры для данного процессора. В итоге получается двоичный код, который сохраняется в виде исполняемого файла. Последний, в свою очередь, может быть запущен операционной системой.

Имея средства для программирования, можно создавать любое прикладное программное обеспечение. Но с точки зрения нашего обсуждения имеет особый интерес разработка системного ПО, особенно операционных систем. Трудность здесь в том, что для работы стандартного компилятора нужна ОС, а для реализации ОС нужен компилятор. Как же быть? Вот, например, как описывается процедура рождения ядра ОС Unix для машины PDP-7 [4]. Сначала на *другой машине* GE-635 готовилась бумажная лента с ассемблерным текстом про-

граммы, читаемая PDP-7. Затем эту ленту переносили и вводили в машину PDP-7. Введенная исходная программа транслировалась «автономным» ассемблером, который генерировал выходной файл и запускал его. Так продолжалось до тех пор, пока не были завершены примитивное ядро Unix, редактор, простой командный интерпретатор и несколько самых необходимых утилит (вроде команды Unix для копирования файлов).

Но затем Д. Ритчи придумал принципиально новый подход: он создал язык Си, который по возможностям не уступал ассемблеру, но в то же время содержал базовые конструкции языка высокого уровня (такой язык называют *машинно-ориентированным языком промежуточного уровня*, т. е., проще говоря, сильно расширенным ассемблером). В результате разработка ОС Unix стала возможной на языке Си. Язык этот и по сей день широко используется для написания системного ПО, в том числе операционных систем.

Сам Д. Ритчи позднее писал про свое детище: «Он, безусловно, удовлетворил потребности в таком языке реализации систем, который был достаточно эффективным, чтобы заменить ассемблер, и в то же время достаточно абстрактным и гибким, чтобы описывать алгоритмы и взаимодействия в широком спектре программных сред» (цит. по [3]).

Приведенные исторические подробности показывают, с каким трудом иногда приходится добавлять новые слои программного обеспечения.

3.6. Прикладное ПО

Наконец, последний, знакомый каждому пользователю слой прикладного программного обеспечения. Именно с ним чаще всего имеют дело в повседневной жизни. По этой самой причине особой необходимости в описании данного слоя нет. Заметим только, что для того, чтобы ваша любимая прикладная программа была реализована и заработала, необходимы все описанные ранее слои.

* * *

Итак, мы рассмотрели все существующие на сегодняшний день слои. Появятся ли новые? Несомненно! Уже сейчас видны контуры мощного ИИ-слоя (слоя искусственного интеллекта), который по мере своего развития почти наверняка «спрячет» под собой все программные слои. Быть может, тогда станет излишним учиться программировать и осваивать графический интерфейс прикладного ПО.

Интересно вспомнить, что современные достижения ИИ очень похожи на то, что обещали сделать (и до конца в свое время не сумели) японские ученые в ЭВМ пятого поколения: управление голосом, формулировка результата на естественном языке, самостоятельный выбор метода решения задачи и т. д. Впрочем, не будем опережать время и забегать слишком далеко вперед...

4. Сопоставление с содержанием школьного курса

Теперь, когда мы познакомились со всеми слоями модели компьютера, посмотрим, как эту модель целесообразно «спроектировать» на содержание школьного

Таблица

Сопоставление слоев модели и содержания школьного курса информатики

Слой	VII—IX базовый [19]	VII—IX углубленный [20]	X—XI базовый [21]	X—XI углубленный [22]
Физическое кодирование 0 и 1	Дискретность данных; двоичный код (алфавит); бит — двоичный разряд; логические элементы		+ передача : сигнал, кодирование; хранение информации в ОЗУ	+ дискретные сигналы; дискретизация информации в цифровых системах
Логические схемы	Логические элементы; логические основы компьютера; поколения ЭВМ	+ сумматор	+ триггер; логическая схема по выражению и наоборот	+ многоразрядный сумматор; <i>микросхемы и технология их производства</i>
Микроархитектура (*)	Современные тенденции; параллельные вычисления; суперкомпьютеры; двоичная арифметика, логические операции (АЛУ)		+ многопроцессорные системы	+ автоматическое выполнение программы; шина
Архитектура	Основные функциональные блоки компьютера; двоичное представление (форматы) данных		+ хранение, объем памяти (байт — ячейка); представление вещественных чисел (форматы)	+ архитектура Неймана и гарвардская; контроллеры, прямой доступ к памяти; переполнение, длинная арифметика
Операционная система	Файлы и папки; работа с ними	+ принципы построения файловых систем	+ операционная система (как разновидность ПО); администрирование	+ утилиты и драйверы; принципы размещения и именования файлов; шаблоны групп файлов
Языки программирования	Язык; система программирования; базовые конструкции; отладка		+ типы данных	+ компиляция и интерпретация; ООП; RAD-среды; обзор языков
Прикладное ПО	Состав ПО; примеры программ; практическая работа с прикладным ПО			

курса. Позиция автора заключается в том, что образованный ученик должен получить некоторое (пусть минимально необходимое) представление о *каждом* из перечисленных слоев.

Проанализируем, как обстоит с этим дело в сложившемся в школе курсе информатики. Для этого внимательно прочитаем федеральные рабочие программы 2023 года, доступные на сайте «Единое содержание общего образования» [19–22]. Перечисленные в них вопросы, связанные с каждым из выделенных слоев, выпишем в таблицу. В ней выделены столбцы, соответствующие требованиям к содержанию базового и углубленного курсов VII—IX и X—XI классов, что соответствует четырем действующим федеральным рабочим программам.

Текст, помеченный в таблице серым фоном, соответствует вопросам, где *можно* дать сведения по соответствующему слою. Но именно *можно*, особенно если знать о существовании этих связей. Например, рассказ о двоичности данных можно сопроводить сведениями об идеях кодирования 0 и 1 (но программы на этом не настаивают и даже не намекают на это). Поколения ЭВМ строятся на основе элементной базы, так что здесь будет уместно дополнение о том, что такое микропроцессоры и как их производят. Двоичное (и особенно байтовое) представление данных существенно влияет

на архитектуру процессора. Мимоходом заметим, что даже в программе углубленного курса не упоминается о существовании в процессоре специально выделенного блока для обработки вещественных чисел — *математического сопроцессора*. А это важно, поскольку без такого обсуждения не каждый школьник обратит внимание на фундаментальный факт, что обработка вещественных чисел существенно отличается от обработки целых.

Из таблицы отчетливо видно, что «все серое» сосредоточено в верхней части таблицы, т. е. именно здесь возможна потеря важных сведений. Напрашивается вывод, что школьный курс слишком «накренен» в сторону практических применений компьютера и слабо освещает базовые идеи его устройства. Если следить за эволюцией содержания школьного курса, то это не удивляет.

Проанализируем содержание приведенной таблицы подробнее. Начнем с нескольких принципиальных замечаний.

Во-первых, только в углубленном курсе X—XI классов, и то курсивом (необязательный материал), есть явная ссылка на «микросхемы и технологии их производства». Иными словами, при изучении современного курса информатики даже просто сообщить школьникам о существовании такой отрасли, как производство чипов, по мнению программы, совсем не обязательно. Не

тут ли лежит корень нашего отставания в данной без преувеличения стратегической области?

Во-вторых, из программы начисто исчезло фундаментальное понятие «система команд». Причем ладно бы речь шла только о системе команд процессора, но и у исполнителей она тоже не упоминается! Между тем в традиционной методике преподавания информатики было принято, что «основной характеристикой исполнителя <...> является система команд исполнителя (СКИ)» [8]. Отсюда же, по-видимому, вытекает и невнимание к архитектуре компьютера, которая всегда начинается с системы команд процессора.

В-третьих, слово «многоядерный» также начисто отсутствует в программах курса. Важно ли это? Думается, в современных условиях очень даже важно. Купить в магазине новый компьютер с *одноядерным* процессором уже невозможно, а понятных разъяснений, как работает *многоядерный* процессор и в чем его реальные преимущества, как говорится, «днем с огнем...». И это при том, что вопрос о параллельных вычислениях в программе прописан в явном виде.

И, наконец, еще одно замечание общего плана, не привязанное к конкретным строчкам программы, но навеянное общим анализом содержания программ [19–22]. Стратегия обучения школьников такова, что нынешний курс информатики незаметно распадается на две относительно самостоятельных части. С одной стороны, строгая абстрактная теория (двоичная система и арифметика для нее, алгебра логики и т. д.), хорошо подходящая для проверки на ЕГЭ. С другой — практические операции (работа с файлами средствами ОС и с прикладными пакетами, написание простых программ на языке высокого уровня и т. д.). Без сомнения, при тщательном обдумывании материалов урока знающий учитель сможет построить «мостики» между этими «половинками». Но располагает ли к этому программа?

Критиковать легче, чем предлагать пути преодоления недостатков. Поэтому приведем вполне конкретный пример, как видятся пути сближения названных частей курса. В частности, очень полезно, наглядно и интересно продемонстрировать школьникам на практике конкретные случаи хранения в памяти двоичных данных. С помощью языков программирования это удастся сделать, экспериментируя с (двоичным!) содержимым переменных [6, 13, 25]. Вероятно, не очень сложно написать некие учебные приложения, которые отображают ту или иную информацию из памяти в двоичной форме. (Хотя это будет менее убедительно: откуда известно, что программа выводит данные именно в том виде, как они лежат в ОЗУ?) В любом случае, требуется внимание к проблеме, которое пока не чувствуется.

Отмеченные особенности, по нашему мнению, говорят о недостатках в стратегическом планировании современного школьного курса.

Еще раз обратимся к таблице и поищем в ней положительные моменты. Есть ли удачные совпадения между программой и изложенной многослойной моделью? Да, несомненно! Прежде всего, это три последние строки таблицы (операционная система и далее). Еще вполне представительно выглядит слой логических схем.

Если сосредоточить внимание только на последнем столбце, то обнаруживается неплохое соответствие

по всем слоям. Исходя из этого, можно сказать, что углубленный курс X—XI классов [22] охватывает рассматриваемую многослойную модель вполне удовлетворительно.

На наш взгляд, наиболее «обделен вниманием» важный слой архитектуры. Поэтому обсудим его подробнее.

Начнем с того, что слой служит связующим звеном между hardware и software, что имеет важное методологическое значение для понимания взаимодействия между названными базовыми компонентами компьютера. Недостаточное внимание к вопросам, связанным с указанной тематикой, может приводить к произвольному разделению курса информатики на две слабо связанные части: теоретическую (про двоичную систему и логические элементы) и практическую (работа с пакетами и основы программирования) [5]. В результате многие ученики теряют из виду естественную связь между hard и soft и порой задают по поводу теории традиционный вопрос: «Зачем (или кому) это нужно?» И в самом деле: если на уроках не обращалось внимания на кодирование программы и данных, то для чего было изучать двоичную систему? Тем более непонятно, что с практической точки зрения дают разговоры об основах двоичной арифметики, если даже в [22] отсутствует термин «арифметико-логическое устройство».

Наконец, несколько личных впечатлений по поводу реакции аудитории на выступления по обсуждаемой тематике на нескольких последних всероссийских конференциях по преподаванию информатики. С сожалением можно констатировать, что предложения о необходимости знакомства в школьном курсе информатики с отдельными вопросами устройства компьютеров и современными специальностями, связанными с производством вычислительной техники, часто воспринимаются негативно. А как же без этого можно рассчитывать на развитие в этой области? Согласитесь, что все-таки надо не только ориентировать школьников на профессии, связанные с программированием и применением прикладного ПО, но и обращать внимание учащихся на перспективность работы в областях, связанных с проектированием и производством новых вычислительных устройств. Политика «Intel inside» по известным причинам дает сбой, а значит, жизненно необходимо ее чем-то заменить.

В качестве аргументации сошлемся на итоговое решение Всероссийской конференции «Преподавание информационных технологий в Российской Федерации», проходившей в 2024 году в Твери [12]. В пункте 23 там записано следующее: «Рекомендовать Минпросвещения России включить в школьную программу по учебному предмету “Информатика” (углубленный уровень) вопросы по архитектуре компьютера, функциям операционных систем, концепции и основным характеристикам языков программирования высокого и низкого уровней, предметно-ориентированных языков».

Завершая обсуждение, отметим, что слой микроархитектуры является скорее технологическим и поэтому менее значим. Если бы не высказанные выше соображения по поводу ориентации школьников на различные профессии в производстве hardware, то, возможно, данный слой можно было бы даже объявить частью слоя архитектуры.

5. Выводы

- Существуют минимум две важные причины, по которым стоит познакомиться с многослойной моделью компьютера и использовать ее в преподавании информатики. Во-первых, обострившаяся необходимость в разработке и развитии производства отечественной вычислительной техники и ее элементной базы; отсюда в профориентационных целях требуется дать школьникам более полное представление об этой стратегически важной отрасли. Во-вторых, учет базовых идей модели может оказаться полезным для систематизации знаний у школьников и формирования у них целостной картины предмета.
- Многослойный принцип действительно облегчает изучение многогранного и объемного материала о функциональных основах работы компьютера. При грамотном использовании он помогает увидеть и зафиксировать связи между выделенными слоями. Образно говоря, описываемый подход может стать *стержнем*, объединяющим отдельные темы курса информатики (кажущиеся, на первый взгляд, независимыми).
- «Потеря» хотя бы одного из слоев приводит к разрушению существующих взаимосвязей между темами учебного материала.
- Как следствие предыдущего, отрицательно влияет на систематичность знаний учащихся исключение из рассмотрения слоя компьютерной архитектуры, которая служит связующим звеном между программной и аппаратной частями. Это именно тот слой, где наиболее отчетливо проявляются связи между теорией (двоичная система счисления и др.) и лежащими выше и применяемыми на практике программными слоями. С указанной точки зрения полезно искать способы наглядной демонстрации двоичного представления различных типов данных и команд программы непосредственно в памяти компьютера или в файлах.
- В свете возможного направления развития отечественной вычислительной техники, вероятно, стоит познакомить школьников с существованием свободно распространяемого стандарта архитектуры компьютера под названием RISC-V. Можно порекомендовать показать несколько его наиболее простых команд, их кодирование и использование.
- Полезно также увеличить внимание к слою физического кодирования 0 и 1, учитывая его большое значение для мировоззрения (а также с точки зрения межпредметных связей).
- Многослойная модель позволяет просто и наглядно показать одну из причин постоянной потребности в росте производительности компьютерной техники: каждый добавляемый слой (или усложнение имеющегося) увеличивает время выполнения программы.
- Важно сказать ученикам, что, помимо упрощения освоения, рост числа слоев между пользователем и собственно компьютером имеет и отрицательные стороны: теряются возможности нижележащих

слоев и падает эффективность взаимодействия прикладного ПО с аппаратной частью. Снижается также понимание логики поведения машины.

Список источников

1. Андреева Е. В., Босова Л. Л., Фалина И. Н. Математические основы информатики. Элективный курс: учебное пособие. М.: БИНОМ. Лаборатория знаний, 2005. 328 с. EDN: WLQUBT.
2. Бершадский М. Е. Педагогическая диагностика уровня понимания // Педагогические измерения. 2012. № 3. С. 60–88. EDN: OFVQZU.
3. Богатырев Р. Летопись языков. Си // Мир ПК. 2001. № 8. <https://www.osp.ru/pcworld/2001/08/162103>
4. Деннис Макалестэйр Ритчи. Между Unix и C // Хабр. 11 сентября 2024. <https://habr.com/ru/companies/timeweb/articles/842238/>
5. Еремин Е. А. Анализ содержательной линии «Компьютер» курса информатики с применением компьютерных средств представления знаний // Информатика («Первое сентября»). 2008. № 9. С. 8–18.
6. Еремин Е. А. Изучение средствами Delphi способов хранения в компьютерной памяти различных данных // Информатика («Первое сентября»). 2010. № 19. С. 4–23.
7. Еремин Е. А. Стандарт MSX для записи на МЛ // Информатика и образование. 1992. № 5–6. С. 56–58. EDN: TIFBDN.
8. Лапчик М. П., Семакин И. Г., Хеннер Е. К. Методика преподавания информатики: учеб. пособие для студ. пед. вузов. М.: Академия, 2001. 624 с. EDN: VFVKNW.
9. Маркс К. Капитал. М.: ИПЛ, 1969. Т. 1. С. 189.
10. Олифер В. Г., Олифер Н. А. Компьютерные сети, принципы, технологии, протоколы. СПб.: Питер, 2005. 864 с.
11. Поляков К. Ю., Еремин Е. А. Информатика. Углубленный уровень: учебник для 10 класса. В 2 ч. Ч. 1. М.: БИНОМ. Лаборатория знаний, 2013. 344 с. EDN: TUVVWD.
12. Решение Двадцать второй открытой Всероссийской конференции «Преподавание информационных технологий в Российской Федерации» (2024 год) // Преподавание информационных технологий в Российской Федерации. <https://it-education.ru/conf2024/agenda/reshenie.php>
13. Романов С. Использование памяти в Python // Хабр. 16 сентября 2013. <https://habr.com/ru/post/193890/>
14. Столлингс В. Структурная организация и архитектура компьютерных систем. М.: Вильямс, 2002. 896 с.
15. Таненбаум Э. С. Архитектура компьютера. СПб.: Питер, 2003. 704 с.
16. Таненбаум Э. С., Бос Х. Современные операционные системы. СПб.: Питер, 2015. 1120 с.
17. Таненбаум Э. С., Фимстер Н., Уэзеролл Д. Компьютерные сети. СПб.: Питер, 2023. 992 с.
18. Токхейм Р. Основы цифровой электроники. М.: Мир, 1988. 392 с.
19. Федеральная рабочая программа основного общего образования. Информатика (базовый уровень) (для 7–9 классов образовательных организаций) // Единое содержание общего образования. https://edsoo.ru/wp-content/uploads/2023/08/15_ФРП_Информатика-7-9-классы_база.pdf
20. Федеральная рабочая программа основного общего образования. Информатика (углубленный уровень) (для 7–9 классов образовательных организаций) // Единое содержание общего образования. https://edsoo.ru/wp-content/uploads/2023/08/16_ФРП_Информатика_7-9-классы_угл.pdf
21. Федеральная рабочая программа среднего общего образования. Информатика (базовый уровень) (для 10–11 классов образовательных организаций) // Единое содержание общего образования. https://edsoo.ru/wp-content/uploads/2023/08/21_ФРП_Информатика_10-11-классы_база.pdf
22. Федеральная рабочая программа среднего общего образования. Информатика (углубленный уровень) (для 10–11 классов образовательных организаций) // Единое со-

держание общего образования. https://edsoo.ru/wp-content/uploads/2023/08/22_ФРП_Информатика-10-11-классы_угл.pdf

23. Фролов В. А., Галактионов В. А., Санжаров В. В. Исследование технологии RISC-V // Труды ИСП РАН. 2020. Т. 32. № 2. С. 81–98. EDN: FYLLGM. DOI: 10.15514/ISPRAS-2020-32(2)-7.

