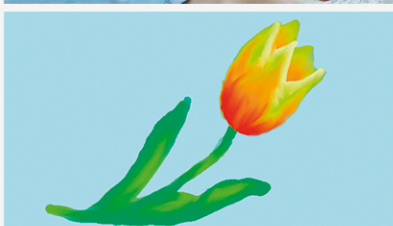
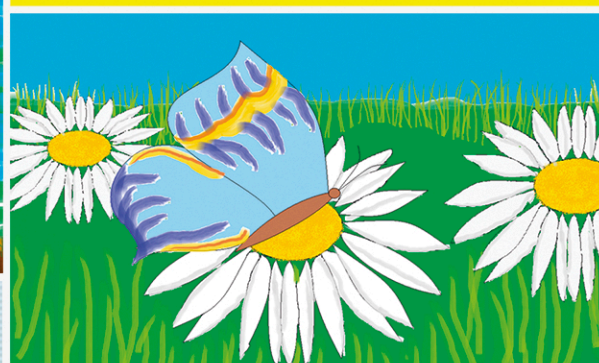
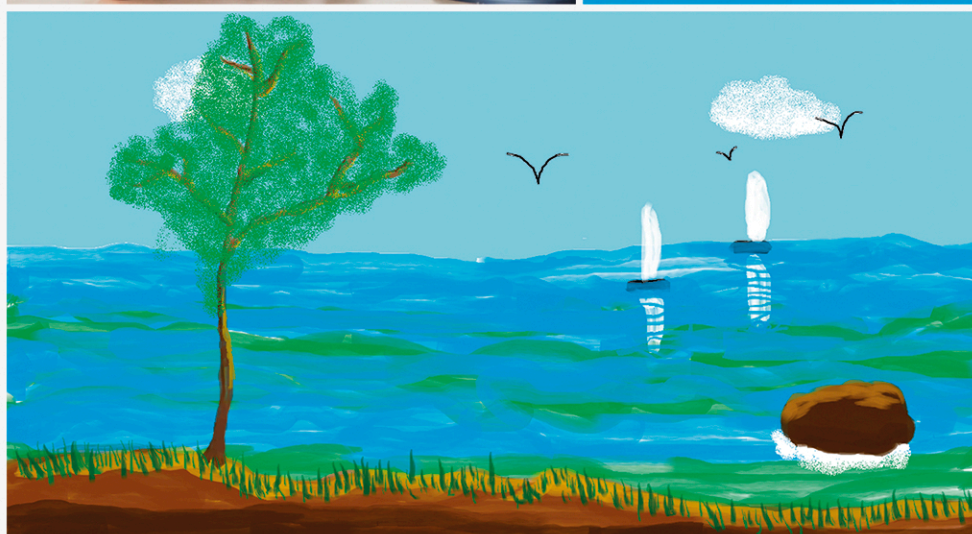
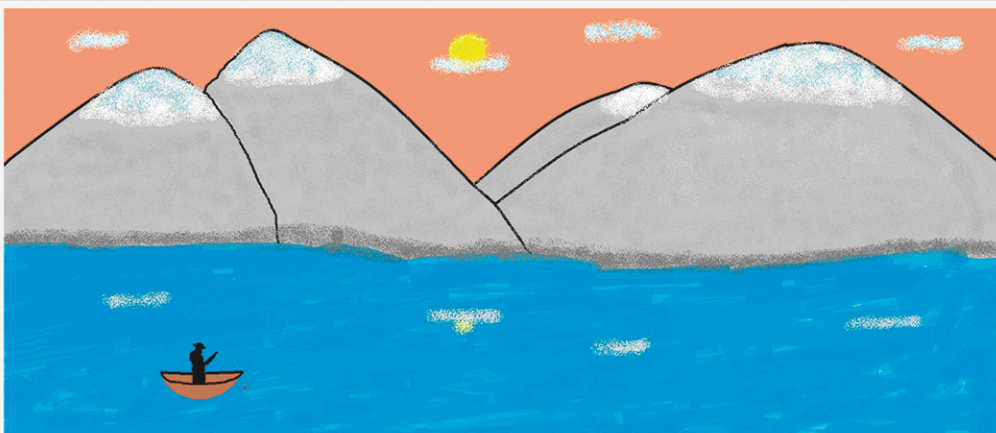


ИНФОРМАТИКА В ШКОЛЕ

ISSN 2221-1993

№ 3'2022

www.infojournal.ru





24 сентября - 1 октября
международный конгресс

Суперкомпьютерные дни в России 2022

<https://Congress.RussianSCDays.org>

Научные школы:
24.09 - 01.10.2022

Научная конференция:
26.09 - 27.09.2022

Семинары

Выставка

Экскурсии

Новый расширенный формат объединяет научную конференцию, научные школы Суперкомпьютерной академии, серию специализированных научных семинаров, экскурсии в ведущие суперкомпьютерные центры и множество других событий, проводимых на различных площадках Москвы и России.

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.

НАУЧНЫЕ ШКОЛЫ СУПЕРКОМПЬЮТЕРНОЙ АКАДЕМИИ — это специализированные мероприятия по актуальным направлениям развития науки и технологий, организуемые и проходящие под руководством известных российских специалистов.

Рабочие дни академии: 24.09 - 01.10.2022

<https://academy.hpc-russia.ru/>

ОДНА НЕДЕЛЯ — МНОЖЕСТВО ПАРАЛЛЕЛЬНЫХ СОБЫТИЙ!

КЛЮЧЕВЫЕ ДАТЫ НАУЧНОЙ КОНФЕРЕНЦИИ

До 15 апреля — представление полных версий работ

15 мая — уведомление о включении работы в программу конференции

30 мая — представление окончательного варианта работы

НАУЧНАЯ КОНФЕРЕНЦИЯ — это множество параллельно идущих секций: выступления мировых лидеров HPC-сообщества, научные и индустриальные секции, постерная секция, конференция молодых ученых. Совещания, круглые столы, живые дискуссии, обмен опытом и инновациями.

Рабочие дни конференции: 26.09 - 27.09.2022

<https://RussianSCDays.org>

РЕГИСТРАЦИЯ
участников
конференции:
с 15 апреля

<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!

ИНФОРМАТИКА В ШКОЛЕ

ИЗДАЕТСЯ С СЕНТЯБРЯ 2002 ГОДА

№ 3 (176) июнь 2022

Содержание

КОНКУРСЫ

Итоги Всероссийского конкурса «Школьная информатика: вчера, сегодня, завтра».....	3
Курганова Н. А. Использование квадрокоптеров Tello EDU при изучении раздела «Основы алгоритмизации» школьного курса информатики.....	5
Шакуров А. Информационные технологии в школе будущего.....	12
Климина Н. В. Виртуальная робототехника на базе робосимуляторов VEXcode VR и Open Roberta Lab.....	13
Батракова В. О., Веснин С. Н., Корепанова А. А., Лаптева Т. Д., Саврасов А. И. Элективный курс «Применение методов моделирования систем при работе с текстовой информацией».....	27
Плаксин М. А. Игры-ходилки для изучения блок-схем.....	36
Алексеева А. Путешествие в школу будущего.....	49
Жумашева С. Завтра — уже сегодня.....	50
Перминов Н. Информатике быть!.....	51
Шевчук Е. Пусть меня научат!.....	52

МЕТОДИЧЕСКАЯ КОПИЛКА

Маркушевич М. В. Элементы методики преподавания компьютерной трехмерной графики на уровне основного общего образования с использованием КОМПАС-3D LT V12.....	53
--	----

ОЛИМПИАДЫ ПО ИНФОРМАТИКЕ

Самылкина Н. Н., Калинин И. А. Решение задач отборочного и основного этапов первой всероссийской олимпиады по искусственному интеллекту для учащихся VIII—XI классов.....	67
--	----

СРЕДСТВА ИКТ В УЧЕБНОМ ПРОЦЕССЕ

Завьялова О. А., Маркелов В. К. Возможности онлайн-сред программирования при обучении языку Python в школе.....	75
--	----

ИНФОРМАТИКА И ИКТ В НАЧАЛЬНОЙ ШКОЛЕ

Князева Н. К. Формирование навыков коммуникации и кооперации у младших школьников при создании анимации на уроках математики.....	83
--	----

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 30.06.22.
Формат 60×90^{1/8}. Усл. печ. л. 11,5
Тираж 2000 экз. Заказ № 1729.
Отпечатано в типографии ООО «Принт сервис групп»,
105187, Россия, г. Москва, Борисовская ул., д. 14, стр. 6,
тел./факс: +7 (499) 785-05-18, e-mail: 3565264@mail.ru

© «Образование и Информатика», 2022



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
akulll@mail.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»
БОСОВА Людмила Леонидовна

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
ГУБКИН Владислав Александрович

Отдел распространения и рекламы
КУЗНЕЦОВА Елена Александровна

НАУЧНО-ПРАКТИЧЕСКИЙ ЖУРНАЛ «ИНФОРМАТИКА В ШКОЛЕ»

УЧРЕДИТЕЛЬ:

ИЗДАТЕЛЬСТВО
«ОБРАЗОВАНИЕ И ИНФОРМАТИКА»

ГЛАВНЫЙ РЕДАКТОР

БОСОВА Людмила Леонидовна

доктор пед. наук, доцент, заслуженный учитель РФ, лауреат премии Правительства РФ в области образования, Институт математики и информатики Московского педагогического государственного университета, зав. кафедрой теории и методики обучения математике и информатике

РЕДАКЦИОННАЯ КОЛЛЕГИЯ

БАРАКИНА Татьяна Вячеславовна

канд. пед. наук, доцент, Омский государственный педагогический университет, доцент кафедры предметных технологий начального и дошкольного образования

БЕШЕНКОВ Сергей Александрович

доктор пед. наук, профессор, Институт управления образованием РАО, Москва, руководитель Центра информатизации образования

ВОЕВОДИН Владимир Валентинович

чл.-корр. РАН, доктор физ.-мат. наук, профессор, лауреат премии Правительства РФ в области образования, Научно-исследовательский вычислительный центр МГУ имени М. В. Ломоносова, директор

ДЕРГАЧЕВА Лариса Михайловна

канд. пед. наук, доцент, научный редактор издательства «Образование и Информатика»

ЗАСЛАВСКАЯ Ольга Юрьевна

доктор пед. наук, профессор, Институт цифрового образования Московского городского педагогического университета, научный руководитель департамента информатизации образования