24. Харрис С. Л., Харрис Д. Цифровая схемотехника и архитектура компьютера: RISC-V / пер. с англ. В. С. Яценкова, А. Ю. Романова; под ред. А. Ю. Романова. М.: ДМК Пресс, 2021. 810 с.

25. Харрисон М. Как устроен Python. Гид для разработчиков, программистов и интересующихся. СПб.: Питер, 2019. 272 с.

26. Черняк Л. С. Чему учиться? // Открытые системы. СУБД. 2003. № 2. <https://www.osp.ru/os/2003/02/182640>

27. Ben-Ari M. Constructivism in computer science education // Journal of Computers in Mathematics and Science Teaching. 2001. Vol. 20. No 1. P. 45–73.

28. Burks A. W., Goldstine H. H., Neumann J. Preliminary discussion of the logical design of an electronic computing instrument. Princeton, NJ: Institute for Advanced Study, 1946. P. 1–33. (Русский перевод: Беркс А., Голдстейн Г., Нейман Дж. Предварительное рассмотрение логической конструкции электронного вычислительного устройства // Кибернетический сборник. № 9. С. 7–67. М.: Мир, 1964.)

29. Du Boulay B. Some difficulties of learning to program // Journal of Educational Computing Research. 1986. Vol. 2. No 1. P. 57–73. DOI: 10.2190/3LFX-9RRF-67T8-UVK9.

30. González R. L., García L. M. C., García M. M., Masa J. A. Possibilities of “Nuclear Concepts Theory” on educational research, a review // Pixel International Conferences. https://conference.pixel-online.net/conferences//edu_future/common/download/Paper_pdf/SOE33-Gonzalez,Garcia,Garcia,Masa.pdf

DOI: 10.32517/2221-1993-2025-24-1-64-68

**К. А. Чундерова***Нижнетагильский государственный социально-педагогический институт (филиал) Российского государственного профессионально-педагогического университета, г. Нижний Тагил, Свердловская область, Россия***Е. С. Васева***Нижнетагильский государственный социально-педагогический институт (филиал) Российского государственного профессионально-педагогического университета, г. Нижний Тагил, Свердловская область, Россия;
Средняя общеобразовательная школа № 61, г. Нижний Тагил, Свердловская область, Россия*

ПРИМЕНЕНИЕ ИНТЕРАКТИВНЫХ МЕТОДОВ ОБУЧЕНИЯ ДЛЯ РАЗВИТИЯ ЛОГИЧЕСКОГО МЫШЛЕНИЯ ШКОЛЬНИКОВ ПРИ ОБУЧЕНИИ ТЕМЕ «ЭЛЕМЕНТЫ АЛГЕБРЫ ЛОГИКИ»

Аннотация

В статье актуализируется необходимость применения интерактивных методов обучения при изучении темы «Элементы алгебры логики» в базовом курсе информатики основной школы. Интерактивные методы обучения рассматриваются как средство активизации познавательной деятельности обучающихся, мотивации к обучению, создания условий для вовлеченности школьников в изучение предмета. Выбор темы обусловлен ее значимостью для формирования универсальных учебных действий, являющихся составляющими логического мышления и необходимых для изучения других тем курса информатики, а также других учебных предметов. Целью настоящей статьи являлось описание методических аспектов использования интерактивных методов обучения, направленных на повышение уровня логического мышления обучающихся, при изучении темы «Элементы алгебры логики». Рассматриваются возможности конкретных методов обучения при формировании отдельных составляющих логического мышления. Представлены методические рекомендации применения интерактивных методов обучения в процессе проведения одного из уроков по теме «Элементы алгебры логики».

Ключевые слова: базовый курс информатики, элементы алгебры логики, интерактивный метод обучения, логическое мышление, операции логического мышления.

Информация об авторах

Чундерова Ксения Андреевна, студентка 5-го курса факультета естествознания, математики и информатики, Нижнетагильский государственный социально-педагогический институт (филиал) Российского государственного профессионально-педагогического университета, г. Нижний Тагил, Свердловская область, Россия; *e-mail:* kchunderovaksusha@gmail.com

Васева Елена Сергеевна, канд. пед. наук, доцент; доцент кафедры информационных технологий и физико-математического образования, факультет естествознания, математики и информатики, Нижнетагильский государственный социально-педагогический институт (филиал) Российского государственного профессионально-педагогического университета, г. Нижний Тагил, Свердловская область, Россия; учитель информатики, средняя общеобразовательная школа № 61, г. Нижний Тагил, Свердловская область, Россия; *e-mail:* e-s-vaseva@mail.ru

K. A. Chunderova

Nizhny Tagil State Socio-Pedagogical Institute (branch) of Russian State Vocational Pedagogical University, Nizhny Tagil, Russia

E. S. VasevaNizhny Tagil State Socio-Pedagogical Institute (branch) of Russian State Vocational Pedagogical University, Nizhny Tagil, Russia;
School 61, Nizhny Tagil, Russia

USING INTERACTIVE TEACHING METHODS FOR THE DEVELOPMENT OF LOGICAL THINKING OF SCHOOLCHILDREN WHEN TEACHING THE THEME "ELEMENTS OF ALGEBRA OF LOGIC"

Abstract

The article actualizes the necessity of using interactive teaching methods when studying the theme "Elements of algebra of logic" in the basic informatics course of secondary school. Interactive teaching methods are considered as a means of activating the cognitive activity of students, motivating them to learn, creating conditions for involving students in studying the subject. The choice of the theme is due to its importance for the formation of universal learning actions that are components of logical thinking and are necessary for studying other themes of the informatics course, as well as other subjects. The purpose of this article was to describe the methodological aspects of using interactive teaching methods aimed at improving the level of logical thinking of students when studying the theme "Elements of algebra of logic". The possibilities of specific teaching methods in the formation of individual components of logical thinking are considered. Methodological recommendations for the use of interactive teaching methods in the process of conducting one of the lessons on the theme "Elements of algebra of logic" are presented.

Keywords: basic informatics course, elements of algebra of logic, interactive teaching method, logical thinking, operations of logical thinking.

Information about the authors

Kseniya A. Chunderova, 5th grade student, Faculty of Natural Science, Mathematics and Informatics, Nizhny Tagil State Socio-Pedagogical Institute (branch) of Russian State Vocational Pedagogical University, Nizhny Tagil, Sverdlovsk Region, Russia; e-mail: kchundervaksusha@gmail.com

Elena S. Vaseva, Candidate of Sciences (Education), Docent, Associate Professor at the Department of Information Technologies, and Physics and Mathematics Education, Faculty of Natural Science, Mathematics and Informatics, Nizhny Tagil State Socio-Pedagogical Institute (branch) of Russian State Vocational Pedagogical University, Nizhny Tagil, Sverdlovsk Region, Russia; School 61, Nizhny Tagil, Sverdlovsk Region, Russia; e-mail: e-s-vaseva@mail.ru

1. Введение

На сегодняшний день общество предъявляет высокие требования к подрастающему поколению. В соответствии с новым ФГОС основного общего образования, обучающиеся должны овладеть навыками выполнения базовых логических действий, также отмечается необходимость формирования навыков анализа, рассуждения и решения проблем [19]. Тема из курса информатики основной школы, которая напрямую связана с данными аспектами, — «Элементы алгебры логики».

Эта тема — одна из важнейших в курсе информатики. Система понятий и навыки, формируемые при изучении данной темы, определяют успешность обучения по другим учебным предметам, возможность освоения последующих разделов информатики.

Процесс обучения теме «Элементы алгебры логики» позволяет создать условия для формирования универсальных учебных действий, таких как:

- выявление и характеристика существенных признаков объектов;
- установление существенного признака классификации, основания для обобщения и сравнения;
- выявление причинно-следственных связей;
- формулирование выводов с использованием дедуктивных и индуктивных умозаключений, умозаключений по аналогии;
- формулирование гипотез о взаимосвязях;
- выбор, анализ, систематизация и интерпретация информации разных видов и форм представления и т. д.

Данные познавательные УУД являются неотъемлемой частью логического мышления.

Так, например, изучение раздела «Алгоритмы и программирование» невозможно осуществить в полной мере при низком уровне логического мышления, поскольку здесь необходимо умение анализировать задачи, строить последовательности действий, определять условия и принимать решения на основе логических операций. Поэтому при изучении темы «Элементы алгебры логики» важно организовать учебный процесс таким образом, чтобы обучающиеся занимали максимально активную позицию и были заинтересованы в изучении предлагаемого материала. Использование интерактивных методов обучения как современной образовательной технологии будет способствовать повышению мотивации к обучению, созданию условий для вовлеченности в изучение темы «Элементы алгебры логики», что сделает процесс обучения более эффективным. Обучающиеся в большей степени начнут воспринимать окружающую действительность с позиции возможности применения логических рассуждений при решении как задач других предметных

областей, так и обыденных ситуаций, что, в свою очередь, и станет результатом развития логического мышления.

Цель данной статьи — описание методических аспектов использования интерактивных методов обучения, направленных на повышение уровня логического мышления обучающихся при изучении темы «Элементы алгебры логики».

2. Интерактивные методы обучения в школьном курсе информатики

Интерактивным методам обучения сегодня отводится особое место в образовании. **Интерактив** (от *англ.* interaction — взаимодействие) обозначает взаимное действие. **Интерактивность** подразумевает способ взаимодействия в режиме диалога с кем-либо (педагогом, другим обучающимся) или чем-либо (компьютером, мобильным устройством, цифровым образовательным ресурсом и т. д.), направленный на самоактивность и взаимную активность участников образовательных отношений [5, 6].

Т. С. Панина, Л. Н. Вавилова дают следующее определение: «Интерактивное обучение — это способ познания, осуществляемый в формах совместной деятельности обучающихся» [10]. Интерактивное обучение предполагает обучение в формате диалога, в котором происходит взаимодействие как учителя и ученика, так и учеников между собой. С. Г. Пузиновская, О. А. Счеснович отмечают, что применение интерактивных методов организации работы в парах позволяет создать условия для формирования навыков учебного сотрудничества [14]. Многие авторы отмечают, что независимо от учебного предмета при применении интерактивных методов обучения происходит обмен знаниями, идеями, опытом как между учениками, так и между учителем и учеником и при условии создания атмосферы доброжелательности и поддержки обучающиеся не только получают новые знания, но и формируют навыки сотрудничества, развивают познавательную активность [4, 7, 20]. И. В. Плаксина пишет, что именно интерактивные методы обучения в большей степени соответствуют личностно-ориентированному подходу в обучении, все обучающиеся оказываются вовлеченными в познавательный процесс, имеют возможность вносить свой собственный вклад в решение учебных проблем, оказываются в ситуации необходимости понимания, анализа своих знаний, мыслей, возможностей [11].

В настоящее время описано достаточно большое количество интерактивных методов обучения, в обучении информатике могут применяться многие из них. Е. В. Белошапкина назначение применения интерактивных методов обучения на уроках информатики видит в создании условий приобретения социального опыта обучающимися. В своей

статье [1] она показывает преимущества использования при обучении подростков методов «мозговой штурм», «дискуссия в малых группах», «микрофон» как способов формирования понимания возможностей информатики для решения практико-ориентированных задач.

С. И. Проценко, Е. А. Янкова главными преимуществами в использовании интерактивных методов обучения информатике видят в возможности эффективного, визуального представления учебного материала, создании соревновательных ситуаций, командном взаимодействии участников образовательного процесса. В статье [13] они предлагают методические рекомендации по внедрению в обучение на уроках информатики такого интерактивного метода обучения, как квиз.

Обучению конкретной теме базового курса информатики основного среднего образования посвящена статья Э. Р. Махмудовой [8]. В ее статье выделяется важность проведения уроков с использованием методов интерактивного обучения. При изучении темы X класса «Элементы теории множеств и алгебры логики» с учетом возраста обучающихся и специфики содержания темы оптимальными методами, по мнению автора, являются проблемное обучение и мозговой штурм.

Д. С. Украинская, Л. Г. Новикова отмечают, что информатика — это учебный предмет, содержание которого постоянно обновляется и модернизируется, поэтому целесообразно в процессе обучения использовать такие методы, которые способствуют развитию познавательной активности, самостоятельности в организации самообучения и усвоению постоянно возобновляющейся информации, например, ролевые игры или совместное решение проблем [17].

Таким образом, можно заключить, что применение интерактивных методов обучения на уроках информатики дает возможность обучающимся стать полноправными участниками образовательного процесса. Учитель уже не преподносит готовые знания, а побуждает учеников к их самостоятельному поиску, создает условия для активной познавательной деятельности учащихся.

Особенностями учебного предмета «Информатика» являются его практическая направленность, наличие большого количества межпредметных связей с другими учебными дисциплинами, постоянно возобновляющееся содержание обучения. Учесть все особенности информатики, максимально продуктивно использовать учебное время, создать условия для развития познавательной активности, интереса к учебному предмету у обучающихся возможно при использовании интерактивных методов обучения.

3. Возможности интерактивных методов обучения при изучении темы «Элементы алгебры логики» для развития логического мышления школьников

Логическое мышление — это процесс последовательного рассуждения для получения нового знания, вывода или заключения. Другими словами, под логическим мышлением следует понимать выстраивание структуры между фактами, цепочки рассуждений. М. П. Голованова и Е. А. Лялина понимают логическое мышление как

способность мыслить и делать выводы [3]. М. Ю. Олешков и В. М. Уваров в своем словаре «Современный образовательный процесс: основные понятия и термины» дают следующее определение логическому мышлению: «Логическое мышление — это вид мышления, сущность которого в оперировании понятиями, суждениями и умозаключениями с использованием законов логики» [9].

Об уровне формирования логического мышления можно судить по выполнению подростками таких операций, как анализ, синтез, сравнение, обобщение, конкретизация, абстрагирование, установление причинно-следственных связей; умениям делать умозаключения и аргументировать свою точку зрения, строить гипотезы и выводить из них следствия [15, 16]. Все они представляют умственные действия с понятиями и фактами, в результате реализации которых получаются новые знания. Для полноценного логического мышления необходимо гармоничное развитие всех операций, так как они тесно взаимосвязаны и взаимодополняют друг друга.

При изучении темы «Элементы алгебры логики» могут применяться следующие интерактивные методы обучения:

- творческое задание;
- работа в малых группах;
- дискуссия;
- круглый стол;
- ролевая игра;
- деловая игра;
- «мозговой штурм»;
- «дерево решений»;
- «микрофон»;
- лекция с заранее запланированными ошибками;
- ПОПС-формула;
- «броуновское движение»;
- «аквариум»;
- «карусель».

При применении в образовательном процессе каждого из перечисленных интерактивных методов обучения развиваются отдельные составляющие логического мышления.

Проанализировав Федеральную рабочую программу основного общего образования по предмету «Информатика» [18], отметим, что изучение логики начинается в VIII классе, продолжение изучения происходит только в X классе. Всего на изучение темы «Элементы алгебры логики» на базовом уровне в VIII классе отводится 5 часов и 1 час на контрольную работу по данной теме.

В соответствии с приказом Министерства просвещения Российской Федерации от 21 сентября 2022 года № 858 [12], рекомендованным для VIII класса является учебник информатики Л. Л. Босовой, А. Ю. Босовой [2]. Учебник соответствует ФГОС 2021 года и входит в состав УМК по информатике для VII—IX классов.

Согласно этому учебнику, в состав параграфа «Элементы алгебры логики» входят следующие темы:

- § 1.4. «Элементы алгебры логики».
- 1.4.1. Высказывания.
- 1.4.2. Логические операции И, ИЛИ, НЕ.
- 1.4.3. Построение таблиц истинности для логических выражений.
- 1.4.4. Свойства логических операций.
- 1.4.5. Решение логических задач.
- 1.4.6. Логические элементы.

Рассмотрим в качестве примера методику проведения урока «Логические операции И, ИЛИ, НЕ» с применением интерактивных методов обучения. В рамках данного урока можно использовать такие интерактивные методы обучения, как «броуновское движение», творческое задание, работа в малых группах, «карусель».

- «Броуновское движение» — это прием интерактивного обучения, который предполагает свободное передвижение обучающихся по классу с целью сбора заданной информации по определенной теме (поиск отсутствующих данных, ответов на поставленные вопросы, идей и способов решения определенной задачи и т. д.). В процессе ребята могут общаться друг с другом, а также обращаться к учителю за помощью.
- Творческое задание представляет собой задачу, решение которой требует от обучающихся использования фантазии.
- Работа в малых группах предполагает разделение класса на небольшие группы по три—пять человек, в которых ученики взаимодействуют и сотрудничают друг с другом с целью решения общих учебных задач.
- «Карусель» — это интерактивный прием обучения, когда учащиеся выстраиваются в два кольца: внутреннее и внешнее. Класс делится на группы. Во внутреннем кольце располагаются учащиеся, стоящие лицом к внешнему кольцу, а во внешнем кольце стоят ученики, которые перемещаются по часовой стрелке через определенные промежутки времени. В течение этого заданного ограниченного промежутка времени каждый раз новая пара должна успеть выполнить поставленное задание.