ЗЕНКИНА Светлана Викторовна

доктор пед. наук, профессор, Академия социального управления, Московская область, профессор кафедры общеобразовательных дисциплин

ЗУБРИЛИН Андрей Анатольевич

канд. филос. наук, доцент, Мордовский государственный педагогический университет имени М. Е. Евсевьева, г. Саранск, зав. кафедрой информатики и вычислительной техники

КИРИЧЕНКО Ирина Борисовна

заместитель главного редактора журнала «Информатика в школе»

ЛАПТЕВ Владимир Валентинович

академик РАО, доктор пед. наук, канд. физ.-мат. наук, профессор, Российский государственный педагогический университет им. А. И. Герцена, Санкт-Петербург, первый проректор

ЛЕВЧЕНКО Ирина Витальевна

доктор пед. наук, профессор, Институт цифрового образования Московского городского педагогического университета, научный руководитель департамента информатики, управления и технологий

РЫБАКОВ Даниил Сергеевич

канд. пед. наук, доцент, директор издательства «Образование и Информатика»

СЛИНКИНА Ирина Николаевна

канд. пед. наук, доцент, Шадринский государственный педагогический университет, декан факультета информатики, математики и естественных наук

ФЕДОСОВ Александр Юрьевич

доктор пед. наук, доцент, Российский государственный социальный университет, Москва, профессор факультета информационных технологий



ИТОГИ ВСЕРОССИЙСКОГО КОНКУРСА «ШКОЛЬНАЯ ИНФОРМАТИКА: ВЧЕРА, СЕГОДНЯ, ЗАВТРА»

Уважаемые коллеги!

В апреле 2022 года издательство «Образование и Информатика» и редакция журнала «Информатика в школе» объявили конкурс «Школьная информатика: вчера, сегодня, завтра».

Конкурс проводился по двум номинациям:

1. Номинация для педагогов: Школьная информатика: вчера, сегодня, завтра.
2. Номинация для учащихся: Школьная информатика: заглянем в будущее.

В конкурсе приняли участие работники образования — учителя, преподаватели вузов, педагоги системы дополнительного образования, — а также студенты вузов, колледжей, учащиеся общеобразовательных организаций из разных регионов Российской Федерации и из Беларуси.

Благодарим всех авторов за интересные работы!

Представляем лауреатов (1-е место) и дипломантов (2-е место) конкурса. Все победители представлены в алфавитном порядке.

НОМИНАЦИЯ «ШКОЛЬНАЯ ИНФОРМАТИКА: ВЧЕРА, СЕГОДНЯ, ЗАВТРА»

Лауреат конкурса (1-е место)

Курганова Наталья Александровна,
доцент кафедры прикладной информатики и математики, Омский государственный педагогический университет, г. Омск, Россия

Дипломанты конкурса (2-е место)

Батракова Виктория Олеговна,
учитель информатики, предметно-языковая школа «Дуплекс», г. Пермь, Россия

Веснин Сергей Николаевич,
учитель информатики, предметно-языковая школа «Дуплекс», г. Пермь, Россия

Корепанова Алла Александровна,
студентка 1-го курса магистратуры, Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия

Лаптева Татьяна Дмитриевна,
студентка 1-го курса магистратуры, Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия

Саврасов Антон Игоревич,
*учитель информатики, предметно-языковая школа «Дуплекс», г. Пермь;
студент 1-го курса магистратуры, Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия*

Климина Наталья Владимировна,
педагог дополнительного образования, структурное подразделение, реализующее дополнительные общеобразовательные программы, «Дворец творчества детей и молодежи» средней общеобразовательной школы № 14 «Центр образования» имени кавалера ордена Ленина Н. Ф. Шутова, г. Сызрань, Самарская область, Россия

Плаксин Михаил Александрович,
доцент кафедры информационных технологий в бизнесе, Национальный исследовательский университет «Высшая школа экономики» (Пермский филиал), г. Пермь, Россия

НОМИНАЦИЯ

«ШКОЛЬНАЯ ИНФОРМАТИКА: ЗАГЛЯНЕМ В БУДУЩЕЕ»

Лауреат конкурса (1-е место)

Шакуров Айзат Рифатович,

ученик VIII класса, лицей-интернат № 24, Дворец творчества детей и молодежи имени Ильдуса Харисовича Садыкова, г. Нижнекамск, Республика Татарстан, Россия
(руководитель работы — Шарыгина Марина Николаевна)

Дипломанты конкурса (2-е место)

Алексеева Александра Андреевна,

ученица VI класса, средняя общеобразовательная школа № 567 Петродворцового района Санкт-Петербурга, Россия
(руководитель работы — Новикова Ирина Иосифовна)

Жумашева Снежана Радиковна,

студентка Оренбургского учетно-финансового техникума, Оренбург, Россия
(руководитель работы — Бардюгова Ирина Николаевна)

Перминов Николай Антонович,

ученик VII класса, учебно-консультационный пункт «Верхний Чов» Республиканского центра образования, г. Сыктывкар, Республика Коми, Россия
(руководитель работы — Мазанова Ольга Юрьевна)

Шевчук Евгения Алексеевна,

ученица VIII класса, средняя школа № 206, г. Минск, Беларусь
(руководитель работы — Гаевская Вера Алексеевна)

Все представленные выше лауреаты (1-е место) и дипломанты (2-е место) конкурса будут награждены дипломами соответствующего достоинства от издательства «Образование и Информатика», а также получат печатный экземпляр журнала «Информатика в школе» № 3-2022 с итогами конкурса и опубликованной работой.

Уважаемые коллеги! Обращаем ваше внимание на то, что с 2022 года электронные версии журналов «Информатика и образование» и «Информатика в школе» размещаются на сайте издательства в режиме открытого доступа (Open Access), поэтому электронная подписка победителям не оформляется (как это было в предыдущих конкурсах).

Также жюри рекомендовало к публикации в журнале «Информатика в школе» работы следующих авторов:

Гербут Светлана Сергеевна,

учитель информатики, лицей № 2, г. Тула, Россия

Егорькова Ирина Александровна,

учитель истории и обществознания, лицей № 82, Нижний Новгород, Россия

Кошелева Людмила Александровна,

учитель информатики, лицей № 82, Нижний Новгород, Россия

Пешкова Елена Александровна,

учитель физики, лицей № 82, Нижний Новгород, Россия

Левушкина Арина Сергеевна,

учитель математики и информатики, средняя общеобразовательная школа № 31, г. Владимир, Россия

Паравина Алина Сергеевна,

учитель информатики, средняя общеобразовательная школа № 5 имени Героя Советского Союза В. Ф. Кравченко, г. Сызрань, Самарская область, 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: <http://www.infojournal.ru/>

DOI: 10.32517/2221-1993-2022-21-3-5-11

Н. А. Курганова

победитель Всероссийского конкурса «Школьная информатика: вчера, сегодня, завтра»

в номинации «Школьная информатика: вчера, сегодня, завтра»,

Омский государственный педагогический университет, г. Омск, Россия

ИСПОЛЬЗОВАНИЕ КВАДРОКОПТЕРОВ TELLO EDU ПРИ ИЗУЧЕНИИ РАЗДЕЛА «ОСНОВЫ АЛГОРИТМИЗАЦИИ» ШКОЛЬНОГО КУРСА ИНФОРМАТИКИ

Аннотация

В статье рассматриваются возможности использования квадрокоптеров Tello EDU при изучении раздела «Основы алгоритмизации» школьного курса информатики. Приводятся обоснования выбора квадрокоптеров в качестве средства для обучения школьников основам алгоритмизации. Перечисляются темы уроков, которые целесообразно выделить при изучении указанного раздела курса информатики с использованием квадрокоптеров. Представлены возможные этапы урока на тему «Алгоритмы и исполнители. Квадрокоптер — исполнитель алгоритмов». В описании этапов урока рассматриваются: основные понятия алгоритмизации, устройство квадрокоптера, особенности интерфейса приложения Tello для управления квадрокоптером со смартфона, техника безопасности по управлению квадрокоптером. Представлена построчная запись испытательного полета квадрокоптера Tello EDU. Рассмотрены примеры заданий, которые можно предложить учащимся на каждом этапе урока.

Ключевые слова: алгоритм, исполнитель, свойства алгоритмов, способы записи алгоритмов, примеры исполнителей, квадрокоптер, Tello EDU.

1. Введение

Одна из задач современного образования — поиск новых форм, методов и средств обучения, которые позволят учителю мотивировать учащихся на усвоение новых знаний.

Сегодня во многих школах открыты «Точки роста», которые имеют такое оборудование, как квадрокоптеры. Однако педагоги этих центров отмечают, что им не хватает методических рекомендаций, знаний того, каким образом квадрокоптеры можно использовать в урочной и внеурочной деятельности, в том числе на уроках информатики [5, 6].

Мы считаем, что квадрокоптеры, в частности квадрокоптеры Tello EDU, целесообразно использовать при изучении раздела «Основы алгоритмизации» школьного курса информатики. Среди причин выбора квадрокоптера в качестве средства обучения при преподавании основ алгоритмизации отметим следующие:

- Раздел «Основы алгоритмизации» является основополагающим для дальнейшего изучения различных языков программирования. Однако нередко учащиеся испытывают трудности при представлении результата формальной записи алгоритма. Написание программы для квадрокоптера, создание определенного сценария полета

Контактная информация

Курганова Наталья Александровна, канд. пед. наук, доцент кафедры прикладной информатики и математики, Омский государственный педагогический университет, г. Омск, Россия; *адрес:* 644099, Россия, г. Омск, наб. Тухачевского, д. 14; *e-mail:* kurganovana@gmail.com

N. A. Kurganova

Omsk State Pedagogical University, Omsk, Russia

USING TELLO EDU QUADROCOPTERS TO STUDY THE SECTION "BASICS OF ALGORITHMIZATION" OF A SCHOOL INFORMATICS COURSE

Abstract

The article discusses the possibilities of using Tello EDU quadcopters when studying the section "The basics of algorithmization" of a school informatics course. The substantiation of the choice of quadcopters as a means for teaching schoolchildren the basics of algorithmization is given. The topics of the lessons are listed, which it is advisable to highlight when studying the above-mentioned section of the informatics course using quadcopters. The possible stages of the lesson on the topic "Algorithms and executors. A quadcopter is an executor of algorithms" are described. The description of the stages of the lesson covers: the basic concepts of algorithmization, the structure of a quadcopter, the features of the Tello application interface for controlling a quadcopter from a smartphone, safety precautions for controlling a quadcopter. A line-by-line recording of the test flight of the Tello EDU quadcopter is presented. Examples of tasks that can be offered to students at each stage of the lesson are considered.

Keywords: algorithm, executor, properties of algorithms, methods of recording algorithms, examples of executors, quadcopter, Tello EDU.

на основе словесной записи алгоритма позволит при запуске квадрокоптера визуализировать исполняемые алгоритмы [9].

- Управление квадрокоптером позволяет формировать у учащихся надпрофессиональные умения и навыки, умение вести себя в условиях неопределенности [1].

2. Изучение раздела «Основы алгоритмизации» с использованием квадрокоптеров

Изучение раздела «Основы алгоритмизации» в школьном курсе информатики с использованием квадрокоптеров целесообразно разбить на **несколько уроков**:

1. Алгоритмы и исполнители. Квадрокоптер — исполнитель алгоритмов.
2. Линейные алгоритмы. Программирование автономного полета квадрокоптера в среде DroneBlocks.
3. Разветвляющиеся алгоритмы. Программирование автономного полета квадрокоптера в среде DroneBlocks.
4. Циклические алгоритмы. Программирование автономного полета квадрокоптера в среде DroneBlocks.

3. Содержание урока «Алгоритмы и исполнители. Квадрокоптер — исполнитель алгоритмов»

В рамках данной статьи рассмотрим основное содержание урока «Алгоритмы и исполнители. Квадрокоптер — исполнитель алгоритмов».

В ходе урока используется презентация, на слайдах которой отражены основные понятия, изучаемые на уроке [7].

Для записи решения заданий каждому учащемуся выдается рабочий лист [8].

Структура урока может быть представлена следующими этапами:

- I. Организационный момент.
- II. Знакомство с основными понятиями.
- III. Квадрокоптер Tello EDU — пример исполнителя.
- IV. Система команд исполнителя — система команд управления квадрокоптером Tello EDU.
- V. Словесное описание или построчная запись алгоритма — алгоритм управления квадрокоптером.
- VI. Домашнее задание. Краткий инструктаж по его выполнению.
- VII. Рефлексия «Термометр настроения».

Ход урока

I. Организационный момент

На рабочем листе на рисунке с термометром учащиеся фиксируют свое настроение перед началом урока (рис. 1). Прием помогает определить эмоциональное состояние учащихся на уровне ощущений.



Рис. 1. «Термометр настроения»

II. Знакомство с основными понятиями

Учитель при помощи слайдов презентации (рис. 2) знакомит учащихся с такими понятиями, как алгоритм, исполнитель алгоритма, свойства алгоритмов, способы записи алгоритмов, приводит примеры исполнителей [2].

III. Квадрокоптер Tello EDU — пример исполнителя

На этом этапе урока учитель дает общее определение квадрокоптера, затем рассказывает о квадрокоптере Tello EDU, акцентирует внимание учащихся на том, что квадрокоптер — это исполнитель.

Квадрокоптер — это беспилотный летательный аппарат с четырьмя пропеллерами, контролируемый на расстоянии с помощью пульта дистанционного управления или смартфона [3].

Tello EDU — это программируемый квадрокоптер, который может перемещаться на точные расстояния, используя внутренние датчики.

Данные определения демонстрируются на слайдах презентации.

Учитель демонстрирует квадрокоптер Tello EDU и рассказывает о его основных элементах [10]. На слайде презентации учащимся предлагается схема, в которой дублируется вся информация об упомянутых в рассказе педагога элементах квадрокоптера (рис. 3).

Далее учащимся предлагается задание по заполнению таблицы на рабочем листе.

Задание 1.

Заполните таблицу «Основные понятия» (табл. 1) применительно к квадрокоптеру.

Таблица 1

Исполнитель	
Круг решаемых задач	
Среда исполнителя	
Режим работы исполнителя	

Пример заполнения таблицы задания 1 учитель демонстрирует на слайде презентации (табл. 2). Учащиеся сверяют свои ответы с таблицей на слайде, исправляют их в случае необходимости.

Алгоритм

описание последовательности шагов в решении задачи, приводящих от исходных данных к требуемому результату.

Исполнитель

некоторый объект (человек, животное, техническое устройство), способный выполнять определённый набор команд.

Босова Л. Л. Информатика: учебник для 8 класса / Л. Л. Босова, А. Ю. Босова. М.: БИНОМ, Лаборатория знаний, 2013. 155 с.

2

Круг решаемых задач исполнителя

исполнитель создается для решения некоторого круга задач: выполнения вычислений, построения рисунков и т. д.

Среда исполнителя

область, обстановка, условия, в которых действует исполнитель.

Босова Л. Л. Информатика: учебник для 8 класса / Л. Л. Босова, А. Ю. Босова. М.: БИНОМ, Лаборатория знаний, 2013. 155 с.

3

Команда исполнителя

предписание исполнителю о выполнении отдельного законченного действия.

Система команд исполнителя

совокупность всех команд, которые могут быть выполнены некоторым исполнителем.

Босова Л. Л. Информатика: учебник для 8 класса / Л. Л. Босова, А. Ю. Босова. М.: БИНОМ, Лаборатория знаний, 2013. 155 с.

4

Режимы работы исполнителей

режим непосредственного управления и программного управления.

Режим непосредственного управления

исполнитель ожидает команд от человека и каждую команду немедленно выполняет.

Режим программного управления

исполнителю сначала задается полная последовательность команд (программа), а затем он выполняет все эти команды в автоматическом режиме.

Босова Л. Л. Информатика: учебник для 8 класса / Л. Л. Босова, А. Ю. Босова. М.: БИНОМ, Лаборатория знаний, 2013. 155 с.

5

Свойства алгоритма

- Дискретность
- Понятность
- Определенность
- Результативность
- Массовость

Босова Л. Л. Информатика: учебник для 8 класса / Л. Л. Босова, А. Ю. Босова. М.: БИНОМ, Лаборатория знаний, 2013. 155 с.

6

Способы записи алгоритма

- Словесное описание
- Построчная запись
- Блок-схема
- Школьный алгоритмический язык

Босова Л. Л. Информатика: учебник для 8 класса / Л. Л. Босова, А. Ю. Босова. М.: БИНОМ, Лаборатория знаний, 2013. 155 с.

7

Примеры исполнителей

- Черепашка
- Вычислитель

Босова Л. Л. Информатика: учебник для 8 класса / Л. Л. Босова, А. Ю. Босова. М.: БИНОМ, Лаборатория знаний, 2013. 155 с.

8

Рис. 2. Слайды презентации, представляющие основные понятия темы

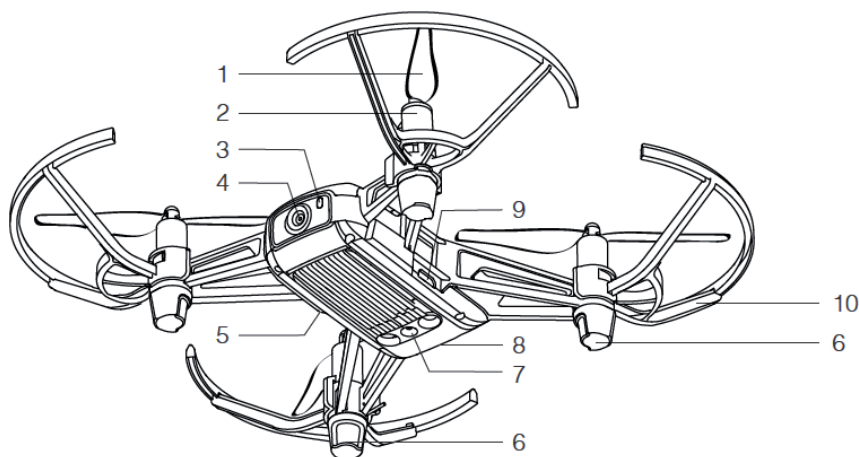


Рис. 3. Схема квадрокоптера [10]:

1 — пропеллеры; 2 — моторы; 3 — индикатор состояния квадрокоптера; 4 — камера; 5 — кнопка включения; 6 — антенны; 7 — система визуального позиционирования; 8 — полетная батарея; 9 — порт Micro USB; 10 — защита пропеллеров

Таблица 2

Исполнитель	Квадрокоптер Tello EDU
Круг решаемых задач	Перемещение на определенные расстояния, фотосъемка и видеосъемка различных объектов и т. п.
Среда исполнителя	Холл, спортивный зал, классная комната, оборудованная зона для полетов
Режим работы исполнителя	Ручной и программный

IV. Система команд исполнителя — система команд управления квадрокоптером Tello EDU

Учитель предлагает учащимся предположить, какой может быть система команд исполнителя — система команд по управлению квадрокоптером. Ученики называют команды на основе личного опыта, делают предположения.

Педагог рассказывает о мобильном приложении Tello, при помощи которого может осуществляться ручной режим работы исполнителя, управление квадрокоптером средствами смартфона. На слайде презентации демонстрируется скриншот интерфейса приложения Tello (рис. 4).

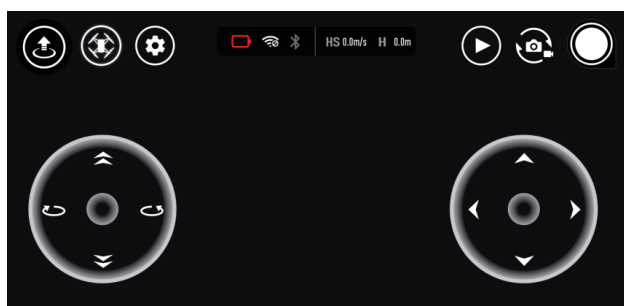


Рис. 4. Интерфейс приложения Tello

Учащимся предлагается на основе иконок интерфейса приложения Tello догадаться о командах по управлению квадрокоптером средствами смартфона и выполнить задание 2 на рабочем листе.

Задание 2.

Заполните таблицу «Система команд исполнителя» (табл. 3).