Покажем, как перечисленные методы могут быть использованы при изучении темы «Элементы алгебры логики».

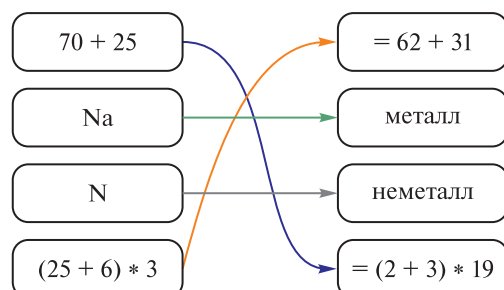
Урок на тему

«Логические операции НЕ, И, ИЛИ»

Цель урока: сформировать представления обучающихся о простейших логических связках.

На этапе актуализации знаний можно использовать интерактивный метод «броуновское движение». Обучающимся выдается карточка, на которой написана половина одного высказывания. Каждому ученику необходимо найти вторую половину высказывания, чтобы получилось истинное высказывание, при этом можно свободно перемещаться по классу. В результате через одну-две минуты ребята должны встать в нужные пары, а учитель должен проверить правильность выполнения задания.

Примеры карточек:



После объяснения теоретического материала в качестве творческого задания можно предложить каждому ученику создать (написать) историю, ситуацию, короткий рассказ или комикс, в которых он будет использовать логические операции НЕ, И, ИЛИ.

Пример рассказа приведен ниже.

Я сидел за компьютером И пытался разобраться в сложной задаче по информатике. Информации в интернете оказалось слишком много, я НЕ понимал, с чего начать решение. Я позвонил другу: ИЛИ он подскажет мне решение, ИЛИ я приду к нему И мы вместе разберемся. Пока я ждал ответа от товарища, нашел интересную статью, в которой было предложено решение моей задачи. Я понял, что мои предыдущие попытки были далеко НЕ правильными. Я разобрался в задаче И предложил другу погулять.

После выполнения задания обучающиеся делятся своей творческой работой с классом.

После выполнения учащимися творческого задания можно провести работу в группах. Для этого ученики делятся на группы по три-четыре человека и каждой группе выдаются карточки с предложениями, в которые необходимо вставить одну из возможных логических связок.

Примеры предложений из карточек:

- Пожарные используют для тушения пожара воду ____ песок.
- Математические задачи можно решить точно ____ приближенно.
- Чтобы достичь успеха, нужно быть трудолюбивым ____ умным.

Группы должны обсудить каждое предложение и дать свое объяснение, почему выбранная логическая связка является правильной. После обсуждения в группах обучающиеся делятся своими ответами с остальными группами.

На этапе закрепления полученных знаний можно использовать интерактивный метод «карусель». Учащиеся делятся на две группы, каждая из групп становится в круг, и группы образуют внешнее и внутреннее кольца. Ученики из внешнего кольца задают вопросы, а из внутреннего — отвечают на них. Затем по хлопку учителя обучающиеся из внешнего кольца сдвигаются на одного по часовой стрелке. После прохождения всего круга учащиеся, стоящие во внутреннем кольце, меняются местами с учениками из внешнего кольца.

Примеры вопросов по теме «Логические операции НЕ, И, ИЛИ»:

- Когда составное высказывание со связкой НЕ является истинным?
- Когда составное высказывание со связкой НЕ является ложным?
- Когда составное высказывание со связкой И является истинным?
- Когда составное высказывание со связкой ИЛИ является ложным?

В качестве домашнего задания можно дать такое: разбить на простые высказывания свое сочинение, составленное на уроке.

Описанные интерактивные методы могут быть применены и на других уроках по теме «Основы алгебры логики», так же, как и другие методы, использованные в описанном уроке.

4. Выводы

На изучение темы «Элементы алгебры логики» в содержании обучения информатике в основной школе на базовом уровне отводится всего шесть часов. Вместе с тем навыки, которые должны быть сформированы в результате изучения этой темы, очень востребованы для изучения других тем курса информатики, а также других учебных предметов. Поэтому важно организовать учебный процесс таким образом, чтобы он был продуктивным, обучающиеся занимали максимально активную позицию и были заинтересованы в изучении преподаваемого материала.

Использование интерактивных методов обучения при изучении темы «Элементы алгебры логики» позволяет сделать процесс обучения более интересным и привлекательным для обучающихся, помогает создать благоприятную обстановку в классе, способствует активной позиции каждого школьника, повышает мотивацию к обучению. Интерактивные методы обучения позволяют создать условия для формирования базовых составляющих логического мышления, таких как анализ, синтез, сравнение, обобщение информации, установление причинно-следственных связей, умения строить гипотезы и выводить из них следствия, умения делать умозаключения и аргументировать свою точку зрения.

Список источников

1. Белошапкина Е. В. Интерактивные методы обучения как средство социализации учащихся на уроках информатики в условиях реализации ФГОС ООО // Школа как платформа для успешной социализации обучающихся на уровне профессионального образования. Материалы V региональной научно-практической (очно-заочной) конференции (Волгоград — Котово, 16 февраля 2018 года). Волгоград, Котово: Волгоградский государственный технический университет, 2018. С. 160–162. EDN: UZPBGY.
2. Босова Л. Л., Босова А. Ю. Информатика. Базовый уровень. 8 класс. М.: Просвещение, 2023. 272 с.
3. Голованова М. П., Лялина Е. А. Развитие логического мышления с помощью игр // Научный альманах. 2016. № 4-2 (18). С. 100–102. EDN: TUYUNY. DOI: 10.17117/na.2016.04.02.100.
4. Густякина В. П. Возможности использования интерактивных методов обучения на уроках технологии // Информационно-коммуникационные технологии в педагогическом образовании. 2023. № 1 (82). С. 99–101. EDN: PMWVTK.
5. Ижденева И. В. Характеристика базовых средств мобильного обучения информатике // Информатика в школе. 2021. № 2 (165). С. 9–13. EDN: MKSKJX. DOI: 10.32517/2221-1993-2021-20-2-9-13.
6. Коротаева Е. В. Интерактивность современного обучения: как явление и как понятие // Педагогическое образование в России. 2022. № 4. С. 8–15. EDN: TOPFWN.
7. Купрюхина К. В., Кравец О. В. Использование интерактивных методов обучения на уроках английского языка // Вестник Таганрогского государственного педагогического института. 2023. № 1. С. 351–357. EDN: FPDHCU.
8. Махмудова Э. Р. Методика применения интерактивных технологий обучения в базовом курсе информатики // Инжиниринг предприятий и управление знаниями (ИП&УЗ-2022). Сборник научных трудов XXV Российской научной конференции. (Москва, 06–07 декабря 2022 года). В 2 т. Т. 2. М.: Российский экономический университет имени Г. В. Плеханова, 2022. С. 133–141. EDN: NXQDJQ.
9. Олешков М. Ю., Уваров В. М. Современный образовательный процесс: основные понятия и термины. М.: Компания Спутник+, 2006. 191 с.
10. Панина Т. С., Вавилова Л. Н. Современные способы активизации обучения: учебное пособие для студентов высших учебных заведений, обучающихся по педагогическим специальностям. М.: Академия, 2008. 176 с. EDN: PEMRWS.
11. Плакшина И. В. Интерактивные технологии в обучении и воспитании: методическое пособие. Владимир: Изд-во ВлГУ, 2014. 163 с.
12. Приказ Министерства просвещения РФ от 21 сентября 2022 г. № 858 «Об утверждении федерального перечня учебников, допущенных к использованию при реализации имеющих государственную аккредитацию образовательных программ начального общего, основного общего, среднего общего образования организациями, осуществляющими образовательную деятельность и установления предельного срока использования исключенных учебников» (с изменениями и дополнениями). <https://base.garant.ru/405590287/>
13. Проценко С. И., Янкова Е. А. Особенности использования интерактивных технологий обучения на уроках информатики // Вестник Башкирского государственного педагогического университета им. М. Акмуллы. 2023. Т. 3. № S1 (68). С. 87–90. EDN: THUHCG.
14. Пузиновская С. Г., Счеснович О. А. Парная работа на уроках информатики как одна из форм организации учебного сотрудничества учащихся // Информатика в школе. 2021. № 7 (170). С. 40–47. EDN: RPNSBE. DOI: 10.32517/2221-1993-2021-20-7-40-47.
15. Совертков П. И., Нурулин А. С. Развитие и оценка логического мышления // Компьютерные инструменты в образовании. 2003. № 6. С. 71–76. <https://cyberleninka.ru/article/n/razvitiye-i-otsenka-logicheskogo-myshleniya>
16. Угланова И. Л., Погужина И. Н. Что может предложить новая методология оценки мышления школьников современному образованию // Вопросы образования. 2021. № 4. С. 8–34. EDN: QKMAXK. DOI: 10.17323/1814-9545-2021-4-8-34.
17. Украинская Д. С., Новикова Л. Г. Использование активных и интерактивных технологий в обучении информатике // Теория и практика современной науки. 2019. № 2 (44). С. 347–350. EDN: AWADQW.
18. Федеральная рабочая программа основного общего образования. Информатика (базовый уровень) (для 7–9 классов образовательных организаций) // Единое содержание общего образования. https://edsoo.ru/wp-content/uploads/2023/08/15_ФРП-Информатика-7-9-классы_база.pdf
19. Федеральный государственный образовательный стандарт основного общего образования (утв. приказом Министерства просвещения Российской Федерации от 31 мая 2021 г. № 287). https://base.garant.ru/401433920/53f89421bbdaf741eb2d1ecc4ddb4c33/#block_1000
20. Храмова Н. А., Храмов Д. А. Использование интерактивных методов обучения на уроках математики // Учебный эксперимент в образовании. 2021. № 3 (99). С. 70–78. EDN: TWLLNH. DOI: 10.51609/2079-875X_2021_03_70.



К. Ю. Поляков

*Санкт-Петербургский государственный морской технический университет, Санкт-Петербург, Россия;
Школа № 174, Санкт-Петербург, Россия*

КЛАСТЕРИЗАЦИЯ В ЗАДАЧАХ ЕГЭ ПО ИНФОРМАТИКЕ

Аннотация

В статье рассмотрены методические подходы к решению задачи на анализ данных, включенной в единый государственный экзамен (ЕГЭ) по информатике с 2025 года. Показано, что при любой форме кластеров возможно разделить их набором прямых, параллельных осям координат, или наклонными прямыми. Предлагается ускоренный алгоритм поиска центров кластеров, позволяющий значительно уменьшить время вычислений и таким образом решить задачу для наборов данных, содержащих более 10 000 точек, за приемлемое время. Рассмотрены усложненные, в сравнении с демонстрационным вариантом контрольно-измерительных материалов, варианты задачи: наличие аномалий, которые нужно игнорировать при поиске центра кластера, и кластеры сложной формы. Проведено сравнение различных подходов к разделению данных на кластеры, включая алгоритм DBSCAN и метод k -средних.

Ключевые слова: информатика, ЕГЭ, КИМ, анализ данных, кластеризация, аномалии, алгоритм DBSCAN, метод k -средних.

Информация об авторе

Поляков Константин Юрьевич, доктор тех. наук, профессор кафедры судовой автоматики и измерений, факультет корабельной энергетики и автоматики, Санкт-Петербургский государственный морской технический университет, Санкт-Петербург, Россия; учитель информатики, школа № 174, Санкт-Петербург, Россия; *e-mail*: kpolyakov@mail.ru

1. Введение

В демонстрационный вариант контрольно-измерительных материалов (КИМ) ЕГЭ по информатике 2025 года впервые включена задача на анализ данных [3]. В этой задаче требуется выполнить кластеризацию данных, заданных координатами на плоскости, и найти центры кластеров. В настоящей статье рассматриваются возможные подходы к решению этой задачи в условиях ЕГЭ и формулируются рекомендации, которые могут быть полезны при подготовке школьников к экзамену. Исследуется не только простейший вариант задачи, приведенный в [3], но и ее возможные вариации: наличие аномалий и сложные формы кластеров.

Задачи кластеризации являются одними из наиболее популярных задач анализа данных — см., например, [1, 8–10]. Для практических вычислений можно использовать как языки программирования [9], так и табличные процессоры [6].

Особенность ситуации на ЕГЭ состоит в том, что специализированные пакеты для решения задач анализа данных и машинного обучения, такие как `scikit-learn`*, недоступны. Поэтому можно использовать только возможности стандартной библиотеки языка программирования.

* Пакет `scikit-learn` можно загрузить с сайта: <https://scikit-learn.org>

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_01-2025/

K. Yu. Polyakov

State Marine Technical University, Saint Petersburg, Russia;
School 174, Saint Petersburg, Russia

CLUSTERING IN TASKS OF THE UNIFIED STATE EXAM IN INFORMATICS

Abstract

The article presents methodological approaches to solving the data analysis problem included in the Unified State Exam (USE) in informatics since 2025. It is shown that for any shape of clusters, it is possible to separate them by a set of straight lines parallel to the coordinate axes or by inclined lines. An accelerated algorithm for finding cluster centers is proposed, which allows to significantly reduce the computation time and thus to solve the problem for data sets containing more than 10,000 points in an acceptable time. Complex versions of the problem, as compared with the demonstration version of the control and measuring materials, are considered: the presence of anomalies that should be ignored when searching for the cluster center, and clusters of complex shape. A comparison of various approaches to dividing data into clusters, including the DBSCAN algorithm and the k -means method, is carried out.

Keywords: informatics, USE, test materials, data analysis, clustering, anomaly, DBSCAN algorithm, k -means method.

Information about the author

Konstantin Yu. Polyakov, Doctor of Sciences (Engineering), Professor at the Department of Ship Automation and Measurements, Faculty of Ship Power Engineering and Automation, State Marine Technical University, Saint Petersburg, Russia; informatics teacher, School 174, Saint Petersburg, Russia; *e-mail*: kpolyakov@mail.ru

Далее все фрагменты программ приведены на языке Python, хотя не составит труда переписать их на любом языке, который поддерживается на ЕГЭ.

2. Формулировка задачи

Рассмотрим сокращенную формулировку задачи из демонстрационного варианта КИМ 2025 года.

Задача 1 [3].

Фрагмент звездного неба спроецирован на плоскость с декартовой системой координат. Ученый решил провести кластеризацию полученных точек, являющихся изображениями звезд, т. е. разбить их множество на N непересекающихся непустых подмножеств (кластеров). Будем называть **центром кластера** точку этого кластера, сумма расстояний от которой до всех остальных точек кластера минимальна. Для каждого кластера гарантируется единственность его центра. Расстояние между двумя точками на плоскости $A(x_1, y_1)$ и $B(x_2, y_2)$ вычисляется по формуле:

$$d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

В файле *input.txt* хранятся координаты точек трех кластеров. Известно, что количество точек не превышает 10 000. Определите координаты центра каждого кластера, затем вычислите два числа: P_x — среднее арифметическое абсцисс центров кластеров, и P_y — среднее арифметическое ординат центров кластеров. В ответе запишите два числа: сначала целую часть произведения $P_x \times 10\,000$, затем целую часть произведения $P_y \times 10\,000$.

Возможные данные проиллюстрированы графиком на рисунке 1.

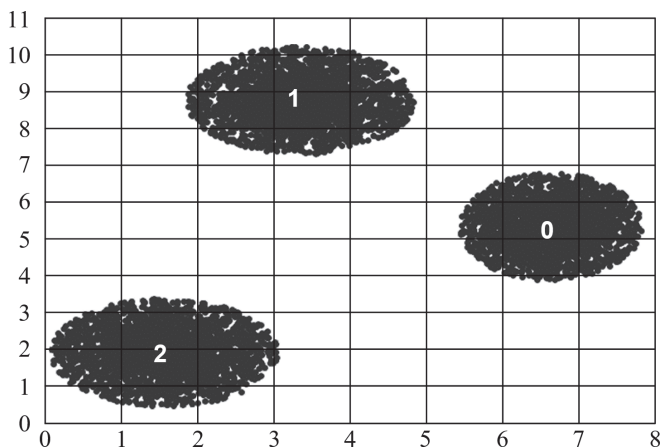


Рис. 1. График к задаче 1

Решение задачи включает два этапа:

- 1) разбиение данных на кластеры;
 - 2) вычисление координат центров кластеров.
- Разберем их последовательно.

3. Разбиение данных на кластеры

Сначала необходимо выполнить *визуализацию*, т. е. представить данные в виде, удобном для наблюдения. По условию задачи данные — это декартовы координаты точек на плоскости, поэтому можно просто отметить

эти точки на графике каким-нибудь маркером, не соединяя их, как это сделано на рисунке 1. Для этой цели удобно использовать табличный процессор, например, LibreOffice Calc или Microsoft Excel.

Кластеры на рисунке 1 пронумерованы, начиная с нуля. В данном случае их легко отделить друг от друга прямыми, параллельными осям координат. По диаграмме видно, что все точки, для которых $x > 5$, относятся к кластеру 0. Оставшиеся точки разделяются на два кластера прямой $y = 6$: при $y > 6$ точка относится к кластеру 1, а в противном случае — к кластеру 2. Следующая функция принимает координаты точки и возвращает номер кластера, к которому она относится:

```
def clusterNo( x, y ):
    return 0 if x > 5 else \
           1 if y > 6 else \
           2
```

Удобно сразу распределить точки по кластерам при чтении данных. Для этого создаем список *clusters*:

```
K = 3          # количество кластеров
clusters = [ [] for i in range(K) ]
```

В случае трех кластеров (при $K = 3$) список *clusters* сначала содержит три пустых списка. В каждый из них мы будем добавлять координаты точек соответствующего кластера. Цикл, в котором данные читаются из файла и точки распределяются по кластерам, можно записать так:

```
for s in open("input.txt"):
    x, y = s.replace(',', '.').split() # 1
    x, y = float(x), float(y)          # 2
    k = clusterNo( x, y )               # 3
    clusters[k].append( (x, y) )       # 4
```

Цикл *for* перебирает все строки файла *input.txt*, так что каждая из них на очередной итерации цикла оказывается в переменной *s*. В исходном файле целые части чисел отделяются от дробных запятой, а не точкой, как в языках программирования. Поэтому в строке 1 (см. нумерацию в комментариях) все запятые заменяются на точки с помощью метода *replace*. Строка сразу разбивается методом *split* на две части: в первой записана *x*-координата очередной точки, а во второй — ее *y*-координата. В строке 2 эти координаты с помощью функции *float* преобразуются из символьной записи в вещественные числа. Номер кластера *k*, к которому относится эта точка, определяем с помощью функции *clusterNo* (строка 3). Затем в строке 4 координаты новой точки, объединенные в кортеж, добавляются к списку *clusters[k]*, который содержит точки кластера *k*.