Таблица 3

Обозначение (иконка)	Команда

Окончание табл. 3

Обозначение (иконка)	Команда

После выполнения задания 2 учитель демонстрирует слайды презентации с правильными ответами (табл. 4). Учащиеся сравнивают ответы, представленные на слайдах, со своими вариантами, записанными на рабочем листе, и при необходимости исправляют свои ответы.

Таблица 4

Обозначение (иконка)	Команда
	Автоматический взлет
	Автоматическая посадка
	Съемка фотографий или запись видео
	Перемещение левого стика вверх и вниз меняет высоту полета квадрокоптера
	Перемещение левого стика влево и вправо управляет вращением квадрокоптера
	Перемещение правого стика вверх и вниз передвигает квадрокоптер вперед и назад
	Перемещение правого стика влево и вправо наклоняет квадрокоптер влево или вправо (крен)

Затем учитель с помощью слайдов презентации знакомит учащихся с остальными элементами интерфейса (рис. 5).

Элементы интерфейса приложения Tello	
	Интеллектуальные режимы полета
	Настройки (скорость полета, VR, джойстик Bluetooth, Wi-Fi)
	Уровень заряда батареи

TELLO. Руководство пользователя, <https://cornu.ru/manuals/gz900.pdf>

18

Элементы интерфейса приложения Tello	
	Состояние Wi-Fi
	Отображает состояние Bluetooth
	Скорость полета

TELLO. Руководство пользователя, <https://cornu.ru/manuals/gz900.pdf>

19

Элементы интерфейса приложения Tello	
	Высота полета
	Воспроизведение и просмотр фотографий
	Переключение Фото / Видео

TELLO. Руководство пользователя, <https://cornu.ru/manuals/gz900.pdf>

20

Рис. 5. Слайды презентации, представляющие элементы интерфейса приложения Tello

V. Словесное описание или построчная запись алгоритма — алгоритм управления квадрокоптером

Правила техники безопасности при управлении квадрокоптером.

Прежде чем начать знакомство с управлением квадрокоптером, с учащимися необходимо обсудить правила техники безопасности [4].

- Обозначьте зону полетов, за которую нельзя заходить, при помощи ленты на полу.
- Осуществляйте полет квадрокоптера только в обозначенной зоне и не допускайте вылет за ее пределы.
- Выполняйте все указания учителя.
- Проверьте, что вращению пропеллеров ничего не мешает.
- Проверьте, что все присутствующие находятся за спиной оператора управления квадрокоптером.
- Не дотрагивайтесь до движущихся частей моторов и винтов.
- Оставайтесь на расстоянии 3 м от квадрокоптера.
- Осуществляйте взлет с пола или с земли только с ровной площадки.

- Обязательно сохраняйте зрительный контакт с квадрокоптером.
- Помните примерное время полета квадрокоптера.
- Остановите полет, если батарея уже близка к разрядке, иначе это может привести к неконтролируемому падению квадрокоптера.
- Отключите питание после приземления квадрокоптера.

Памятку по технике безопасности при управлении квадрокоптером целесообразно оформить в виде плаката.

Рекомендуется осуществлять управление квадрокоптером или в холле, или в спортивном зале, в классе — в обозначенной лентой зоне полета.

Управление квадрокоптером со смартфона при помощи приложения Tello.

Для того чтобы начать управлять квадрокоптером со смартфона при помощи приложения Tello на основе словесного описания или построчной записи алгоритма, необходимо выполнить следующее:

1. Скачать и установить приложение Tello на смартфон.
2. Нажать кнопку включения на квадрокоптере.
3. Включить Wi-Fi на смартфоне и выбрать сеть Tello-XXXXXX.

Данные рекомендации для учащихся выводятся на слайде презентации.

Учитель выполняет испытательный полет и демонстрирует ученикам основы управления квадрокоптером.

На основе рекомендаций учителя, размещенных на слайде, несколько учеников поочередно выполняют задание 3 — испытательный полет в обозначенной зоне [10], повторяя все шаги построчной записи алгоритма, размещенной на рабочем листе.

Задание 3.

Выполните испытательный полет по предложенному сценарию [10]:

1. Поместите квадрокоптер на плоскую поверхность с батареей полета, направленной на себя.
2. Включите квадрокоптер.
3. Запустите приложение Tello и войдите в режим просмотра камеры.
4. Выберите автоматический взлет.
5. Используя виртуальные джойстики для управления квадрокоптером, поднимите квадрокоптер вверх примерно на 1 м.
6. Выберите автоматическую посадку.
7. Выключите квадрокоптер.

Далее ученикам предлагается самостоятельно выполнить еще два задания.

Задание 4.

Создание собственного сценария полета:

1. Составьте сценарий полета квадрокоптера в обозначенной зоне: в результате полета квадрокоптер должен схематично нарисовать в воздухе прямоугольник. Сделайте на рабочем листе построчную запись алгоритма.
2. С разрешения учителя реализуйте ваш сценарий полета в обозначенной зоне.

Задание 5.

Полетная миссия квадрокоптера «Сопровождение и видеосъемка велосипедных гонок по заданной трассе»:

1. Составьте примерный сценарий полета квадрокоптера над трассой велосипедных гонок. Сценарий должен учитывать сопровождение участников гонки и видеосъемку. Начало составления сценария — место старта. Сделайте на рабочем листе построчную запись алгоритма.
2. Представьте примерный сценарий полета при помощи значков интерфейса приложения Tello.
3. С разрешения учителя реализуйте ваш сценарий полета в обозначенной зоне на распечатанном поле трассы на основе составленной схемы (рис. 6). Адаптируйте алгоритм к реальным условиям.

VI. Домашнее задание. Краткий инструктаж по его выполнению

В качестве домашней работы учащимся предлагаются два задания.

Первое задание — на запоминание основных элементов квадрокоптера — предлагается на электронной доске Padlet (рис. 7). Учитель публикует на доске пост, в котором представлены схема квадрокоптера и формулировка задания для учащихся. В настройках доски педагог отмечает, что публикуемые учениками ответы будут оцениваться по пятибалльной шкале, а над каждым постом будет отображаться имя автора ответа. Отметим, что пост учителя со схемой можно открыть, тем самым увеличив масштаб изображения.

Задание 6.

Запишите обозначенные цифрами элементы устройства на электронной доске.

Ученик нажимает кнопку «+» и добавляет пост с ответом на доску.



Рис. 6. Схема трассы

Домашнее задание представлено на рабочем листе в виде QR-кода — ссылки на электронную доску (рис. 8).

Второе задание предполагает изучение учащимися технических характеристик квадрокоптера Tello EDU.

Задание 7.

Заполните таблицу «Основные характеристики квадрокоптера Tello EDU» (табл. 5) на основании руководства пользователя TELLO:

<https://cornu.ru/manuals/gyz900.pdf> [10].



Рис. 7. Пример домашнего задания на электронной доске Padlet



Рис. 8. QR-код — ссылка на электронную доску

Таблица 5

Характеристика	Описание
Вес	
Примерное время полета	
Режимы полета	
Интеллектуальные режимы полета	1. 2. 3. 4. 5. 6.
Камера	

Задание также можно найти на рабочем листе [8].
Пример заполнения представлен в таблице 6.

Таблица 6

Характеристика	Описание
Вес	87 г
Примерное время полета	Примерно 13 минут
Режимы полета	Быстрый и медленный
Интеллектуальные режимы полета	1. Отскок 2. Флипы 8D 3. Бросить и пойти 4. 360° 5. Круг 6. Вверх и отлет
Камера	5 мегапикселей

VII. Рефлексия «Термометр настроения»

На рисунке с термометром на рабочем листе учащиеся фиксируют свое настроение по окончании урока.

4. Заключение

Опыт проведения занятий с учениками восьмого класса показывает, что использование квадрокоптеров на уроке информатики при изучении темы «Алгоритмы и исполнители. Квадрокоптер — исполнитель алгоритмов» дает возможность учащимся перейти от теоретической деятельности к практико-ориентированной, визуализировать изучаемый материал, позволяет начать формировать у учащихся надпрофессиональные умения и навыки.

Список источников

1. Атлас новых профессий. <https://new.atlas100.ru/>
2. Босова Л. Л., Босова А. Ю. Информатика. 8 класс: учебник. ФГОС. М.: БИНОМ. Лаборатория знаний, 2020. 176 с.
3. Все о квадрокоптерах: применение, классификация, производители // Цифровой квадрат. <https://digitalsquare.ru/stati/vse-o-kvadrokopterah.html>
4. Инструкция по технике безопасности для обучающихся по направлению «Аэроквантум». <https://kvantorium54.ru/company/programi/Инструкция%20ТБ%20Аэроквантум.pdf>
5. Квадрокоптеры для школы // Комплектант. Комплектация образовательных учреждений. <https://fgoskomplekt.ru/blog/kvadrokoptery-dlya-shkoly/>
6. Квадрокоптеры для школы: что такое, зачем нужны, как выбрать // Новация. Комплексное оснащение школ и детских садов. https://n-72.ru/company/news/kvadrokoptery_dlya_shkoly_chno_takoe_zachem_nuzhny_kak_vybrat/
7. Курганова Н. А. Презентация к уроку «Алгоритмы и исполнители». https://docs.google.com/presentation/d/1ib-xDixIG7XavYMkNTsWzyRY_2h5TPcSDUsk4-Yfy9Y/edit?usp=sharing
8. Курганова Н. А. Рабочий лист к уроку «Алгоритмы и исполнители». <https://docs.google.com/document/d/1KpMtBXWiqC3qjkn7qKg963jUqEPp4xhwfwC2wpLWQo/edit?usp=sharing>
9. Первый опыт программирования DJI Tello // PRO-TELLO. <http://protello.com/tello-programming-1>
10. TELLO. Руководство пользователя. <https://cornu.ru/manuals/gyz900.pdf>

DOI: 10.32517/2221-1993-2022-21-3-12-12

Айзат Шакуров

победитель Всероссийского конкурса «Школьная информатика: вчера, сегодня, завтра» в номинации «Школьная информатика: заглянем в будущее», лицей-интернат № 24, Дворец творчества детей и молодежи имени Ильдуса Харисовича Садыкова, г. Нижнекамск, Республика Татарстан, Россия

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ШКОЛЕ БУДУЩЕГО

В XXI веке технологии развиваются с неимоверной скоростью. Сегодня практически в каждой квартире, в каждом доме есть разнообразная компьютеризированная техника.

Современные достижения технологического прогресса находят отражение и в школьном образовании. Например, в лицее № 24 Нижнекамска, где я учусь, ребята активно занимаются робототехникой и программированием.

А как же дальше будут развиваться технологии? И пойдет ли это развитие на пользу нам — ученикам школ? Рассмотрим варианты.

Мне кажется, что технологический прогресс в нашей стране ничуть не сбавит свою скорость, несмотря на сегодняшнее столь большое внешнее давление на Российскую Федерацию. И развитие технологий будет все больше и больше влиять на обучение в школе.

Возможно, вместо пяти-шести учебников школьники станут приносить на уроки один-единственный предмет — ноутбук, который каждому ученику будут выдавать бесплатно — школы и другие образовательные организации будут покупать ноутбуки на средства, выделяемые государством.

Применение на уроках новейших технологий изменит методы изучения учебного материала.

Так, использование очков виртуальной реальности позволит на уроках истории увидеть картины сражений или, например, «посетить» Древнюю Грецию.

На уроках географии благодаря современным технологиям можно будет «почти вживую» попутешествовать по всей нашей планете — наглядно познакомиться даже с самыми труднодоступными уголками Земли, подробно рассмотреть достопримечательности разных стран.

На геометрии ученики будут строить чертежи сразу на ноутбуке.

Для создания макетов различных построек и сооружений на уроках технологии будут использоваться программы проектирования, 3D-принтеры, а вся работа будет больше похожа на игру — как сборка из кубиков LEGO.

На уроках информатики учащиеся школ будущего смогут создавать собственные программы и игры для гаджетов.

Уроки изобразительного искусства будут проходить с использованием графических планшетов, что сэкономит бумагу и краски, а процесс рисования и сами рисунки сделает более интересными.

Хотелось бы модернизировать учебную мебель, оснастив ее датчиками: с их помощью можно будет настраивать высоту и наклон стола и стула в зависимости от физических данных каждого ученика.

Вопрос с меню и температурой еды в столовых можно решить с помощью программ, управляющих автоматами, которые будут предлагать каждому ученику школы блюда с учетом его предпочтений в еде, сохраненных в компьютерной базе.

Многие уроки будут проходить в формате веб-квестов. Голограммы в школьных коридорах позволят проводить виртуальные эксперименты. Интернет-ассистенты на мониторах подскажут ответ на любой вопрос. Среди учителей появится новая специальность — игропедагог.

На первый взгляд все это кажется прекрасным. И, безусловно, должно пойти на пользу ученикам. Но насколько использование различных технологий безопасно для нас? На этот вопрос невозможно дать однозначный ответ. Найдутся такие ученики, которые будут слишком много времени проводить за компьютером, лишая себя обычных житейских радостей общения со сверстниками и портя свое здоровье.

Однако можно предположить, что при использовании новейших технологий школьникам будет гораздо интереснее учиться, а значит, учебная программа по всем предметам будет изучаться ими глубже. А это увеличит процент образованных людей, что полезно скажется на развитии нашей страны.

Так что, пожалуй, технологический прогресс в школе — это прекрасно. И, возможно, все, что я изложил в своей версии развития информационных технологий в школьном образовании, придет к среднестатистическому ученику совсем скоро, а не в каком-то отдаленном будущем. Я надеюсь, что применение информационных технологий в школах будет только развиваться и приносить всем пользу.

DOI: 10.32517/2221-1993-2022-21-3-13-26

Н. В. Климина

дипломант Всероссийского конкурса «Школьная информатика: вчера, сегодня, завтра» в номинации «Школьная информатика: вчера, сегодня, завтра», структурное подразделение, реализующее дополнительные общеобразовательные программы, «Дворец творчества детей и молодежи» средней общеобразовательной школы № 14 «Центр образования» имени кавалера ордена Ленина Н. Ф. ШUTOVA, г. Сызрань, Самарская область, Россия

ВИРТУАЛЬНАЯ РОБОТОТЕХНИКА НА БАЗЕ РОБОСИМУЛЯТОРОВ VEXCODE VR И OPEN ROBERTA LAB*

Аннотация

В статье обосновывается выбор робосимуляторов VEXcode VR и Open Roberta Lab для обучения робототехнике в школе. Приведено описание некоторых задач, решаемых виртуальными роботами в указанных средах, представлены возможные алгоритмы решения задач. Методические разработки в виде мини-задач будут полезны учителям, преподающим робототехнику в школе, для проведения занятий с использованием нескольких виртуальных роботов в робосимуляторах VEXcode VR и Open Roberta Lab со II по VII класс. Знакомство с элементарными понятиями теории автоматического управления на базе представленной в статье методики будет способствовать формированию у обучающихся навыков практического применения программирования, умения применять информационные технологии для решения задач управления робототехническими системами.

Ключевые слова: робототехника, робосимулятор, конфигурация робота, датчики, фон симуляции, игровая площадка, симуляция.

1. Введение

Робототехника — актуальное современное направление в технике и технологиях, которое в настоящее время активно внедряется в образование и получает все большее развитие в рамках школьных курсов информатики и технологии, а также в сфере дополнительного образования детей. Образовательная робототехника привлекательна для школьников практически значимыми заданиями, которые позволяют наглядно увидеть результаты программирования в виде поведения роботов и благодаря этому

повысить интерес школьников к изучаемым техническим дисциплинам. Программирование роботов знакомит обучающихся с элементарными понятиями теории автоматического управления и развивает инженерное, вычислительное, креативное и критическое мышление [1, 6].

С одной стороны, сегодня далеко не каждая российская школа обладает робототехническим оборудованием. С другой стороны, у учеников есть желание изучать робототехнику, а у педагогов — знакомить с ней учащихся. В такой ситуации хорошим решением будет использование на занятиях робосимуляторов.

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_03-2022/

Контактная информация

Климина Наталья Владимировна, педагог дополнительного образования, структурное подразделение, реализующее дополнительные общеобразовательные программы, «Дворец творчества детей и молодежи» средней общеобразовательной школы № 14 «Центр образования» имени кавалера ордена Ленина Н. Ф. ШUTOVA, г. Сызрань, Самарская область, Россия; *адрес:* 446001, Россия, Самарская область, г. Сызрань, ул. Ульяновская, д. 145; *e-mail:* natklimina@yandex.ru

N. V. Klimina

Palace of Creativity for Children and Youth of School 14 "Center of Education" named after the Knight of the Order of Lenin N. F. Shutov, Syzran, Samara Region, Russia

VIRTUAL ROBOTICS BASED ON VEXCODE VR AND OPEN ROBERTA LAB ROBOT SIMULATORS

Abstract

The article substantiates the choice of VEXcode VR and Open Roberta Lab robot simulators for teaching robotics at school. A description of some problems solved by virtual robots in these environments is given, and possible algorithms for solving problems are presented. Methods of teaching presented in the form of mini tasks will be useful for teachers teaching robotics at school to conduct classes using several virtual robots in VEXcode VR and Open Roberta Lab robot simulators from 2nd to 7th grade. Acquaintance with the elementary concepts of the theory of automatic control on the basis of the methodology presented in the article will contribute to the formation of students' skills in the practical application of programming, the ability to apply information technologies to solve problems of control of robotic systems.

Keywords: robotics, robot simulator, robot configuration, sensors, simulation background, playground, simulation.

Применение робосимуляторов позволяет решить следующие задачи:

- обучение школьников современной робототехнике с использованием виртуальных роботов;
- знакомство учащихся с принципами работы нескольких виртуальных роботов;
- дистанционное обучение для школьников и выполнение заданий дистанционных конкурсов из дома;
- использование дополнительных полей для роботов, в том числе авторских полей.

К недостаткам робосимуляторов можно отнести ограниченную конфигурацию роботов и идеальные модельные условия среды действия. Однако, несмотря

на эти недостатки, дидактические возможности робосимуляторов очень высоки и позволяют достичь различных целей обучения [3, 5, 6].

Одними из современных сред программирования виртуальных роботов являются онлайн-платформы VEXcode VR и Open Roberta Lab, которые мы рассмотрим в данной статье.

2. Особенности робосимулятора VEXcode VR

Облачный робосимулятор VEXcode VR (<https://vr.vex.com>) — бесплатная русифицированная среда блочного и текстового программирования на базе языков