4. Вычисление координат центра кластера

Заметим, что по условию задачи центр кластера — это *одна из точек кластера*, координаты которой присутствуют в файле, а не произвольная точка плоскости. Для определения центра можно перебрать все точки, принадлежащие кластеру, вычисляя для каждой из них сумму расстояний до остальных точек этого же кластера.

В качестве центра выбирается точка, для которой эта сумма расстояний минимальна.

Если в кластере N точек, то нам нужно для каждой из них вычислить $N - 1$ расстояний до других точек. Поэтому сложность алгоритма — квадратичная, $O(N^2)$. Для 10 000 точек время работы программы должно составить несколько секунд, поэтому такой алгоритм вполне можно использовать на экзамене.

Для определения расстояния между двумя точками удобно написать отдельную функцию:

```
def dist( p1, p2 ):
    return ((p1[0]-p2[0])**2 +
            (p1[1]-p2[1])**2)**0.5
```

Параметры функции $p1$ и $p2$ — это пары координат (кортежи), которые описывают две точки. Первые элементы пар, $p1[0]$ и $p2[0]$, — это x -координаты точек, а вторые элементы, $p1[1]$ и $p2[1]$, — это их y -координаты. Функция возвращает квадратный корень из суммы квадратов разностей координат точек по каждой оси, т. е. евклидово расстояние между ними.

Для ускорения вычислений можно вместо нашей функции *dist* использовать одноименную функцию из модуля *math*. Благодаря эффективной реализации функции из стандартной библиотеки время поиска центров трех кластеров для данных из [3] сокращается в два-три раза.

Удобно оформить алгоритм определения центра кластера как отдельную функцию *getCenter*, которая принимает один параметр — список точек кластера:

```
def getCenter( cluster ):
    minSumDist = float("inf")
    for p0 in cluster:
        sumDist = sum( dist(p0,p)
                       for p in cluster )
        if sumDist < minSumDist:
            minSumDist = sumDist
            center = p0
    return center
```

Сумму расстояний от очередной точки $p0$ до остальных точек кластера вычисляем с помощью функции *sum* и затем сравниваем эту сумму с минимальной, которая записана в переменной *minSumDist*. Если найдена новая минимальная сумма, запоминаем точку $p0$ как новый центр кластера в переменной *center*.

После распределения данных по кластерам координаты всех точек кластера k находятся в списке *clusters[k]*. Используя функцию *getCenter*, центры всех кластеров можно найти с помощью генератора списка:

```
centers = [ getCenter(cluster)
            for cluster in clusters ]
```

Теперь *centers[k]* — это кортеж, содержащий координаты центра кластера с номером k . Первый элемент кортежа, *centers[k][0]*, — это x -координата центра, а *centers[k][1]* — его y -координата. Остается вычислить среднее арифметическое x -координат и y -координат центров кластеров и вывести результат:

```
Px = sum( centers[k][0] for k in range(K) ) / K
Py = sum( centers[k][1] for k in range(K) ) / K
print( int(Px*10000), int(Py*10000) )
```

5. Ускорение вычислений

Попробуем ускорить вычисление координат центра кластера. Очевидно, что центр находится где-то в середине кластера, а не у его края. Определим координаты **средней точки (центроида)** кластера как средние арифметические координат по каждой оси:

```
N = len(cluster)
cx = sum( p[0] for p in cluster ) / N
cy = sum( p[1] for p in cluster ) / N
pMid = (cx, cy)
```

Теперь в переменной *cx* записана x -координата средней точки, а в переменной *cy* — ее y -координата; эти координаты объединены в кортеж *pMid*.

Естественно предположить, что центр кластера, который мы ищем, расположен вблизи средней точки. Чтобы сократить перебор, будем рассматривать только те точки, которые находятся на расстоянии не более некоторого δ от средней точки *pMid*. Из этих точек составим список *pointsNearCenter*:

```
delta = 0.2
pointsNearCenter = [p for p in cluster
                     if dist(pMid,p) < delta ]
```

и будем выбирать центр только среди точек из этого списка. Их будет намного меньше, чем всех точек кластера, так что алгоритм закончит работу быстрее.

Приведем ускоренный вариант функции *getCenter*:

```
def getCenter( cluster ):
    N = len(cluster)
    pMid = ( sum( p[0] for p in cluster ) / N,
             sum( p[1] for p in cluster ) / N )
    delta = 0.2
    pointsNearCenter = [p for p in cluster
                        if dist(pMid,p) < delta ]
    minSumDist = float("inf")
    for p0 in pointsNearCenter:
        sumDist = sum( dist(p0,p)
                       for p in cluster )
        if sumDist < minSumDist:
            minSumDist = sumDist
            center = p0
    return center
```

Для набора данных из [3] при $\delta = 0,2$ улучшенная версия алгоритма работает за счет сокращения перебора примерно в 6 раз быстрее, чем первоначальная.

Значение δ выбирается с запасом в зависимости от размеров кластера и плотности точек в нем. Как правило, можно выбирать δ равным 10–15 % от радиуса кластера; для кластеров сложной формы иногда приходится использовать большие значения δ . Если выбрано слишком малое δ , то точка, которая является центром кластера, может не попасть в δ -окрестность средней точки, и тогда получится неверный ответ. Чем больше δ , тем медленнее работает алгоритм, но меньше вероятность ошибки. Радиусы кластеров для данных из [3] примерно равны 1,5 единицы, при этом правильное решение получается для любого $\delta \geq 0,02$.

6. Аномалии

Аномалии — это значения в наборе данных, которые сильно отличаются от всех остальных. В задачах кластеризации, которые могут быть предложены на ЕГЭ, аномалии — это точки, не принадлежащие ни одному из кластеров. Перед обработкой данных аномалии нужно удалить — это часть процедуры, которая называется *очисткой данных*. Для этого необходимо точно определить, что в данной задаче мы считаем аномалиями.

Задача 2.

[Текст, совпадающий с условием задачи 1].

Аномалиями назовем группы не более чем из 10 точек, находящиеся на расстоянии более одной условной единицы от точек всех кластеров. При расчетах аномалии учитывать не нужно.

Возможные данные проиллюстрированы графиком на рисунке 2. Аномалии выделены штриховыми линиями.

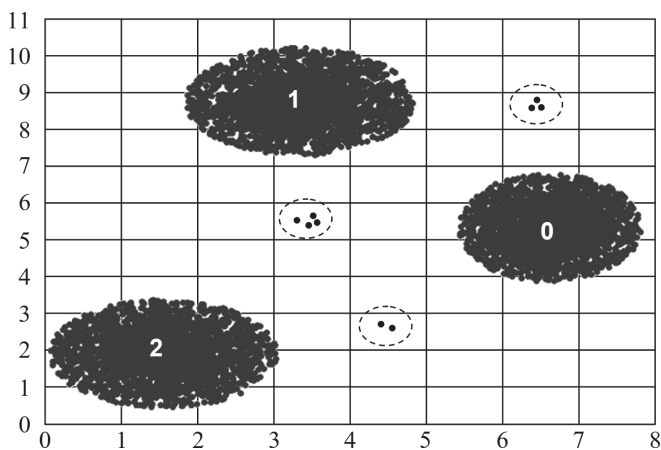


Рис. 2. Иллюстрация данных задачи 2

В этой задаче нам не удастся так просто разделить точки на кластеры, как мы это сделали при решении задачи 1. Дело в том, что условию $x > 5$, например, соответствуют не только точки кластера 0, но и точки аномалии в полосе $6 < x < 7$. Поэтому для выделения кластера 0 необходимо применить сложное условие $x > 5$ and $y < 7$. Точки кластера 1 определяются условием $x < 5$ and $y > 6$, а точки кластера 2 — условием $x < 4$ and $y < 4$. Каждое из трех условий задает **квадрант** — четверть плоскости, ограниченную прямым углом, причем ни в одну из этих областей аномалии не попадают.

Таким образом, с помощью условий, которые выделяют только интересующие нас кластеры, мы исключили все аномалии, так что дальнейший алгоритм решения задачи изменять не нужно.

Изменим функцию *clusterNo* так, чтобы она возвращала значение -1 для точек аномалий, не принадлежащих ни одному из кластеров:

```
def clusterNo( x, y ):
    return 0 if x > 5 and y < 7 else \
           1 if x < 5 and y > 6 else \
           2 if x < 4 and y < 4 else \
           -1
```

Аномалии не должны добавляться в список *clusters*, поэтому нужна дополнительная проверка в цикле ввода данных (эта строка программы выделена фоном):

```
for s in open("input.txt"):
    x, y = s.replace(',', '.').split()
    x, y = float(x), float(y)
    k = clusterNo( x, y )
    if k >= 0:
        clusters[k].append( (x, y) )
```

Заметим, что у нас получилась универсальная программа: при изменении количества и расположения кластеров и аномалий требуется изменить только значение переменной k в начале программы и функцию *clusterNo*, которая определяет номер кластера по координатам точки.

7. Наклонные разделительные прямые

Рассмотрим три кластера на рисунке 3. Ограничить каждый кластер квадрантом здесь не получится, однако можно отделить кластеры друг от друга наклонными прямыми I и II, которые показаны на рисунке штриховыми линиями.

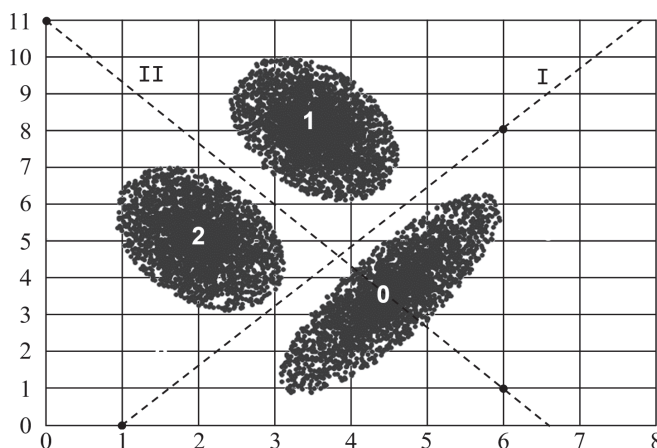


Рис. 3. Отделение кластеров друг от друга наклонными прямыми

Уравнения этих прямых можно построить аналитически, используя сведения из курса школьной математики. В данном случае получаем следующие уравнения:

$$\text{прямая I: } y = \frac{8}{5}(x - 1);$$

$$\text{прямая II: } y = -\frac{5}{3}x + 11.$$

Для проверки можно построить эти прямые на диаграмме, добавив к данным из файла дополнительные столбцы и расширив на них область данных диаграммы. Например, на рисунке 4 на диаграмму добавлена прямая I.

В ячейку C1 (рис. 4) записана формула: $=\$D\$2*A1+\$E\2 , которая потом была скопирована на весь столбец C. Изменяя значения ячеек D2 и E2, можно подбирать коэффициенты k и b экспериментально.

Используя полученные выше уравнения разделяющих прямых, запишем функцию *clusterNo* для рассмотренного случая:

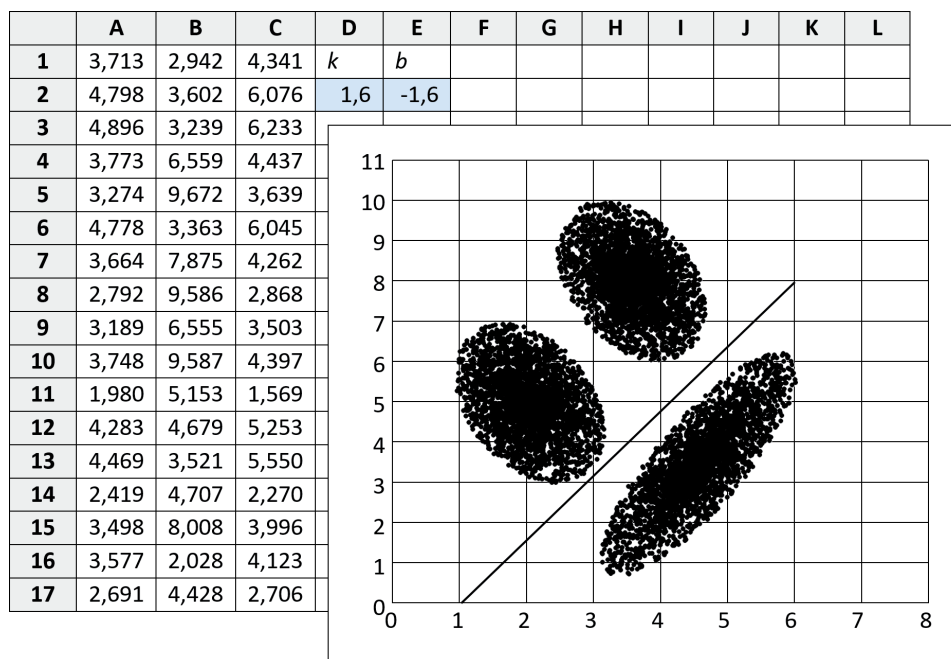


Рис. 4. Добавление наклонной прямой на диаграмму

```
def clusterNo( x, y ):
    return 0 if y < (8/5)*(x-1) else \
        1 if y > -5/3*x+11 else \
        2
```

Если в данных есть аномалии, их можно учесть так же, как мы делали ранее. Отметим, что остальная часть программы не изменится.

8. Метод покрытия прямоугольниками

Рассмотрим диаграмму на рисунке 5.

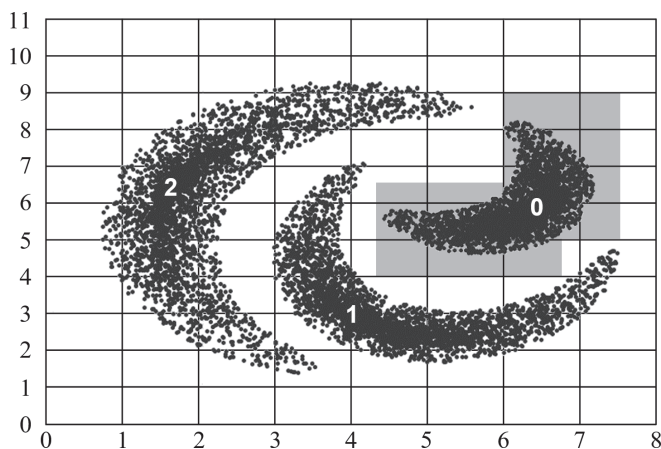


Рис. 5. Пример усложненной диаграммы. Покрытие прямоугольниками кластера 0

Чтобы разделить кластеры в этой задаче, можно провести между ними какие-нибудь сложные границы или собирать кластеры из частей.

Покажем, как можно записать условие, определяющее кластер 0 на рисунке 5. Заметим, что этот кластер полностью перекрывается двумя прямоугольниками, которые выделены фоном на рисунке 5. Поэтому условие

принадлежности точки с координатами (x, y) кластеру 0 можно записать в виде:

$$(4,3 < x < 6,8 \text{ and } 4 < y < 6,5) \text{ or } (6 < x < 7,5 \text{ and } 5 < y < 9).$$

В данном случае правую и верхнюю границы второго прямоугольника можно вообще не указывать, потому что правее и выше этого прямоугольника никаких «лишних» точек нет:

$$(4,3 < x < 6,8 \text{ and } 4 < y < 6,5) \text{ or } (x > 6 \text{ and } y > 5).$$

На рисунке 6 показано покрытие прямоугольниками кластера 1 (кластер 0 удален с диаграммы, поскольку мы его уже обработали).

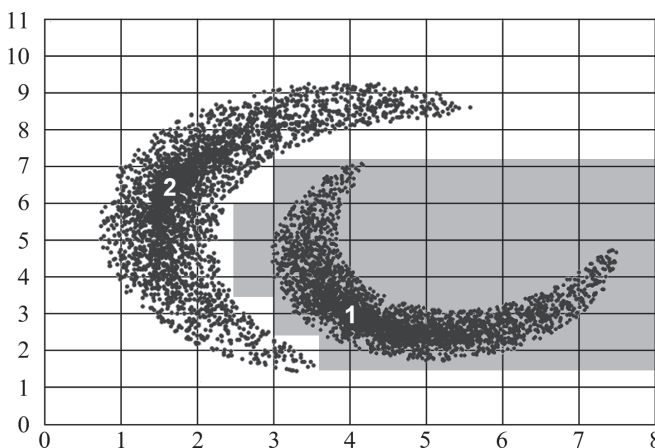


Рис. 6. Покрытие прямоугольниками кластера 1

Чтобы выделить кластер 1, нужно использовать три полосы:

$$(x > 3 \text{ and } 2,4 < y < 7,2)$$

$$\text{or } (x > 2,5 \text{ and } 3,5 < y < 6)$$

$$\text{or } (x > 3,6 \text{ and } 1,5 < y < 3)$$

Аномалий в этой задаче нет, поэтому все точки, которые не попали в кластеры 0 и 1, относятся к кластеру 2. Функция *clusterNo* запишется следующим образом:

```
def clusterNo( x, y ):
    return 0 if \
        (4.3 < x < 6.8 and 4 < y < 6.5) \
        or (x > 6 and y > 5) else \
    1 if (x > 3 and 2.4 < y < 7.2) or \
        (x > 2.5 and 3.5 < y < 6) or \
        (x > 3.6 and 1.5 < y < 3) else \
    2
```

Этот простой метод применим для кластеров любой формы. Поэтому в задачах ЕГЭ нет необходимости использовать разделительные линии, которые задаются сложными уравнениями.