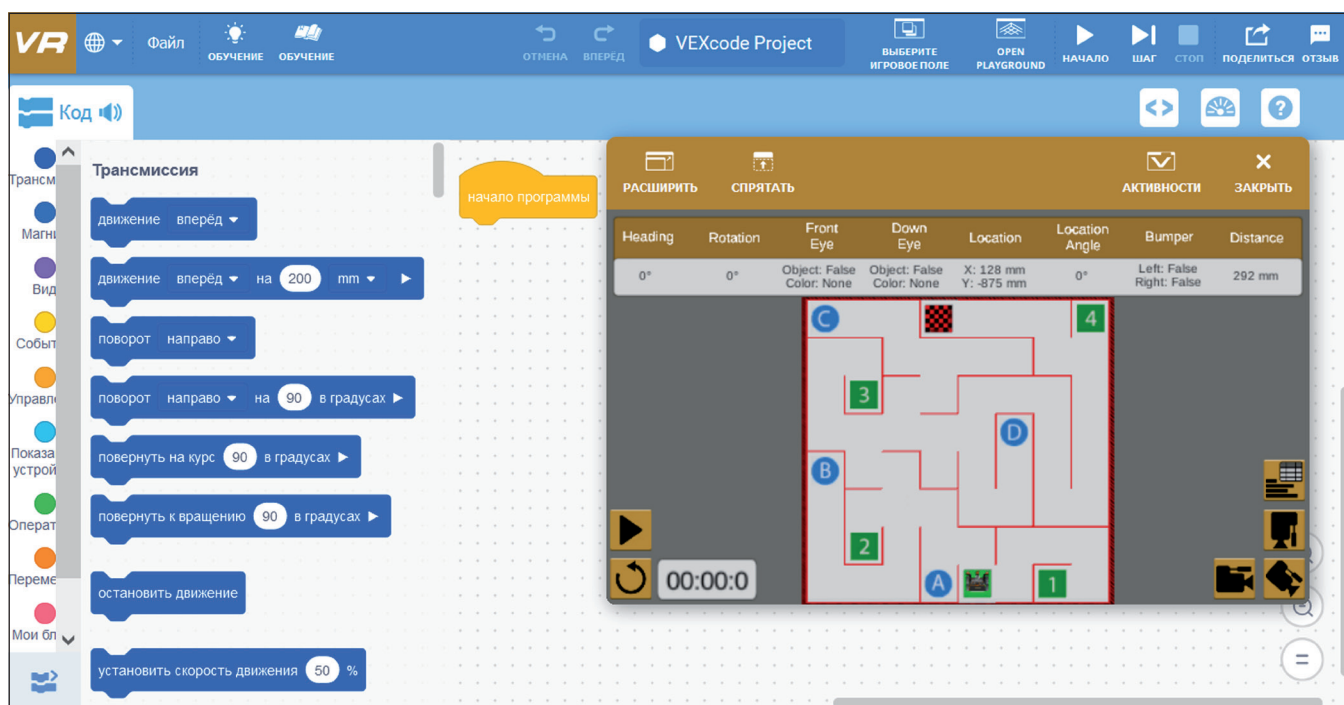


Рис. 1. Интерфейс среды симуляции VEXcode VR

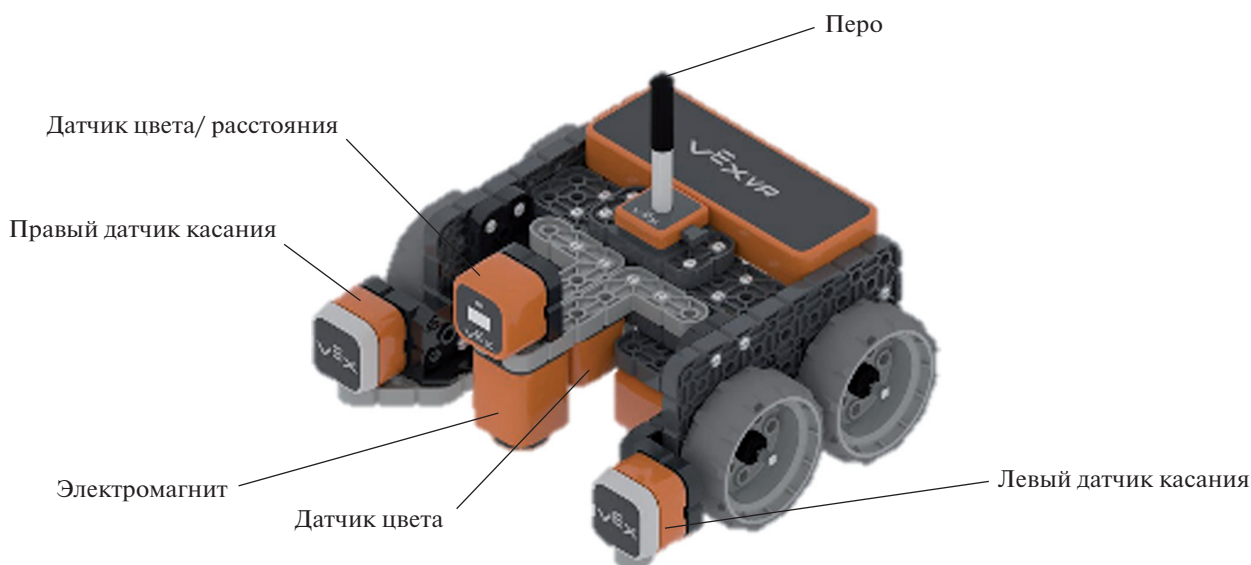


Рис. 2. Конфигурация виртуального робота VEX

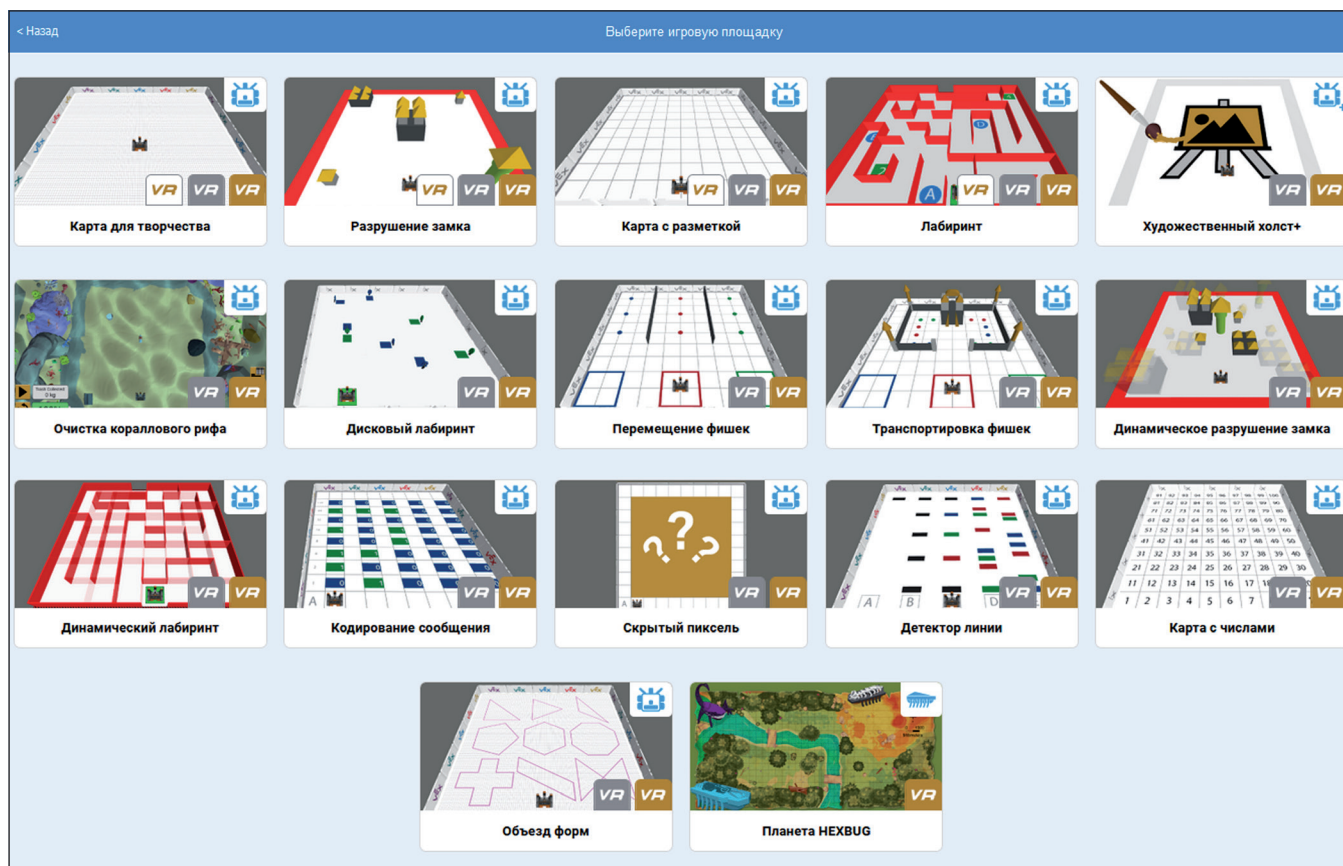


Рис. 3. Игровые площадки робосимулятора VEXcode VR

программирования Scratch Blocks и Python [10]. Поддерживается большинством современных браузеров (рис. 1).

Робосимулятор предоставляет удобный интерфейс для программирования поведения виртуального трехмерного робота VEX на множестве встроенных локаций — *игровых площадок*. Понятный язык программирования, интересные локации и трехмерный робот позволяют использовать данный робосимулятор уже в начальной школе [2, 3, 6, 9, 10].

Конфигурация виртуального трехмерного робота VEX представлена на рисунке 2.

Игровые площадки робосимулятора (рис. 3) хорошо подходят для организации занятий в виде игры, миссии, мини-соревнований.

Программа для робота представляет собой набор команд из блоков (рис. 4). Эти блоки выбираются из библиотеки команд, расположенной на экране симулятора слева от области кода (см. рис. 1).

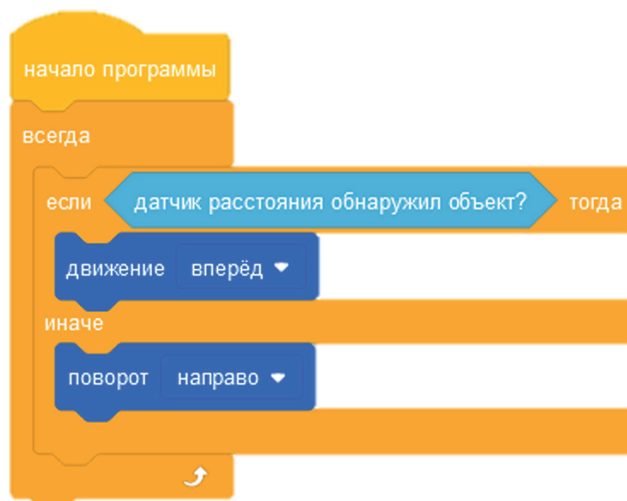


Рис. 4. Программа для робота VEX

3. Примеры задач для решения в робосимуляторе VEXcode VR по теме «Сенсоры»

Задача 1. Игровая площадка «Очистка кораллового рифа».

Условие задачи.

Написать программу для робота — сборщика мусора кораллового рифа. Робот собирает мусор, когда его коснется. Время работы робота ограничено зарядом его батареи. Оптимизировать программу, чтобы было собрано как можно больше мусора.

Решение задачи представлено на рисунке 5.

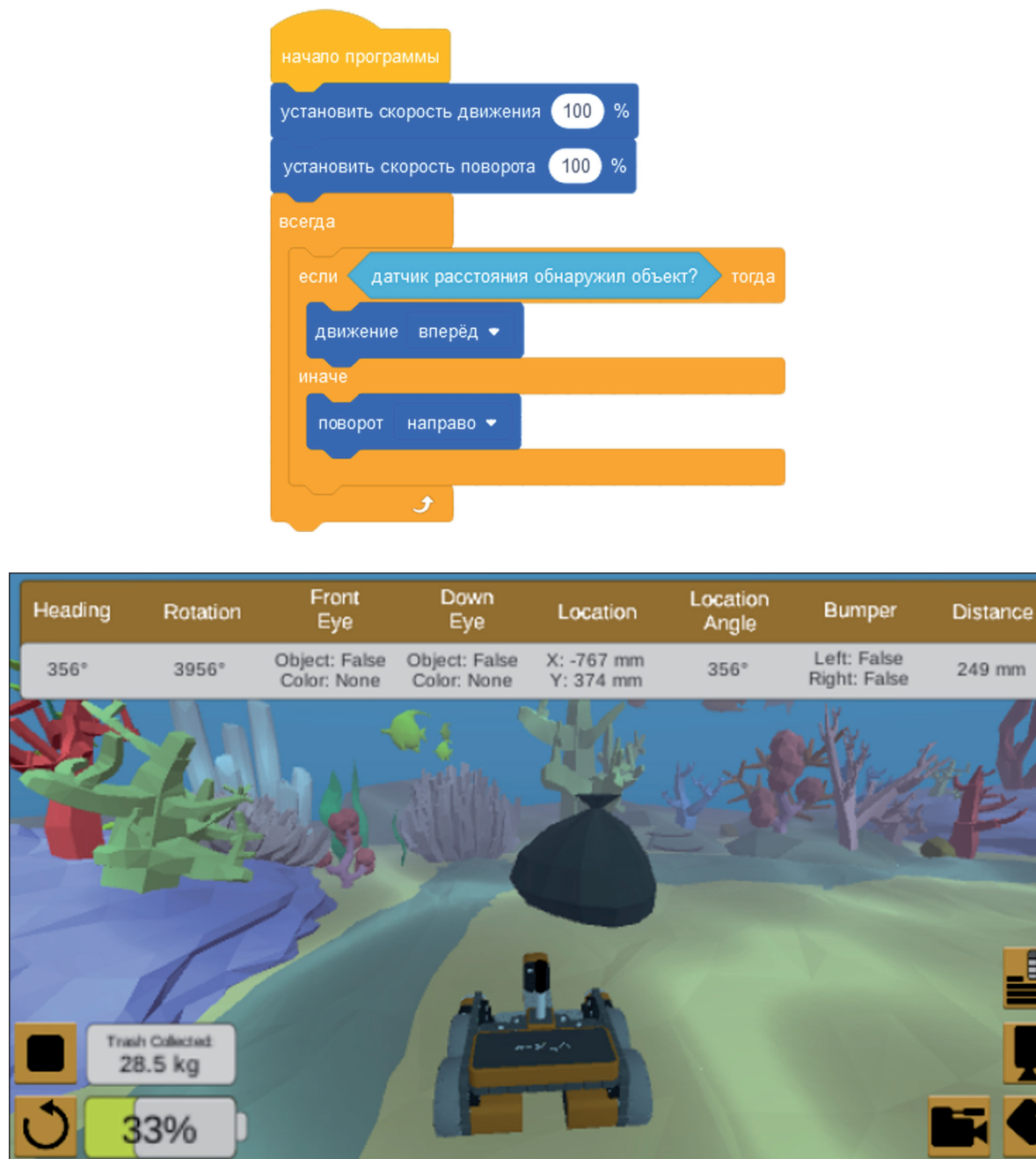


Рис. 5. Решение задачи 1

Задача 2. Игровая площадка «Динамическое разрушение замка».

Условие задачи.

Написать универсальную программу для разрушения построек на поле и очистки поля. При выезде за красную линию робот падает со стола. При перезапуске программы расположение построек меняется. Программа должна работать на любом выдаваемом варианте расположения построек.

Решение задачи представлено на рисунке 6.

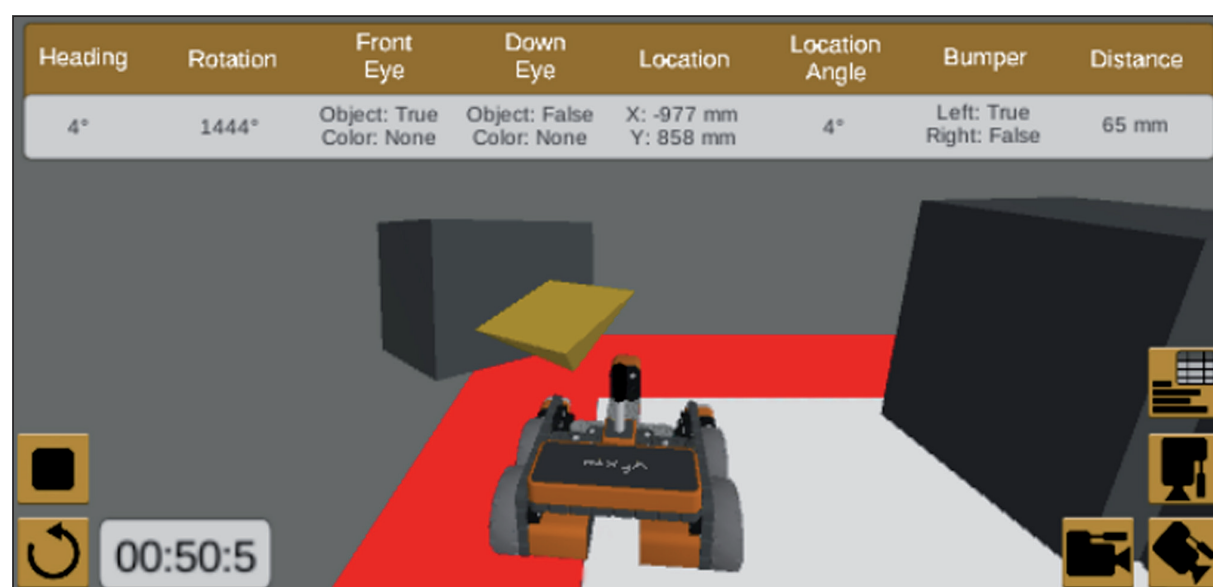


Рис. 6. Решение задачи 2

Задача 3. Игровая площадка «Детектор линий».

Условие задачи.

Написать программу, которая позволяет роботу проехать по ряду Е, останавливаясь на красных линиях.

Решение задачи представлено на рисунке 7.

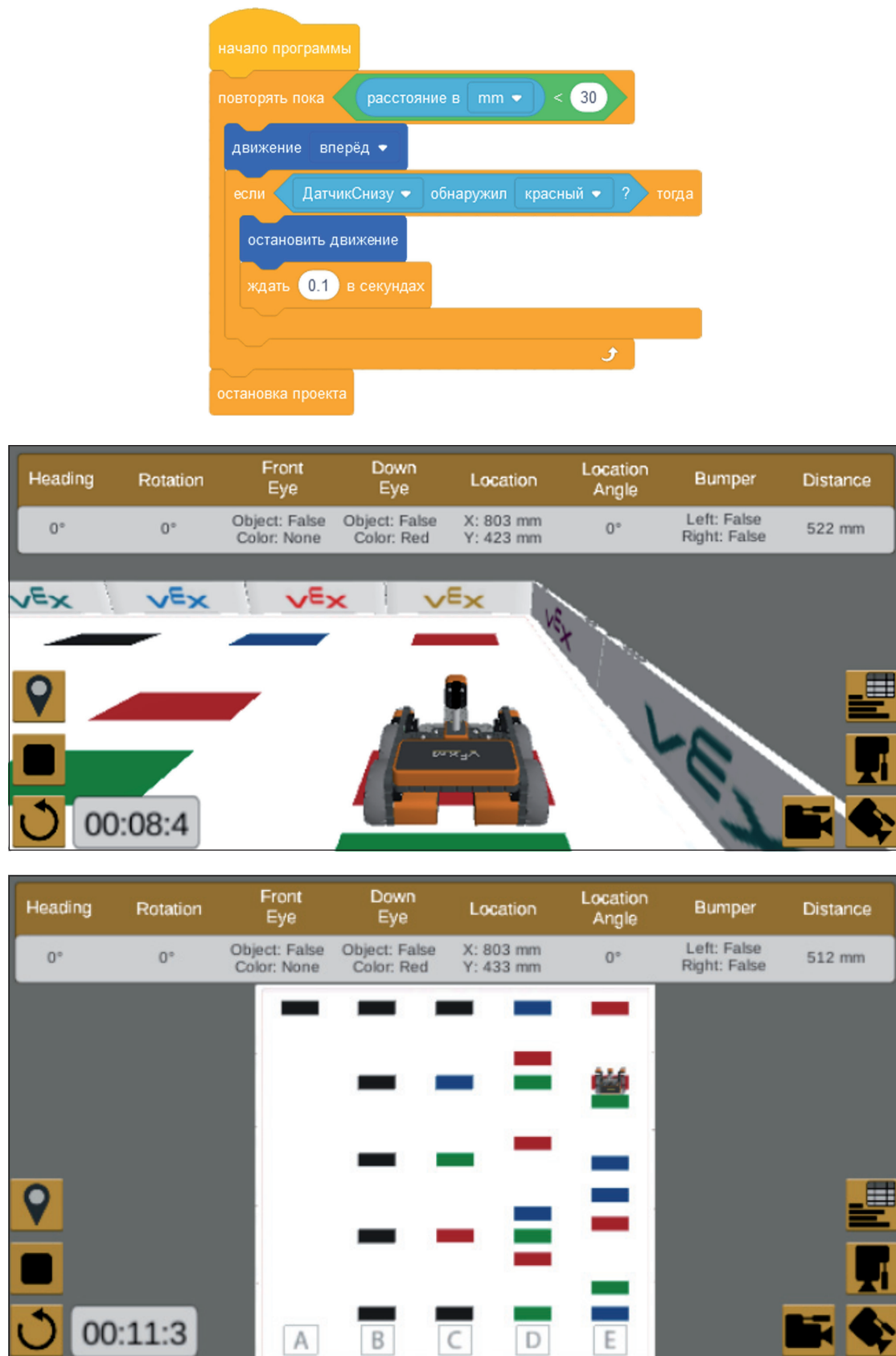


Рис. 7. Решение задачи 3

Задача 4. Игровая площадка «Динамический лабиринт».

Условие задачи.

Написать программу, позволяющую роботу выбраться из лабиринта. При перезапуске программы лабиринт меняется. Программа должна работать на всех выдаваемых вариантах лабиринтов.

Решение задачи представлено на рисунке 8.