9. Алгоритм DBSCAN

Во многих случаях возможно автоматическое разбиение точек на кластеры, выполняемое программой без участия человека. В случае, когда точки кластера расположены достаточно плотно, а именно это характерно для задач ЕГЭ, хорошо работает алгоритм DBSCAN (*англ.* Density-based spatial clustering of applications with noise) [8, 10]. Его основная идея состоит в том, что точки, расстояние между которыми меньше некоторого ε , считаются соседями, т. е. относятся к одному и тому же кластеру*.

Мы начинаем строить кластер, взяв произвольную точку из набора данных. Пусть это будет точка 0, обозначенная белым цветом на рисунке 7, а. Среди оставшихся точек выбираем и добавляем в тот же кластер всех ее соседей, т. е. все точки ее ε -окрестности, расстояние от которых до точки 0 меньше ε . Эти точки (с номерами 1, 2 и 3) обозначены на рисунке 7, а черным цветом, а все еще не выбранные точки — серым.

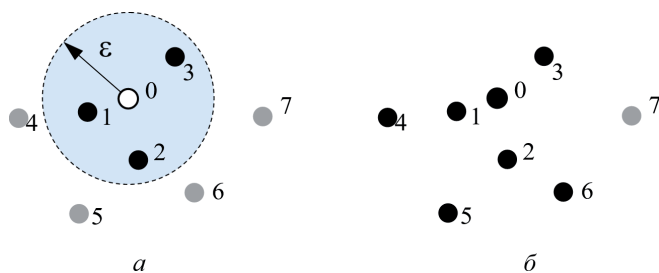


Рис. 7. Построение кластера

Затем в кластер добавляются все точки в ε -окрестностях вновь добавленных точек. В ε -окрестности точки 1 находим одного нового соседа — точку 4. В ε -окрестности точки 2 попадают два новых соседа: точки 5 и 6. Просмотр ε -окрестностей точек 3, 4, 5 и 6 не дает результатов: оставшаяся точка 7 находится от любой точки кластера на расстоянии, большем, чем ε . Окончательный результат показан на рисунке 7, б: в кластер включены точки 0, 1, 2, 3, 4, 5 и 6, а точка 7 этому кластеру не принадлежит.

Алгоритм DBSCAN позволяет автоматически разбить все точки на кластеры и выявить аномалии. Напри-

мер, если обнаружен кластер, в котором менее 10 точек, он может считаться аномалией. В отличие от многих алгоритмов кластеризации, количество кластеров задавать заранее не нужно.

Рассмотренный вариант алгоритма DBSCAN имеет единственный параметр — величину ε . Для каждого конкретного набора данных подходящее значение ε определяется визуально по диаграмме.

Напишем программу, которая реализует алгоритм DBSCAN. Координаты всех точек читаем из файла и записываем в список *data* как кортежи, состоящие из двух координат:

```
data = []
for s in open("input.txt"):
    x, y = s.replace(' ', ',').split()
    data.append( (float(x), float(y)) )
```

Основную идею алгоритма реализует функция *addNeighbors*, которая добавляет к кластеру *cluster* всех соседей заданной точки *p0*:

```
def addNeighbors( cluster, p0, eps ):
    neighbors = [ p for p in data
                  if dist(p, p0) < eps ]
    cluster += neighbors
    for p in neighbors:
        data.remove( p )
    for p in neighbors:
        addNeighbors( cluster, p, eps )
```

Функция принимает три параметра: *cluster* — список точек кластера, который мы хотим расширить; *p0* — кортеж, содержащий координаты точки, соседей которой мы добавляем; *eps* — значение параметра ε . Функция ничего не возвращает (в ней нет оператора *return*), в результате ее работы изменяется список *cluster*, переданный ей как параметр.

Сначала генератор списка находит все точки, находящиеся от точки *p0* на расстоянии, меньшем, чем ε , и строит из них список *neighbors*. Затем все выбранные точки удаляются из глобального списка *data*. Второй цикл в теле функции добавляет в кластер все точки, соседние с уже выбранными, т. е. «соседей соседей». Рекурсия закончится тогда, когда для очередной точки не удастся найти новых (еще не выбранных) соседей в ее ε -окрестности.

Теперь можно написать функцию *getCluster*, которая собирает в список все точки кластера, к которому принадлежит заданная точка *p0*:

```
def getCluster( p0, eps ):
    cluster = [ p0 ]
    data.remove( p0 )
    addNeighbors( cluster, p0, eps )
    return cluster
```

Функция создает новый кластер, содержащий сначала только одну точку *p0*, и удаляет эту точку из списка *data*. Затем в кластер добавляются все соседи начальной точки, соседи соседей и т. д. Результат работы функции — список точек кластера, содержащего точку *p0*.

Если *p0* — это изолированная точка, в ε -окрестности которой нет ни одного соседа, то функция вернет список, состоящий из одной этой точки. Если аномалия включа-

* Далее описывается упрощенный вариант алгоритма DBSCAN, применимый для решения задач ЕГЭ.

ет несколько точек рядом с точкой p_0 , только эта группа и будет выделена в новый кластер.

Разбиение на кластеры в основной программе выполняется с помощью цикла, который работает до тех пор, пока в списке *data* не останется невыбранных точек.

```
clusters = []
while data:
    cluster = getCluster( data[0], eps )
    if len(cluster) > 10:
        clusters.append( cluster )
```

Проверка условия, выделенная фоном, учитывает аномалии, состоящие не более чем из 10 точек. Аномалии не добавляются в список *clusters*.

Другие варианты реализации алгоритма DBSCAN при решении задач ЕГЭ, в том числе нерекурсивные, можно найти в [2, 4, 5, 7].

Для того чтобы убедиться, что алгоритм правильно разделил данные на кластеры, полезно построить картинку на экране так, чтобы точки одного кластера имели одинаковый цвет, а точки разных кластеров были раскрашены по-разному. Для этой цели в [2] было предложено использовать встроенный модуль «черепаший графики» *turtle*. Отметим, что из-за особенностей графики модуля *turtle* картинка для большого количества точек строится довольно долго, особенно на слабых компьютерах.

Несомненное достоинство метода DBSCAN состоит в том, что он позволяет разделить точки на кластеры полностью автоматически. При этом количество кластеров не нужно знать заранее, необходимо только выбрать подходящее значение параметра ϵ . Если значение ϵ слишком велико, программа может ошибочно объединить несколько кластеров в один. Это значит, что в данных оказались точки, принадлежащие разным кластерам, но отстоящие друг от друга на расстояние, меньшее, чем ϵ . Если задать слишком маленькое значение ϵ , программа может воспринять один кластер как несколько отдельных кластеров. Обычно правильное значение ϵ несложно найти визуально по построенной диаграмме.

Основной недостаток алгоритма DBSCAN связан с тем, что он не позволяет автоматически разделить кластеры, точки которых находятся близко друг к другу (например, когда кластеры треугольной формы почти касаются вершинами).

10. Метод k -средних

Метод k -средних (*англ.* *k-means*) позволяет не только разделить данные на кластеры в автоматическом режиме, но и сразу найти их центры. Для этого требуется заранее задать количество кластеров (в задачах ЕГЭ оно известно из условия).

В классическом варианте методом k -средних определяют **центроиды** кластеров, которые могут располагаться в произвольных точках плоскости. В задачах ЕГЭ центр кластера, который нужно найти, должен обязательно совпадать с одной из точек этого кластера. Такая точка называется **медоидом** кластера, а соответствующий вариант решения задачи — методом k -медоидов (*англ.* *k-medoids*). Следуя формулировке

задания 27 из [3], мы будем далее называть медоид центром кластера.

Пусть нужно разделить данные на k кластеров и для каждого кластера найти его центр. Эта задача строго решается только полным перебором вариантов. Однако есть несколько эвристических методов, которые позволяют быстро найти хорошее решение, но не гарантируют, что оно будет оптимальным. Один из алгоритмов выглядит так:

Шаг 1. Выбрать k точек — временные центры кластеров.

Шаг 2. Каждую точку данных связать с ближайшим центром; таким образом выполняется разделение на кластеры.

Шаг 3. Для каждого кластера найти новый центр — точку кластера, суммарное расстояние от которой до всех остальных точек этого кластера минимально.

Шаг 4. Если центры кластеров поменялись, перейти к шагу 2, иначе — завершить работу алгоритма.

К сожалению, этот алгоритм нельзя использовать «вслепую», без анализа конкретных данных. Он не всегда находит самое лучшее решение, результат его работы зависит от выбора начальных координат центров на шаге 1. Как их лучше выбирать — точно не известно. Обычно используют один из двух вариантов: случайный выбор или назначение вручную по результатам анализа диаграммы. Например, для данных, изображенных на рисунке 1, начальные центры кластеров 0, 1 и 2 можно выбрать в точках (6,5; 5), (3,5; 9) и (1,5; 2) соответственно. Неважно, что точки с такими координатами, вероятнее всего, не входят в набор данных. Это не влияет на окончательный результат работы алгоритма, так как уже на первой итерации после шага 3 центры будут размещены в точках, принадлежащих кластерам.

Так же, как и в программе для метода DBSCAN, загружаем данные из файла в список *data*. Затем задаем количество кластеров и примерные координаты их центров:

```
K = 3 # количество кластеров
centers = [(6.5, 5), (3.5, 9), (1.5, 2)]
```

Список *oldCenters*, где будем сохранять предыдущие координаты центров, оставляем пустым:

```
oldCenters = []
```

Основной цикл выполняется до тех пор, пока координаты центров кластеров не перестанут изменяться, т. е. пока списки *centers* и *oldCenters* не совпадут:

```
while centers != oldCenters:
    oldCenters = centers
    clusters = [ [] for i in range(K) ]
    for p in data:
        distToCenters = [ dist(p, center)
                           for center in centers ]
        k = distToCenters.index \
            ( min(distToCenters) )
        clusters[k].append( p )
    centers = [ getCenter(cluster)
               for cluster in clusters ]
```

Список *clusters* сначала содержит k пустых списков, в которые будут добавляться точки. Вложенный цикл *for*

перебирает все точки из списка *data*, определяя, к какому центру ближе всего каждая из них, т. е. к какому кластеру она относится. Для этого в списке *distToCenters* вычисляются расстояния от выбранной точки до всех центров. Затем определяется индекс минимального элемента в этом списке — это и будет номер нужного кластера *k*, в который добавляется эта точка. После разделения на кластеры центр каждого кластера пересчитывается с помощью функции *getCenter*.

Увы, алгоритм *k*-медоидов может приводить к неверному ответу. Рассмотрим данные, представленные на рисунке 8.

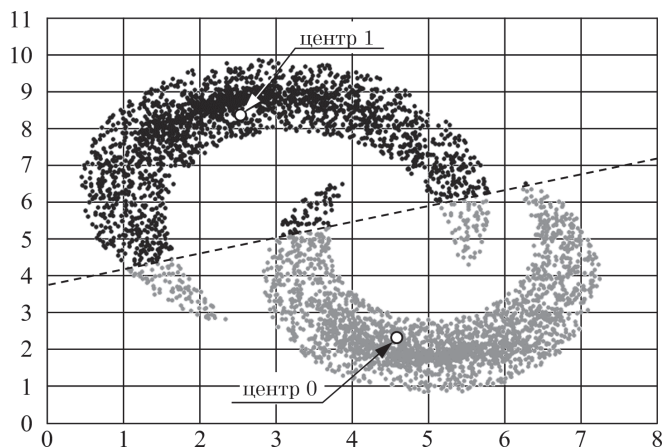


Рис. 8. Пример данных для иллюстрации неверного результата при использовании алгоритма *k*-медоидов

Приведенный выше алгоритм определяет центры кластеров как точки с координатами* (2,49; 8,29) и (4,93; 2,30). К сожалению, это ошибочный результат. Проблему легко увидеть, если построить диаграмму и раскрасить точки разных кластеров разными цветами. Штриховая линия на рисунке 8 показывает границу между кластерами — это точки, равноотстоящие от обоих центров, найденных с помощью нашего алгоритма. Понятно, что такое деление неверно: часть каждого кластера отнесена к другому кластеру. Границы между кластерами образуют **диаграмму Вороного** — разбиение плоскости, при котором точки в каждой области расположены ближе к одному из центров, чем к любому другому центру.

Итак, метод *k*-средних (и его вариант — метод *k*-медоидов) во многих случаях позволяет разбить данные на кластеры в автоматическом режиме. Он отлично работает для кластеров, форма которых похожа на круг с нормальным (гауссовым) распределением точек внутри круга. Если данные образуют кластеры сложной формы, как, например, на рисунке 8, этот метод дает неправильный результат.

Второй недостаток метода — он не учитывает аномалии. Если в задаче с аномалиями все же хочется использовать именно этот метод, можно считать каждую группу аномалий отдельным кластером и при вычислении ответов на вопрос задачи не учитывать эти «аномальные» кластеры. К сожалению, такой прием тоже не всегда приводит к правильному ответу.

11. Выводы

В настоящей статье рассмотрена методика решения задач на кластеризацию, которые могут предлагаться на ЕГЭ по информатике. Рассмотрены три простых метода кластеризации данных:

- 1) ручное разделение точек на кластеры с помощью разделительных линий или покрытия кластеров прямоугольниками;
- 2) алгоритм DBSCAN;
- 3) метод *k*-средних.

Последние два метода позволяют выделить кластеры автоматически, но не всегда приводят к верному результату: алгоритм DBSCAN не может разделить кластеры, расположенные близко друг к другу; метод *k*-средних неверно выделяет кластеры сложной формы; при его использовании не всегда удастся правильно обрабатывать аномалии.

В следующей таблице сравнивается время работы программы при решении задачи 27 из [3] разными методами. За единицу принято время при ручном выделении кластеров (на компьютере автора оно составило примерно одну секунду).

Метод решения	Время
Ручное разбиение на кластеры	1,0
DBSCAN	8,6
<i>k</i> -средних	2,2

Таким образом, **метод, использующий ручное разделение на кластеры, работает быстрее всего и надежнее всего**. Именно его рекомендуется использовать при решении задачи на кластеризацию на экзамене.

Список источников

1. Баюк Д. А., Березин Д. В., Иванюк В. А. Практическое применение методов кластеризации, классификации и аппроксимации. М.: Прометей, 2020. 448 с.
2. Богданов А. А. Солвер всех прототипов новых 27х на кластеризацию. https://vkvideo.ru/video-142628541_456239405
3. Демонстрационный вариант контрольных измерительных материалов единого государственного экзамена 2025 года по информатике // ФИПИ. https://doc.fipi.ru/ege/demoversii-spezifikacii-kodifikatory/2025/inf_11_2025.zip
4. Кабанов А. М. Задание 27 | КЕГЭ по информатике 2025. https://vkvideo.ru/video-205865487_456240432
5. Лашин В. А. Самое простое решение 27 задания — кластеризация слиянием. <https://rutube.ru/video/82c13ef41e32b3eabc844fd6d40bb3bd/>
6. Хун Чжоу. Машинное обучение сквозь призму Excel: примеры и упражнения / пер. с англ. А. Ю. Гинько. М.: ДМК Пресс, 2025. 270 с.
7. Шастин Л. Д. Задание 27 — Метод DBSCAN. <https://rutube.ru/video/5476cc0a61b984cb9b69d223cfca17ec/>
8. Scitovski R., Sabo K., Martínez-Álvarez F., Ungar Š. Cluster Analysis and Applications. Cham: Springer International Publishing, 2021. 271 p. DOI: 10.1007/978-3-030-74552-3.
9. Vasques X. Machine learning theory and applications: hands-on use cases with Python on classical and quantum machines. Hoboken, New Jersey: Wiley, 2024. 512 p.
10. Wierchoń S., Kłopotek M. Modern Algorithms of cluster analysis. Cham: Springer International Publishing, 2018. 421 p. DOI: 10.1007/978-3-319-69308-8.

* Здесь и далее все числовые значения округлены до двух знаков после запятой.

DOI: 10.32517/2221-1993-2025-24-1-77-84



Т. В. Шамаева

Школа № 2007 ФМШ, г. Москва, Россия

ИСПОЛЬЗОВАНИЕ ЭЛЕКТРОННЫХ ТАБЛИЦ В РЕШЕНИИ ЗАДАНИЙ КЕГЭ ПО ИНФОРМАТИКЕ*

Аннотация

В статье предлагаются методы решения некоторых заданий КЕГЭ по информатике с помощью электронных таблиц как альтернатива традиционно принятому способу решения задач с помощью написания программ. В частности, рассматриваются задания 13, 23, 26, представленные в демоверсии 2025 года.

В силу того что на изучение темы «Программирование» не всегда есть возможность отвести достаточное количество часов в рамках школьного курса, а также из-за того что не все учащиеся успевают глубоко овладеть навыками написания кода, вариант решения заданий с помощью табличных процессоров выглядит весьма реалистичным и привлекательным. Нужно отметить, что во многих случаях таким способом удается решать задания достаточно быстро и точно, так как электронные таблицы позволяют наглядно представить обрабатываемые данные и обучающимся легче в них сориентироваться и найти требуемый ответ.

Статья может быть полезна школьникам, планирующим сдавать КЕГЭ, преподавателям информатики для подготовки учеников к экзамену, а также для изучения темы «Электронные таблицы» в школьном курсе информатики.

Решения задач в статье выполнены с помощью электронного процессора LibreOffice Calc. К статье прилагаются дополнительные материалы в виде файлов с решениями в формате указанного процессора.

Ключевые слова: КЕГЭ, КИМ, информатика, демоверсия, электронная таблица, IP-адрес, маска, сортировка.

Информация об авторе

Шамаева Татьяна Владимировна, учитель информатики, школа № 2007 ФМШ, г. Москва, Россия; *e-mail*: shamaeva-tv@fms2007.ru

1. Введение

Требования к базовому и углубленному уровням изучения информатики, зафиксированные в ФГОС среднего общего образования [16], включают в себя раздел «Информационные технологии», который охва-

тывает вопросы применения информационных технологий, реализованных в прикладных программных продуктах и интернет-сервисах, использования баз данных и электронных таблиц для решения прикладных задач.

Электронные таблицы — мощный компьютерный инструмент, широко применяемый для автоматизации

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_01-2025/

T. V. Shamaeva

School 2007 FMSH, Moscow, Russia

THE USE OF SPREADSHEETS IN SOLVING TASKS FROM THE COMPUTER BASED UNIFIED STATE EXAM IN INFORMATICS

Abstract

The article proposes methods for solving some tasks from the computer based Unified State Exam in informatics using spreadsheets as an alternative to the traditionally accepted method of solving problems by coding. In particular, the tasks 13, 23, 26 presented in the 2025 demo version are considered.

Due to the fact that it is not always possible to allocate a sufficient number of hours to study the theme "Programming" within the school course, and also due to the fact that not all students have time to deeply master the skills of coding, the option of solving tasks using spreadsheet processors looks very realistic and attractive. It should be noted that in many cases this method allows you to solve tasks quite quickly and accurately, since spreadsheets allow you to visually present the data being processed and it is easier for students to navigate them and find the required answer.

The article may be useful for students planning to take computer based Unified State Exam in informatics, informatics teachers to prepare students for the exam, as well as for studying the theme "Spreadsheets" in a school informatics course.

The solutions to the problems in the article are implemented using the LibreOffice Calc electronic processor. Additional materials are attached to the article in the form of files with tasks solutions in the format of the specified spreadsheets.

Keywords: computer based Unified State Exam, computer based USE, test materials, informatics, demo, spreadsheet, IP address, mask, sorting.

Information about the author

Tatyana V. Shamaeva, informatics teacher, school 2007 FMSH, Moscow, Russia; *e-mail*: shamaeva-tv@fms2007.ru

вычислений, анализа и визуализации числовых данных, математического моделирования [2]. Этой теме уделено внимание как в основной [1], так и в старшей [2] школе.

Решения некоторых заданий КЕГЭ с помощью электронных таблиц MS Excel неоднократно рассматривались различными авторами (см., например, задание 8 в работе [6], задание 22 в работе [7]).

В данной статье будут предложены решения заданий 13, 23 и 26, выполненные с помощью офисного пакета LibreOffice (версия 7.3.3.2 (x64)). Для знакомства с программами LibreOffice предусмотрена возможность использования документации [19] и различных учебных материалов (см., например, [2, 14]).

Согласно кодификатору ([4], Кодификатор, п. 2.13 «Проверяемые требования к предметным результатам освоения основной образовательной программы»), в КЕГЭ 2025 проверяется «умение использовать электронные таблицы для анализа, представления и обработки данных». Этот навык и будет использоваться для решения задач, представленных в данной статье.

2. Задание 13

В задании проверяется умение «использовать маску подсети и разделять IP-сети на подсети с помощью масок подсетей» ([4], Кодификатор, п. 1.2). Теория, необходимая для понимания данной задачи, изложена в учебнике К. Ю. Полякова, Е. А. Еремина [12]. В статье К. Ю. Полякова [11] предложен способ решения этой задачи с помощью написания программ на языках программирования Python и PascalABC.Net с подключением специальных библиотек. В данной статье мы рассмотрим вариант решения без программирования, только лишь с помощью встроенных функций табличного процессора. В статье А. А. Емельяновой [5] для получения результата также применяется табличный процессор, но решение основано на переборе вариантов и удобно в использовании для сетей только с небольшим количеством узлов.

Задача Демо-25.

Сеть задана IP-адресом 172.16.168.0 и маской сети 255.255.248.0. Сколько в этой сети IP-адресов, для ко-

торых количество единиц в двоичной записи IP-адреса не кратно 5?

Решение.

Решим задачу, используя правила комбинаторики и электронные таблицы LibreOffice Calc.

Переведем значения октетов в маску и сети в двоичный код с помощью функции ДЕС.В.ДВ() и запишем ответ в восьми разрядах (рис. 1). Формулу удобно распространить из ячейки A1 с помощью маркера автозаполнения.

Маска сети определяет, какая часть IP-адреса узла относится к адресу сети, а какая — к адресу самого узла в этой сети. Значения третьего октета маски покажут, где заканчиваются разряды, соответствующие адресу сети (под «1» маски). Разряды адреса узла под нулевыми разрядами маски относятся к адресу компьютера внутри сети и могут варьироваться (рис. 2).

В изменяемую часть можем добавить такое количество единиц, чтобы общее количество единиц не делилось на 5: $((8 + x) \% 5 \neq 0)$.

По формуле комбинаторики вычислим количество способов, которыми можно расположить единицы по 11 разрядам, т. е. вычислим количество сочетаний из 11 по добавленному числу единиц. В электронных таблицах используем функцию ЧКОМБ().

Общий вид решения в формате отображения значений и формул показан на рисунке 3.

3. Задание 23

Согласно спецификации [4, Спецификация], в задании 23 необходимо «умение анализировать ход исполнения алгоритма». Для достижения этого понадобится использовать прием «динамического программирования, как метод решения задач с сохранением промежуточных результатов, подсчет количества вариантов» [4, Кодификатор, п. 3.16]. Принципы динамического программирования, а также примеры и методы их реализации с помощью программ рассмотрены в [9, 13]. В статье А. А. Чапурных [17] также предложено программное решение этого задания на языках программирования

	A	B	C	D
1	255	255	248	0 Маска
2	172	16	168	0 Сеть
3				
4	=ДЕС.В.ДВ(A1;8)	=ДЕС.В.ДВ(B1;8)	=ДЕС.В.ДВ(C1;8)	=ДЕС.В.ДВ(D1;8)
5	=ДЕС.В.ДВ(A2;8)	=ДЕС.В.ДВ(B2;8)	=ДЕС.В.ДВ(C2;8)	=ДЕС.В.ДВ(D2;8)

Рис. 1. Перевод в двоичную систему

	A	B	C	D	E
1	255	255	248	0 Маска	
2	172	16	168	0 Сеть	
3					
4	11111111	11111111	11111000	00000000	
5	10101100	00010000	10101000	00000000	
6	8 единиц есть		11 разрядов варьируем		
7					
8					

Рис. 2. Анализ маски и адреса сети

Добавим x единиц, чтобы $x + 8$ не делилось на 5	Сочетания из 11 по x	7	Добавим x единиц, чтобы $x + 8$ не делилось на 5	Сочетания из 11 по x
x	$C(11, x)$		x	$C(11, x)$
0	1	12	0	=ЧКОМБ(11;B12)
1	11	13	1	=ЧКОМБ(11;B13)
3	165	14	3	=ЧКОМБ(11;B14)
4	330	15	4	=ЧКОМБ(11;B15)
5	462	16	5	=ЧКОМБ(11;B16)
6	462	17	6	=ЧКОМБ(11;B17)
8	165	18	8	=ЧКОМБ(11;B18)
9	55	19	9	=ЧКОМБ(11;B19)
10	11	20	10	=ЧКОМБ(11;B20)
11	1	21	11	=ЧКОМБ(11;B21)
Ответ:	1663	22	Ответ:	=СУММ(C12:C21)

Рис. 3. Решение задачи

Python и PascalABC.Net с помощью рекурсии. В статье В. С. Попова, Е. А. Алефиренко, Л. Ю. Черницыной [15] анализ решения с помощью программ детализируется моделированием с помощью дерева рекурсивных вызовов, уделяется также внимание меморизации как ключевому отличию метода динамического программирования от рекурсии. В данной статье, опираясь на работу [18], мы рассмотрим решение с использованием электронных таблиц. Промежуточные результаты для подсчета количества вариантов, необходимые при «динамическом» подходе, удобно сохранять в последовательных ячейках таблицы. Покажем это на примере двух задач из демо-версий последних лет [4, Демонстрационный вариант].

Задача Демо-24.

Исполнитель преобразует число на экране.

У исполнителя есть три команды, которые обозначены латинскими буквами:

- А. Прибавить 1
- В. Умножить на 2
- С. Возвести в квадрат

Программа для исполнителя — это последовательность команд. Сколько существует программ, для которых при исходном числе 2 результатом является число 20, при этом траектория вычислений не содержит числа 11?

Решение.

Используя принцип динамического программирования, будем последовательно вычислять результат (т. е. находить количество программ) для всех величин N , от начальной до конечной. Обозначим это количество как K_N . Тогда математические утверждения выглядят так:

$K_N = K_{N-1}$ — для N , не кратных 2 и не являющихся квадратом числа;

$K_N = K_{N-1} + K_{N/2}$ — для N , кратных 2 и не являющихся квадратом числа;

$K_N = K_{N-1} + K_{N^{**0.5}}$ — для N , не кратных 2 и являющихся квадратом числа*;

$K_N = K_{N-1} + K_{N/2} + K_{N^{**0.5}}$ — для N , кратных 2 и являющихся квадратом числа.

В результате задача сводится к поиску K_{20} .

На первом этапе создадим таблицу с пятью столбцами.

В первом укажем величины N от 2 до 20, в трех следующих будем хранить количество программ по каждой из операций, в последнем столбце будем хранить искомое количество программ K_N для каждого N .

Как видно из математической модели, K_{N-1} входит как слагаемое во всех случаях, поэтому удобно для второго столбца указать ссылку на общий результат предыдущей строки и распространить формулу.

Для заполнения третьего и четвертого столбцов потребуются вручную указать ссылки на ячейки или написать более сложную формулу (например, через ДВССЫЛ() или ВПР(), здесь рассматривать такие варианты не будем).

В результирующем столбце хранится сумма по всей строке, т. е. по всем трем видам операций.

Среди видов динамического программирования выделяют метод «динамика вперед». При таком подходе рассматривается, на какую позицию повлияет данная точка. При операции возведения в степень K_N войдет как слагаемое в сумму для $K_{N^{**2}}$. Поэтому для столбца, содержащего операцию возведения в степень, поставим ссылки на ячейки, из которых вычисляется квадрат (стрелки на рисунке 4). Например, для $N = 4$ это K_2 , для $N = 9$ это K_3 . Аналогично для второго столбца K_N войдет как слагаемое в сумму для K_{2*N} , поэтому необходимо поставить ссылки на ячейки, откуда можем получить каждую из четных величин (рис. 4).

Необходимо принять во внимание, что траектория вычислений не содержит числа 11. Чтобы учесть это требование, достаточно выставить ноль в итоговом столбце напротив $K = 11$.

Задача Демо-25.

У исполнителя есть две команды, которые обозначены латинскими буквами:

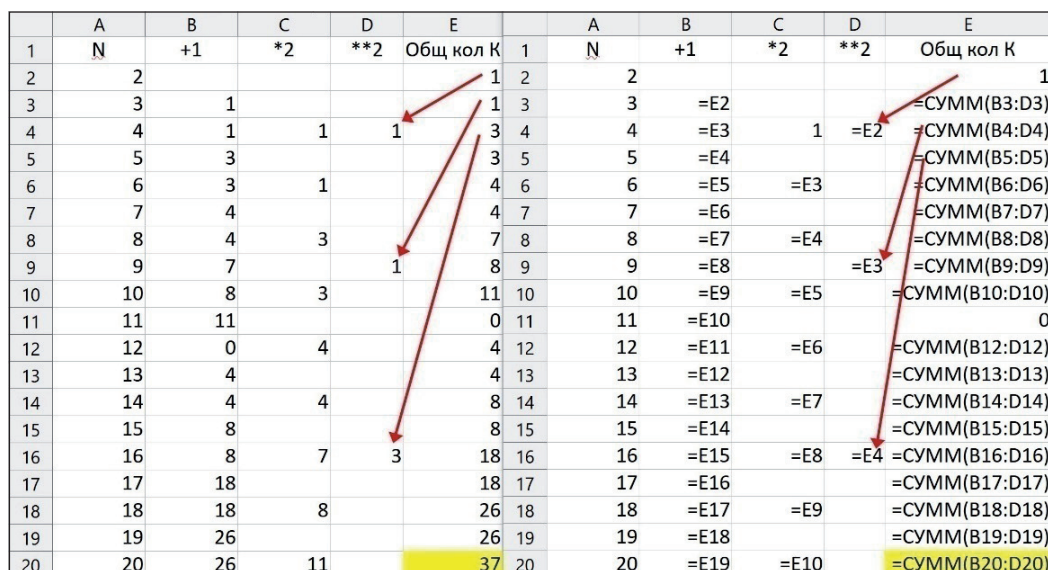
- А. Вычти 2
- В. Найди целую часть от деления на 2

Программа для исполнителя — это последовательность команд. Сколько существует программ, для которых при исходном числе 38 результатом является число 2 и при этом траектория вычислений содержит число 16?

Решение.

Для решения заданий, содержащих обязательное посещение некоторой точки, удобно разбить задачу на две

* $N^{**0.5}$ обозначена операция взятия квадратного корня.



	A	B	C	D	E		A	B	C	D	E
1	N	+1	*2	**2	Общ кол K	1	N	+1	*2	**2	Общ кол K
2	2				1	2	2				1
3	3	1			1	3	3	=E2			=СУММ(B3:D3)
4	4	1	1	1	3	4	4	=E3	1	=E2	=СУММ(B4:D4)
5	5	3			3	5	5	=E4			=СУММ(B5:D5)
6	6	3	1		4	6	6	=E5	=E3		=СУММ(B6:D6)
7	7	4			4	7	7	=E6			=СУММ(B7:D7)
8	8	4	3		7	8	8	=E7	=E4		=СУММ(B8:D8)
9	9	7		1	8	9	9	=E8		=E3	=СУММ(B9:D9)
10	10	8	3		11	10	10	=E9	=E5		=СУММ(B10:D10)
11	11	11			0	11	11	=E10			0
12	12	0	4		4	12	12	=E11	=E6		=СУММ(B12:D12)
13	13	4			4	13	13	=E12			=СУММ(B13:D13)
14	14	4	4		8	14	14	=E13	=E7		=СУММ(B14:D14)
15	15	8			8	15	15	=E14			=СУММ(B15:D15)
16	16	8	7	3	18	16	16	=E15	=E8	=E4	=СУММ(B16:D16)
17	17	18			18	17	17	=E16			=СУММ(B17:D17)
18	18	18	8		26	18	18	=E17	=E9		=СУММ(B18:D18)
19	19	26			26	19	19	=E18			=СУММ(B19:D19)
20	20	26	11		37	20	20	=E19	=E10		=СУММ(B20:D20)

Рис. 4. Использование метода динамического программирования

подзадачи. Посчитаем количество решений от исходной точки до промежуточной, затем посчитаем количество решений от промежуточной точки до конечной. Результатом будет произведение найденных величин.

Из преимуществ такого подхода следует отметить следующие:

- этот метод гарантированно проведет траекторию через обязательную для посещения точку;
- величины, с которыми работаем, будут меньше, чем при решении задачи без разбиения, что сделает решение более наглядным и облегчит поиск ошибок;
- для задач, содержащих только операцию сложения, таблицы будут содержать одинаковые значения в итоговом столбце;
- подзадачи легко копировать с некоторой корректировкой.

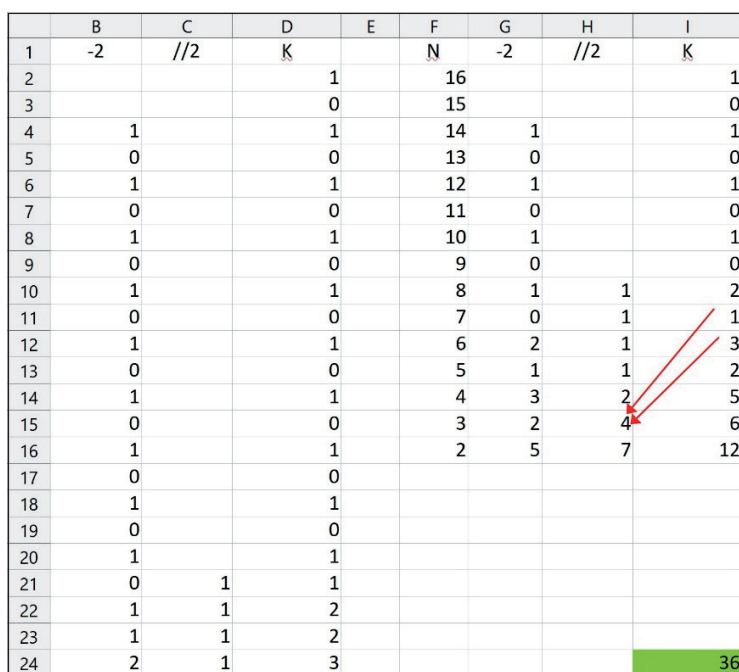
Для данной задачи получим:

$$K_{38-2} = K_{38-16} * K_{16-2}.$$

Особенность этой задачи также состоит в использовании целочисленного деления на два. Нужно отметить, что один и тот же результат мы будем получать из двух точек. Например, $7 // 2 = 3$ и $6 // 2 = 3$. Это значит, что в столбце, соответствующем данной операции, результат будет суммой двух величин: $K_3 = K_7 + K_6$ (рис. 5, 6).

4. Задание 26

Согласно спецификации [4, Спецификация], в задании 26 требуется «умение обрабатывать целочисленную информацию с использованием сортировки». Это задание относится к заданиям высокого уровня сложности и предполагает поиск решения в течение 35 минут. Электронные



	B	C	D	E	F	G	H	I
1	-2	//2	K		N	-2	//2	K
2				1	16			1
3				0	15			0
4	1			1	14	1		1
5	0			0	13	0		0
6	1			1	12	1		1
7	0			0	11	0		0
8	1			1	10	1		1
9	0			0	9	0		0
10	1			1	8	1	1	2
11	0			0	7	0	1	1
12	1			1	6	2	1	3
13	0			0	5	1	1	2
14	1			1	4	3	2	5
15	0			0	3	2	4	6
16	1			1	2	5	7	12
17	0			0				
18	1			1				
19	0			0				
20	1			1				
21	0	1		1				
22	1	1		2				
23	1	1		2				
24	2	1		3				36

Рис. 5. Решение задачи методом динамического программирования

	A	B	C	D	E	F	G	H	I
1	N	-2	//2	K		N	-2	//2	K
2	38				1	16			1
3	37			=СУММ(B3:C3)		15			=СУММ(G3:H3)
4	36	=D2		=СУММ(B4:C4)		14	=I2		=СУММ(G4:H4)
5	35	=D3		=СУММ(B5:C5)		13	=I3		=СУММ(G5:H5)
6	34	=D4		=СУММ(B6:C6)		12	=I4		=СУММ(G6:H6)
7	33	=D5		=СУММ(B7:C7)		11	=I5		=СУММ(G7:H7)
8	32	=D6		=СУММ(B8:C8)		10	=I6		=СУММ(G8:H8)
9	31	=D7		=СУММ(B9:C9)		9	=I7		=СУММ(G9:H9)
10	30	=D8		=СУММ(B10:C10)		8	=I8	=I2	=СУММ(G10:H10)
11	29	=D9		=СУММ(B11:C11)		7	=I9	=I3+I4	=СУММ(G11:H11)
12	28	=D10		=СУММ(B12:C12)		6	=I10	=I5+I6	=СУММ(G12:H12)
13	27	=D11		=СУММ(B13:C13)		5	=I11	=I7+I8	=СУММ(G13:H13)
14	26	=D12		=СУММ(B14:C14)		4	=I12	=I9+I10	=СУММ(G14:H14)
15	25	=D13		=СУММ(B15:C15)		3	=I13	=I11+I12	=СУММ(G15:H15)
16	24	=D14		=СУММ(B16:C16)		2	=I14	=I13+I14	=СУММ(G16:H16)
17	23	=D15		=СУММ(B17:C17)					
18	22	=D16		=СУММ(B18:C18)					
19	21	=D17		=СУММ(B19:C19)					
20	20	=D18		=СУММ(B20:C20)					
21	19	=D19	=D2	=СУММ(B21:C21)					
22	18	=D20	=D3+D4	=СУММ(B22:C22)					
23	17	=D21	=D5+D6	=СУММ(B23:C23)					
24	16	=D22	=D7+D8	=СУММ(B24:C24)					=I16*D24

Рис. 6. Решение задачи методом динамического программирования, отображение формул

таблицы удобно использовать в данном задании, если алгоритм решения не подразумевает вложенных циклов, например, как в демоверсиях 2024 и 2025 годов. Такие задания можно решить значительно быстрее указанного времени с помощью табличного процессора.

Задача Демо-25.

Во время сессии студенты сдают 4 экзамена, за каждый из которых можно получить от 2 до 5 баллов. Студенты, получившие хотя бы одну «двойку», считаются не сдавшими сессию. Результаты сессии публикуются в виде рейтингового списка, в котором сначала указаны идентификационные номера студентов (ID), сдавших сессию, в порядке убывания среднего балла за сессию, а в случае равенства средних баллов — в порядке возрастания ID. Затем располагаются ID студентов, не сдавших сессию: сначала — получивших одну «двойку», затем — получивших две «двойки», потом ID студентов с тремя «двойками» и, наконец, ID студентов, получивших по 2 балла за каждый из экзаменов. Если студенты имеют одинаковое количество «двоек», то их ID в рейтинге располагаются в порядке возрастания. Повышенную стипендию получают студенты, занявшие в рейтинговом списке первые 25 % мест, при условии отсутствия у них «двоек». Гарантируется, что без «двоек» сессию сдали не менее 25 % студентов. Найдите ID студента, который занимает последнее место среди студентов с повышенной стипендией, а также ID первого в рейтинговом списке студента, который имеет более двух «двоек». В ответе запишите два целых положительных числа: сначала ID студента, который занимает последнее место среди студентов с повышенной стипендией, затем ID первого в рейтинговом списке студента, который имеет более двух «двоек».

Входные данные.

В первой строке входного файла находится число N , обозначающее количество студентов (целое положительное число, не превышающее 10 000). Каждая из следую-

щих N строк содержит 5 чисел через пробел: ID студента (целое положительное число, не превышающее 100 000) и четыре оценки, полученные им за сессию. Гарантируется, что общее число студентов N кратно 4 и хотя бы один студент имеет более двух «двоек». Во входном файле все ID различны.

Выходные данные.

Два натуральных числа: искомые ID студентов в порядке, указанном в условии задачи.

Типовой пример организации данных во входном файле:

```
8
4 4 4 4 4
7 5 5 5 2
10 3 4 4 5
1 4 4 4 3
6 3 5 5 3
2 2 2 2 2
13 2 2 2 3
3 3 3 3 3
```

При таких исходных данных рейтинговый список ID имеет вид:

```
4 6 10 1 3 7 13 2.
```

Ответ: 6 13.

Решение.

Откроем текстовый файл в LibreOffice Calc и выберем пробел в качестве разделителя (рис. 7).

Количество студентов N , находящееся в первой строке, перенесем на вторую строку любого свободного столбца и удалим первую строку. Далее в столбце F вычислим значение среднего балла для каждого студента. Для этого в ячейку F1 поместим формулу: =СРЗНАЧ(B1:E1) и распространим ее на весь столбец F. В ячейку G1 поместим формулу: =СЧЁТЕСЛИ(B1:E1;2), подсчитав количество «двоек» у студента, и также распространим эту формулу на весь столбец G.

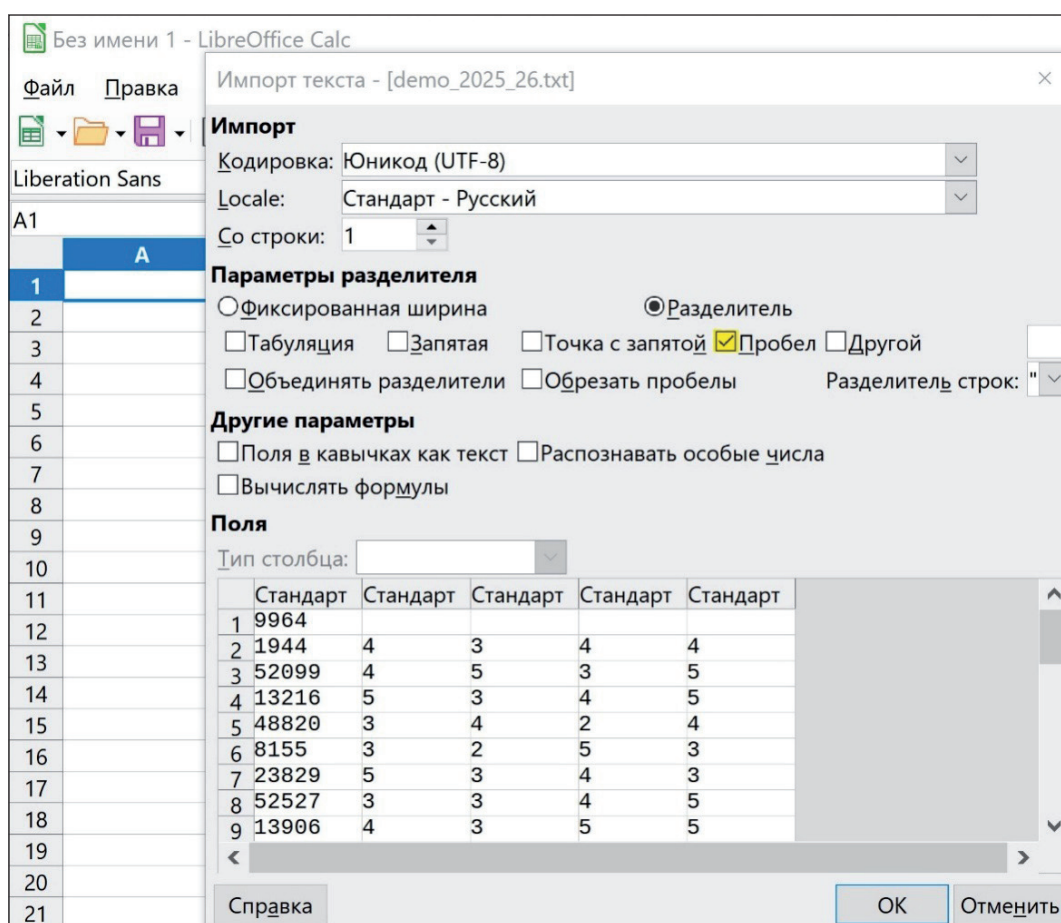


Рис. 7. Настройка параметров открытия текстового файла

Поскольку для решения задачи потребуются сортировки по нескольким параметрам, выделим все столбцы и выберем в меню *Данные, Сортировка*. В открывшемся меню используем три ключа сортировки. Прежде всего отсортируем данные по возрастанию значений столбца G, содержащего количество «двоек».

Таким образом, студенты, не имеющие «двоек», будут находиться в списке сверху. Следующий ключ (столбец F) расположит студентов с равным количеством «двоек» по убыванию среднего балла. Последний ключ применит сортировку по ID в столбце A при равенстве количества «двоек» и среднего балла (рис. 8).

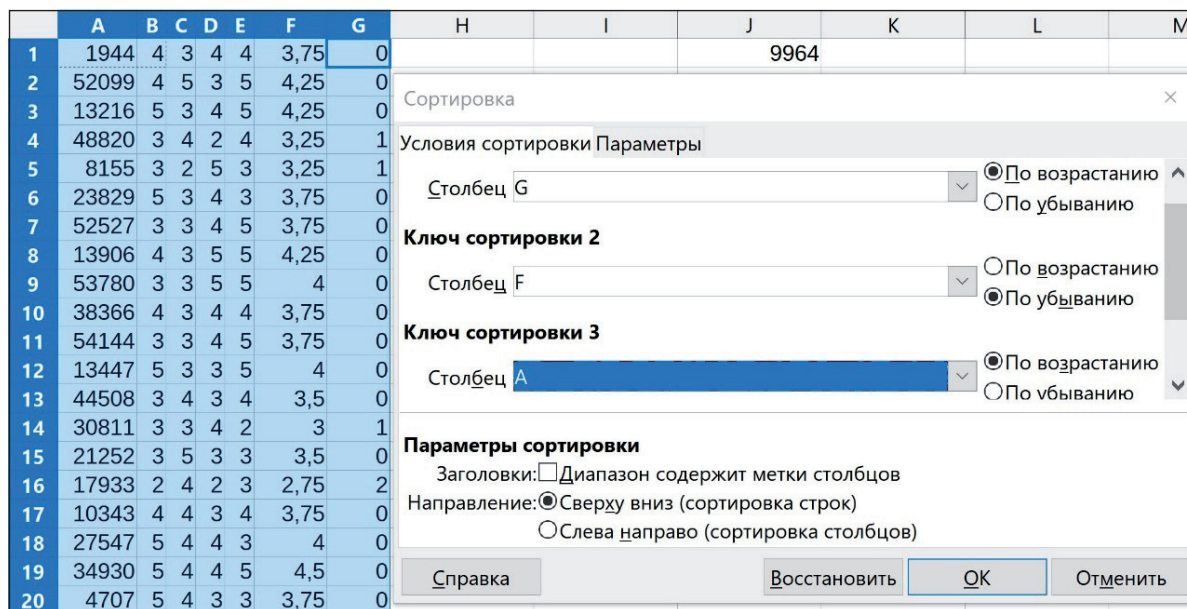


Рис. 8. Настройка параметров сортировки

	A	B	C	D	E	F	G
2487	52128	5	5	4	3	4,25	0
2488	52155	5	3	5	4	4,25	0
2489	52181	5	5	4	3	4,25	0
2490	52268	5	3	5	4	4,25	0
2491	52326	4	5	3	5	4,25	0
2492	52380	5	3	4	5	4,25	0
2493	52435	5	3	4	5	4,25	0
2494	52525	5	3	4	5	4,25	0
2495	52541	3	4	5	5	4,25	0
2496	52563	3	5	5	4	4,25	0

Рис. 9. Ответ на первый вопрос

	A	B	C	D	E	F	G
1	635	2	2	5	2	2,75	3
2	907	2	2	2	4	2,5	3
3	917	2	2	2	5	2,75	3
4	1302	2	3	2	2	2,25	3
5	1474	2	2	4	2	2,5	3
6	1532	2	4	2	2	2,5	3
7	1598	2	2	2	3	2,25	3
8	1853	2	2	2	4	2,5	3
9	2538	2	2	2	3	2,25	3
10	2708	2	2	5	2	2,75	3

Рис. 10. Ответ на второй вопрос

По отсортированным данным найдем ID студента, находящегося на строке с номером 2491 (9964/4). Таким образом, студент с ID = 52326 и есть последний в списке, кто получил повышенную стипендию. Это ответ на первый вопрос задачи (рис. 9).

Далее необходимо найти первого студента, имеющего более двух «двоек», т. е. три штуки. Поскольку для расположения студентов, имеющих «двойки», не важен средний балл, нужно отсортировать интересующих нас студентов заново. Для этого с помощью фильтра выберем студентов, имеющих ровно три «двойки», и перенесем их, например, на новый лист. Далее отсортируем строки только по ID.

Видим, что самый маленький ID = 635 (рис 10). Это ответ на второй вопрос.

5. Заключение

На рассмотренных примерах показано, что с помощью электронных таблиц можно успешно решать некоторые задания КЕГЭ, не прибегая к традиционному написанию программ на языках высокого уровня. В этом случае предполагается использование встроенных функций, написание собственных формул, а также применение других средств электронных процессоров, таких как сортировка, фильтрация и т. п. К преимуществам такого метода можно отнести наглядность обрабатываемых данных и скорость получения ответа. Предлагаемый подход является менее затратным во временном контексте подготовки к экзамену, что, однако, ни в коем случае не умаляет ценности и важности владения навыками программирования.

Для отработки навыков решения и самоподготовки рекомендуется использовать материалы, размещенные на сайте К. Ю. Полякова [10], в Открытом банке заданий [8], а также на ресурсе «Демонстрационная версия станции КЕГЭ» [3].

Список источников

- Босова Л. Л. О новых подходах к изучению школьной информатики в условиях цифровой трансформации общества // Информатика в школе. 2022. № 4. С. 5–14. EDN: DKRLZV. DOI: 10.32517/2221-1993-2022-21-4-5-14.
- Босова Л. Л., Куклина И. Д., Мирончик Е. А. Практикум по работе с электронными таблицами в старшей школе // Информатика в школе. 2018. № 10. С. 3–24. EDN: VOSXLR. DOI: 10.32517/2221-1993-2018-17-10-3-24.

3. Демонстрационная версия станции КЕГЭ. <https://kompege.ru>

4. Единый государственный экзамен по информатике // ФИПИ. Демоверсии, спецификации, кодификаторы. <https://fipi.ru/ege/demoversii-specifikacii-kodifikatory#1/tab/151883967-5>

5. Емельянова А. А. Различные способы решения задания № 13 КЕГЭ 2024 на IP адреса и маски // Наука и образование: актуальные вопросы, достижения и инновации. Сборник статей X Международной научно-практической конференции (Пенза, 12 октября 2023 года). Пенза: Наука и Просвещение (ИП Гуляев Г. Ю.), 2023. С. 38–44. EDN: IICUKE.

6. Златопольский Д. М. Выполняем задание 8 ЕГЭ по информатике в среде электронной таблицы // Мир информатики. 2023. № 76. С. 1–8.

7. Златопольский Д. М. Методика выполнения задания 22 демонстрационного варианта ЕГЭ по информатике 2023 года // Мир информатики. 2023. № 75. С. 1–8.

8. Открытый банк тестовых заданий // ФИПИ. <https://ege.fipi.ru/bank>

9. Поляков К. Ю. Динамическое программирование в задачах обработки последовательностей ЕГЭ по информатике // Информатика в школе. 2020. № 5. С. 55–63. EDN: SXZKLG. DOI: 10.32517/2221-1993-2020-19-5-55-63.

10. Поляков К. Ю. Материалы для подготовки к ЕГЭ по информатике (2025). <https://kpolyakov.spb.ru/school/ege.htm>

11. Поляков К. Ю. Программное решение задач ЕГЭ по информатике на IP-адреса и маски подсетей // Информатика в школе. 2024. № 2. С. 73–80. EDN: FQIZOZ. DOI: 10.32517/2221-1993-2024-23-2-73-80.

12. Поляков К. Ю., Еремин Е. А. Информатика. 10 класс (базовый и углубленный уровни). В 2 ч.: учебник. М.: БИНОМ. Лаборатория знаний, 2019. Ч. 1. 352 с.; Ч. 2. 352 с.

13. Поляков К. Ю., Еремин Е. А. Информатика. 11 класс (базовый и углубленный уровни). В 2 ч.: учебник. М.: БИНОМ. Лаборатория знаний, 2019. Ч. 1. 240 с.; Ч. 2. 304 с.

14. Пономарёва Е. А. Электронные таблицы (табличный процессор). Обработка информации в среде табличного процессора: учебно-методическое пособие. Пермь: ИПЦ «Прокрость», 2022. С. 20–25, 29–30.

15. Попов В. С., Алефиренко Е. А., Черницына Л. Ю. Рекурсивная функция для решения задания на динамическое программирование: от беззнания формы компетенции к знанию и пониманию // Информатика в школе. 2024. № 5. С. 85–94. EDN: MDGTEQ. DOI: 10.32517/2221-1993-2024-23-5-85-94.

16. Федеральный государственный образовательный стандарт среднего общего образования. <https://base.garant.ru/70188902/8ef641d3b80ff01d34be16ce9baf6e0/>

17. Чапурных А. А. Различные способы решения задания № 23 КЕГЭ на перебор вариантов и динамическое программирование // Наука, общество, образование в условиях цифровизации и глобальных изменений. Сборник статей III Междуна-

родной научно-практической конференции (Пенза, 20 августа 2022 года). Пенза: Наука и Просвещение (ИП Гуляев Г. Ю.), 2022. С. 44–48. EDN: LAKNFW.

18. *Шамаева Т. В.* «Динамическое программирование» с помощью MS Excel на примере 23 номера из КЕГЭ по информатике // Архимед: Научно-методический сборник / отв. за

выпуск А. Рубин, Т. Струков, П. Чулков, А. Шевкин. Вып. 18. М.: ООО «МАКС Пресс», 2022. С. 124–136. EDN: RTVFKK. https://www.elemschool.mathedu.ru/text/arhimed_2022_v18/p116/

19. LibreOffice. English documentation. <https://documentation.libreoffice.org/en/english-documentation/>



ИТОГИ XXVIII ВСЕРОССИЙСКОГО КОНКУРСА ЦИФРОВЫХ ИЗОБРАЖЕНИЙ И ФОТОГРАФИЙ ЖУРНАЛА «ИНФОРМАТИКА В ШКОЛЕ» ФОТО 1-2025

Уважаемые коллеги!

В октябре 2024 года издательство «Образование и Информатика» объявило очередной конкурс цифровых изображений и фотографий для размещения их на обложке журнала «Информатика в школе» первого полугодия 2025 года.

В конкурсе приняли участие школьники, студенты и педагоги из России и Беларуси.

Благодарим всех конкурсантов за участие и представляем авторов победивших работ.

В номинации «Цифровые изображения» победителями стали:

Аникина Оксана Александровна,

студентка Мордовского государственного педагогического университета имени М. Е. Евсевьева, г. Саранск, Республика Мордовия, Россия

Ахметшин Аяз Айратович,

студент Мамадышского политехнического колледжа, г. Мамадыш, Республика Татарстан, Россия
(руководитель работы — Порываева Нафиса Сабитовна)

Ахтямов Богдан Айдарович,

студент Мамадышского политехнического колледжа, г. Мамадыш, Республика Татарстан, Россия
(руководитель работы — Порываева Нафиса Сабитовна)

Бартенева Виктория Викторовна,

ученица VII класса средней общеобразовательной школы № 41 им. В. В. Сизова, г. Курск, Россия
(руководитель работы — Скляр Елен Александровна)

Быстрова Софья Валерьевна,

ученица V класса средней школы № 1 им. А. М. Горького, г. Фролово, Волгоградская область, Россия
(руководитель работы — Фирсова Мария Николаевна)

Вишневская Светлана Леонидовна,

ученица IX класса гимназии № 7 «Сибирская», г. Новосибирск, Россия
(руководитель работы — Федорова Елена Александровна)

Желнина Виктория Юрьевна,

студентка Барнаульского государственного педагогического колледжа имени Василия Константиновича Штильке, г. Барнаул, Алтайский край, Россия
(руководитель работы — Лукьянова Наталия Владимировна)

Катаева Аксения Антоновна,

ученица XI класса средней общеобразовательной школы № 31 имени Героя Советского Союза С. Д. Василичина, г. Владимир, Россия

(руководитель работы — Левушкина Арина Сергеевна)

Краснова Ульяна Денисовна,

ученица VII класса гимназии № 1, г. Новосибирск, Россия
(руководитель работы — Лейман Екатерина Владимировна)

Кузнецова Ольга Александровна,

ученица XI класса лицея № 31, г. Саранск, Республика Мордовия, Россия
(руководитель работы — Захарова Кристина Валерьевна)

Куртукова Ярослава Артемовна,

ученица VII класса гимназии № 1, г. Новосибирск, Россия
(руководитель работы — Лейман Екатерина Владимировна)

Латыпов Дамир Артемович,

студент Мамадышского политехнического колледжа, г. Мамадыш, Республика Татарстан, Россия
(руководитель работы — Порываева Нафиса Сабитовна)

Малашкина Ксения Алексеевна,

студентка Барнаульского государственного педагогического колледжа имени Василия Константиновича Штильке, г. Барнаул, Алтайский край, Россия
(руководитель работы — Лукьянова Наталия Владимировна)

Марголина Маргарита Борисовна,

ученица VII класса средней школы № 206 им. А. М. Кижеватова, г. Минск, Беларусь
(руководитель работы — Гаевская Вера Алексеевна)

Маслов Максимильян Александрович,

учащийся объединения «Информатика» Центра технического творчества, г. Киров, Кировская область;
ученик VII класса средней общеобразовательной школы № 58, г. Киров, Кировская область, Россия
(руководитель работы — Обухова Галина Геннадьевна)

Миннекаев Гадель Ильсурович,

студент Мамадышского политехнического колледжа, г. Мамадыш, Республика Татарстан, Россия
(руководитель работы — Порываева Нафиса Сабитовна)

Михайлова Александра Сергеевна,

студентка Барнаульского государственного педагогического колледжа
имени Василия Константиновича Штильке, г. Барнаул, Алтайский край, Россия
(руководитель работы — Лукьянова Наталья Владимировна)

Набиуллина Азалия Ильфатовна,

студентка Мамадышского политехнического колледжа, г. Мамадыш, Республика Татарстан, Россия
(руководитель работы — Порываева Нафиса Сабитовна)

Огородникова Мария Андреевна,

учащаяся объединения «Информатика» Центра технического творчества, г. Киров, Кировская область;
ученица V класса средней общеобразовательной школы № 57, г. Киров, Кировская область, Россия
(руководитель работы — Обухова Галина Геннадьевна)

Петунина Варвара Олеговна,

студентка Барнаульского государственного педагогического колледжа
имени Василия Константиновича Штильке, г. Барнаул, Алтайский край, Россия
(руководитель работы — Лукьянова Наталья Владимировна)

Петухова Дина Михайловна,

студентка Мамадышского политехнического колледжа, г. Мамадыш, Республика Татарстан, Россия
(руководитель работы — Порываева Нафиса Сабитовна)

Сиротенко Мария Павловна,

ученица V класса гимназии № 1, г. Новосибирск, Россия
(руководитель работы — Лейман Екатерина Владимировна)

Трусова Арина Сергеевна,

студентка Мордовского государственного педагогического университета имени М. Е. Евсевьева, г. Саранск,
Республика Мордовия, Россия

Тюрикова Виктория Владимировна,

студентка Барнаульского государственного педагогического колледжа
имени Василия Константиновича Штильке, г. Барнаул, Алтайский край, Россия
(руководитель работы — Лукьянова Наталья Владимировна)

Хилькевич Анна Андреевна,

ученица VIII класса средней школы № 206 им. А. М. Кижеватова, г. Минск, Беларусь
(руководитель работы — Гаевская Вера Алексеевна)

Чембарова София Александровна,

ученица VII класса средней школы № 206 им. А. М. Кижеватова, г. Минск, Беларусь
(руководитель работы — Гаевская Вера Алексеевна)

Щемелев Артём Михайлович,

учащийся Донецкого республиканского дворца детского и юношеского творчества, г. Донецк,
Донецкая Народная Республика;
ученик VII класса профильной гимназии № 122, г. Донецк, Донецкая Народная Республика, Россия
(руководитель работы — Гончарова Татьяна Семеновна)

Юдин Михаил Николаевич,

студент Мордовского государственного педагогического университета имени М. Е. Евсевьева, г. Саранск,
Республика Мордовия, Россия

Ярмухаметова Диана Дамировна,

студентка Мордовского государственного педагогического университета имени М. Е. Евсевьева, г. Саранск,
Республика Мордовия, Россия

В номинации «Цифровые фотографии» победителями стали:

Жучков Андрей Сергеевич,

студент Омского государственного педагогического университета, г. Омск, Россия

Крекин Михаил Евгеньевич,

учитель информатики, специалист по системному администрированию школы № 1228 «Лефортово»,
г. Москва, Россия

Склярова Елена Александровна,

учитель информатики средней общеобразовательной школы № 41 им. В. В. Сизова, г. Курск, Россия

Степанова Раиса Александровна,

учитель информатики Кюсюрской средней общеобразовательной школы, с. Кюсюр, Булунский улус,
Республика Саха (Якутия), Россия

Ямов Виктор Александрович,

студент Мордовского государственного педагогического университета имени М. Е. Евсевьева, г. Саранск,
Республика Мордовия, Россия

Все победители конкурса получают (бесплатно):

- диплом от издательства «Образование и Информатика»;
- печатный экземпляр журнала «Информатика в школе» № 1-2025 с итогами конкурса;
- авторский печатный экземпляр журнала, на обложке которого будет опубликована работа.

Остальные конкурсанты могут получить сертификат об участии, который будет подготовлен по индивидуальному запросу.

Спасибо за внимание и участие в конкурсе!

Следите за информацией о новых конкурсах в журналах «Информатика в школе» и «Информатика и образование», а также на сайте ИНФО: <http://www.infojournal.ru/>

В оформлении обложки данного номера журнала «Информатика в школе» использованы работы следующих авторов:



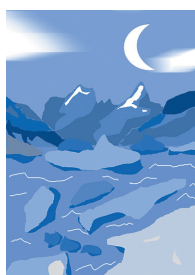
Азалия Набиуллина



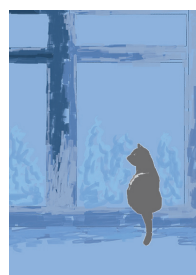
Артём Щемелев



Аяз Ахметшин



Варвара Петунина



Виктория Тюрикова



Ульяна Краснова



Ярослава Куртукова



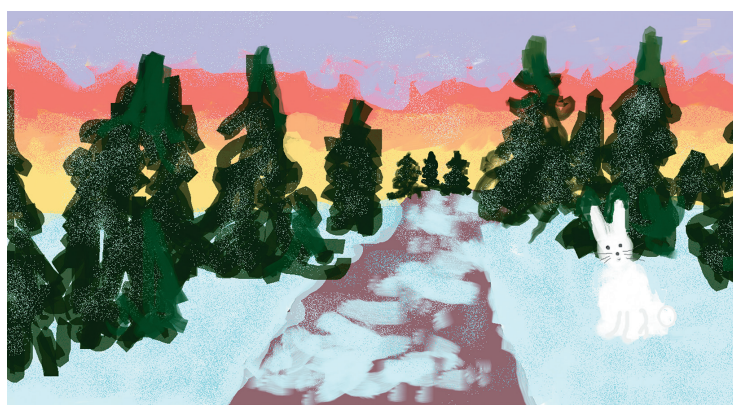
А. С. Жучков



Александра Михайлова



Софья Быстрова



Мария Огородникова

Уважаемые авторы и читатели!

Издательство «Образование и Информатика» объявляет о проведении конкурса цифровых изображений и фотографий, которые будут размещены на обложке журнала «Информатика в школе» во втором полугодии 2025 года.

Подробности — на сайте издательства «Образование и Информатика»: <http://infojournal.ru/>

ПОДПИСКА

Журнал «Информатика в школе»

Индекс подписки
на 1-е полугодие 2025 года
(«Урал-Пресс», «АРЗИ» и другие агентства подписки)
81407

Периодичность выхода: 3 номера в полугодие (февраль, апрель, июнь)
Объем — не менее 88 полос

Редакционная стоимость — 500 руб.

ПРАВИЛА ДЛЯ АВТОРОВ

Уважаемые коллеги!

Статьи для публикации в журналах «Информатика и образование» и «Информатика в школе» должны отправляться в редакцию **только через электронную форму на сайте ИНФО (раздел «Авторам → Отправка статьи»):**

<http://infojournal.ru/authors/send-article/>

Обращаем ваше внимание, что для отправки статьи необходимо предварительно зарегистрироваться на сайте ИНФО (или авторизоваться — для зарегистрированных пользователей).

С требованиями к оформлению представляемых для публикации материалов можно ознакомиться на сайте ИНФО в разделе «Авторам»:

<http://infojournal.ru/authors/>

Обратите внимание: требования к оформлению файла рукописи — **разные** для журналов «Информатика и образование» и «Информатика в школе». При подготовке файла рукописи ориентируйтесь на требования для того журнала, в который вы представляете статью. Если вы представляете рукопись в оба журнала (для публикации в одном из изданий — на усмотрение редакции), при ее оформлении следует руководствоваться требованиями к оформлению рукописи в журнал «Информатика и образование».

Дополнительную информацию можно получить в разделе **«Авторам → Часто задаваемые вопросы»:**

<http://infojournal.ru/authors/faq/>

а также в редакции ИНФО:

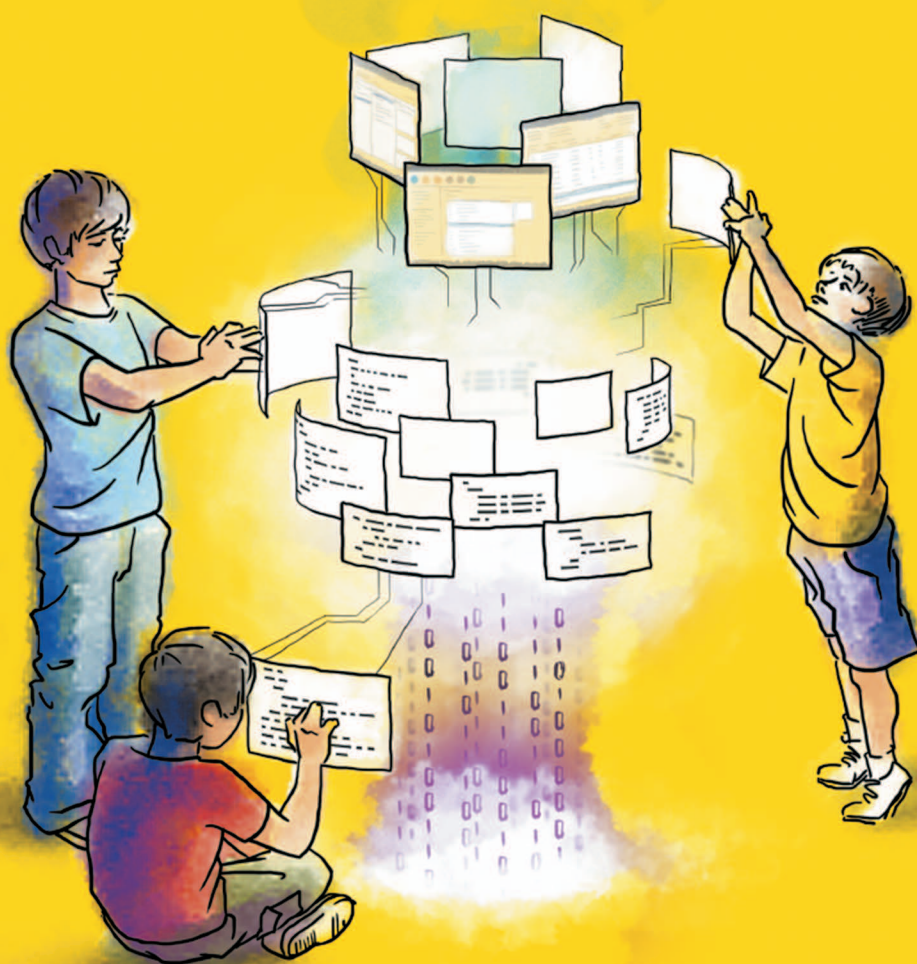
E-mail: readinfo@infojournal.ru

Телефон: +7 (495) 140-19-86



Программируем будущее вместе

Курсы для школьников 8-17 лет
Онлайн и очно по всей стране
Старт занятий 28 сентября



Тысячи наших выпускников уже работают программистами

Тел.: +7 (495) 688-90-02, линия 4
E-mail: teen@1c.ru, club.1c.ru





29 - 30 сентября
международный конгресс

Суперкомпьютерные дни в России 2025

<https://Congress.RussianSCDays.org>

Научная конференция:
29.09 - 30.09.2025

Семинары

Выставка

Экскурсии

Одно из ключевых IT-событий года, каждый сентябрь собирающее наиболее квалифицированных специалистов в области высокопроизводительных систем и параллельной обработки данных.

ТЕМАТИКА мероприятий конгресса — суперкомпьютерные технологии во всем многообразии: параллельные и распределенные вычисления, высокопроизводительные программные и аппаратные решения, масштабируемые алгоритмы, индустриальные суперкомпьютерные решения, большие данные, машинное обучение, суперкомпьютерное образование и многое другое.

АУДИТОРИЯ — российские и зарубежные представители науки, промышленности, бизнеса, образования, государственных органов.

Мы пытаемся понять законы мира, которые быстро меняются. Быстро реагировать на самые сложные актуальные проблемы нам помогают суперкомпьютеры. С помощью технологий математического моделирования, обработки больших данных и искусственного интеллекта решаются сверхзадачи в науке, промышленности, бизнесе.

Два исключительно наполненных суперкомпьютерными событиями дня

МЕСТО

Центральное событие конгресса — научная конференция — будет проходить в Шуваловском корпусе Московского государственного университета имени М.В.Ломоносова.



НАУЧНАЯ КОНФЕРЕНЦИЯ — это бесценный опыт общения с единомышленниками, партнерами, лидерами рынка и представителями индустрии, которые заинтересованы в использовании представленных разработок на практике. Быстрее отвечая на вызовы нового времени, вместе мы делаем мир лучше, экологичнее и технологичнее.

Конференция посвящена 270-летию МГУ имени М.В. Ломоносова

Рабочие дни конференции: 29.09 - 30.09.2025

<https://RussianSCDays.org>



СТАНЬТЕ ЧАСТЬЮ «СУПЕРКОМПЬЮТЕРНЫХ ДНЕЙ В РОССИИ»!

Посетите конференцию и научные школы, узнайте о работе ведущих российских и мировых суперкомпьютерных центров, организуйте свое мероприятие в рамках конгресса!

Приглашаем к организации семинаров и мастер-классов суперкомпьютерного конгресса! Семинары могут проводиться удаленно на различных площадках в пределах России.

Приглашаем принять участие в выставке!