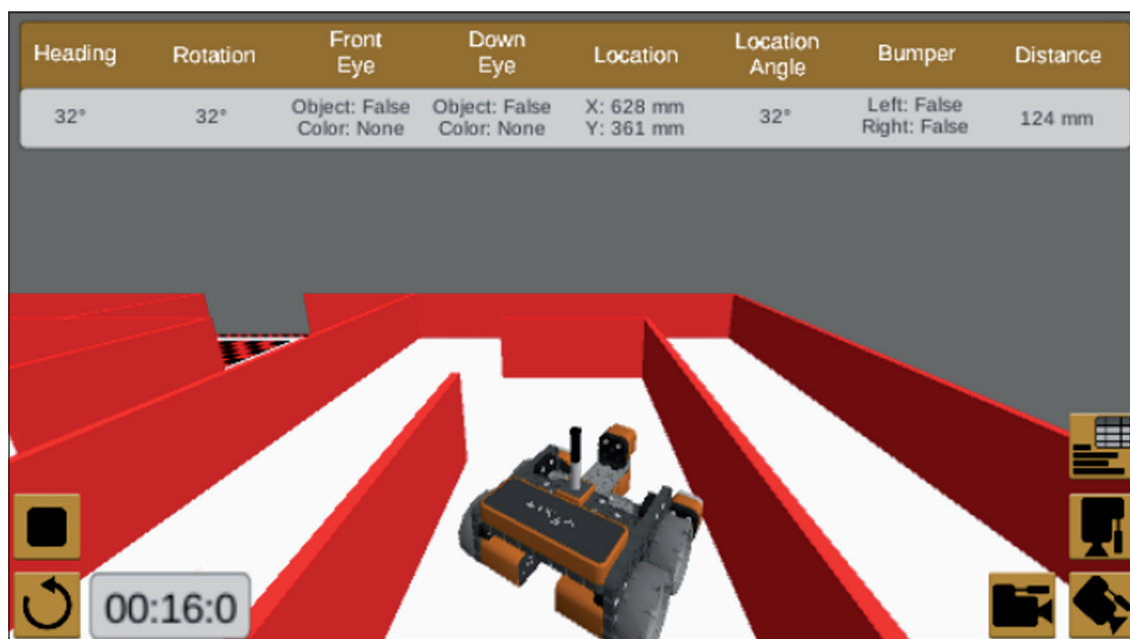


Рис. 8. Решение задачи 4

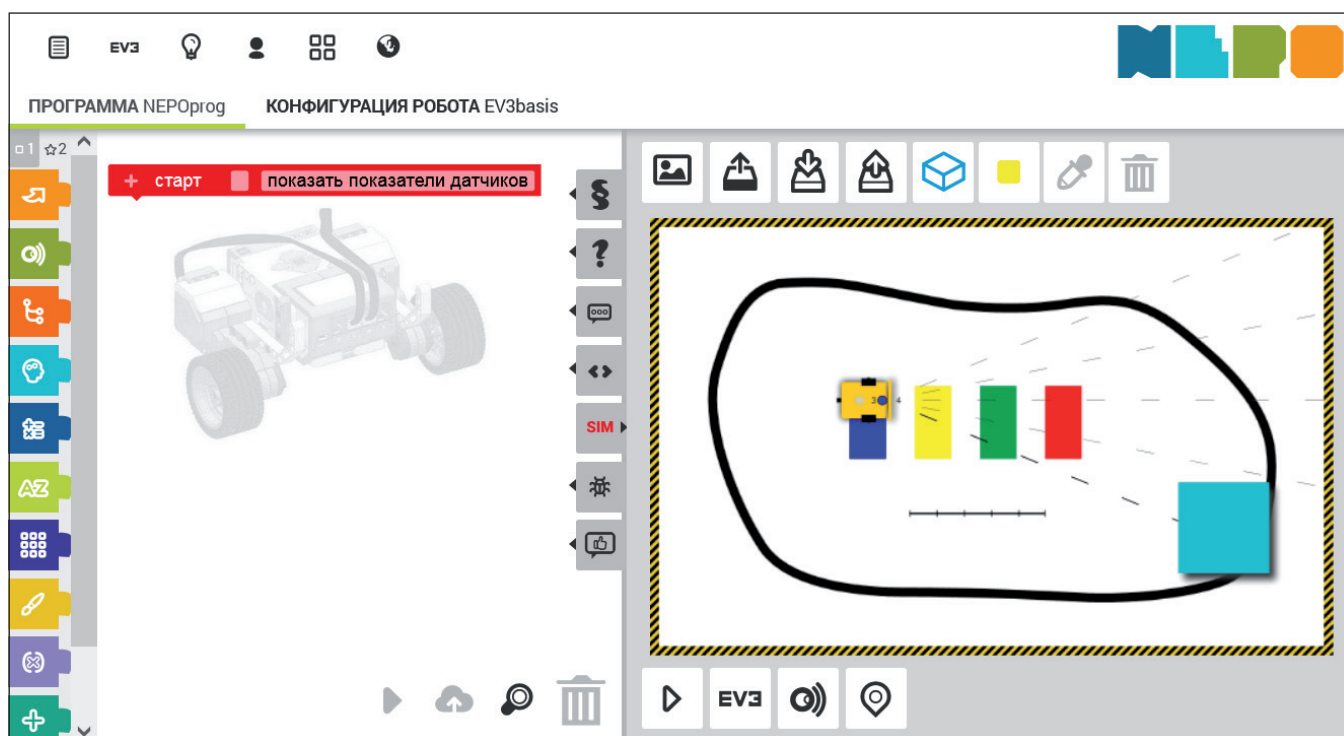


Рис. 9. Интерфейс платформы Open Roberta Lab

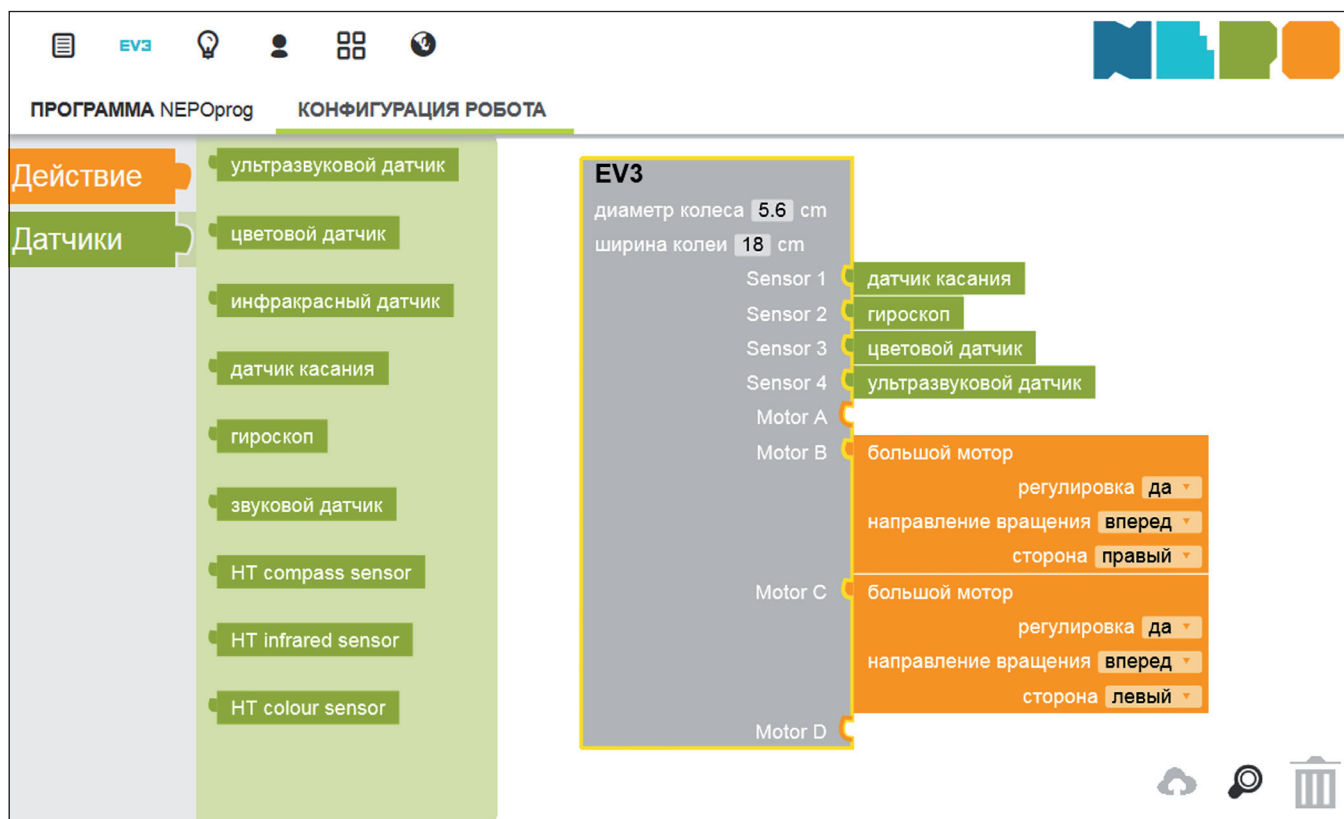


Рис. 10. Конфигурация виртуального робота Mindstorms EV3

4. Особенности робосимулятора Open Roberta Lab

Облачный робосимулятор Open Roberta Lab (<https://lab.open-roberta.org>) — бесплатная русифицированная платформа блочного программирования большого количества устройств на базе языка программирования NEPO. Поддерживается большинством современных браузеров (рис. 9). Это проект в рамках немецкой образовательной инициативы «Roberta — Обучение с роботами», инициированной Институтом интегральных схем общества Фраунгофера (нем. Fraunhofer-Institut für Integrierte Schaltungen, Fraunhofer IIS, FIIS), который является институтом, принадлежащим Фраунгоферовскому обществу [1].

Робосимулятор предоставляет удобный интерфейс программирования поведения нескольких виртуальных

устройств с возможностью симуляции на нескольких встроенных локациях — *фонах симуляции*. Понятный язык программирования, большая библиотека виртуальных роботов и возможность загружать собственные фоны симуляции позволяют использовать данный робосимулятор как в начальной школе, так и в среднем звене при проведении занятий по робототехнике [1, 7, 8].

Конфигурацию каждого виртуального робота можно подстраивать под цели и задачи конкретного занятия — снимать или добавлять датчики, моторы (рис. 10).

Программа для робота представляет собой набор команд из блоков (рис. 11). Эти блоки выбираются из библиотеки команд, расположенной на экране симулятора слева от области кода.

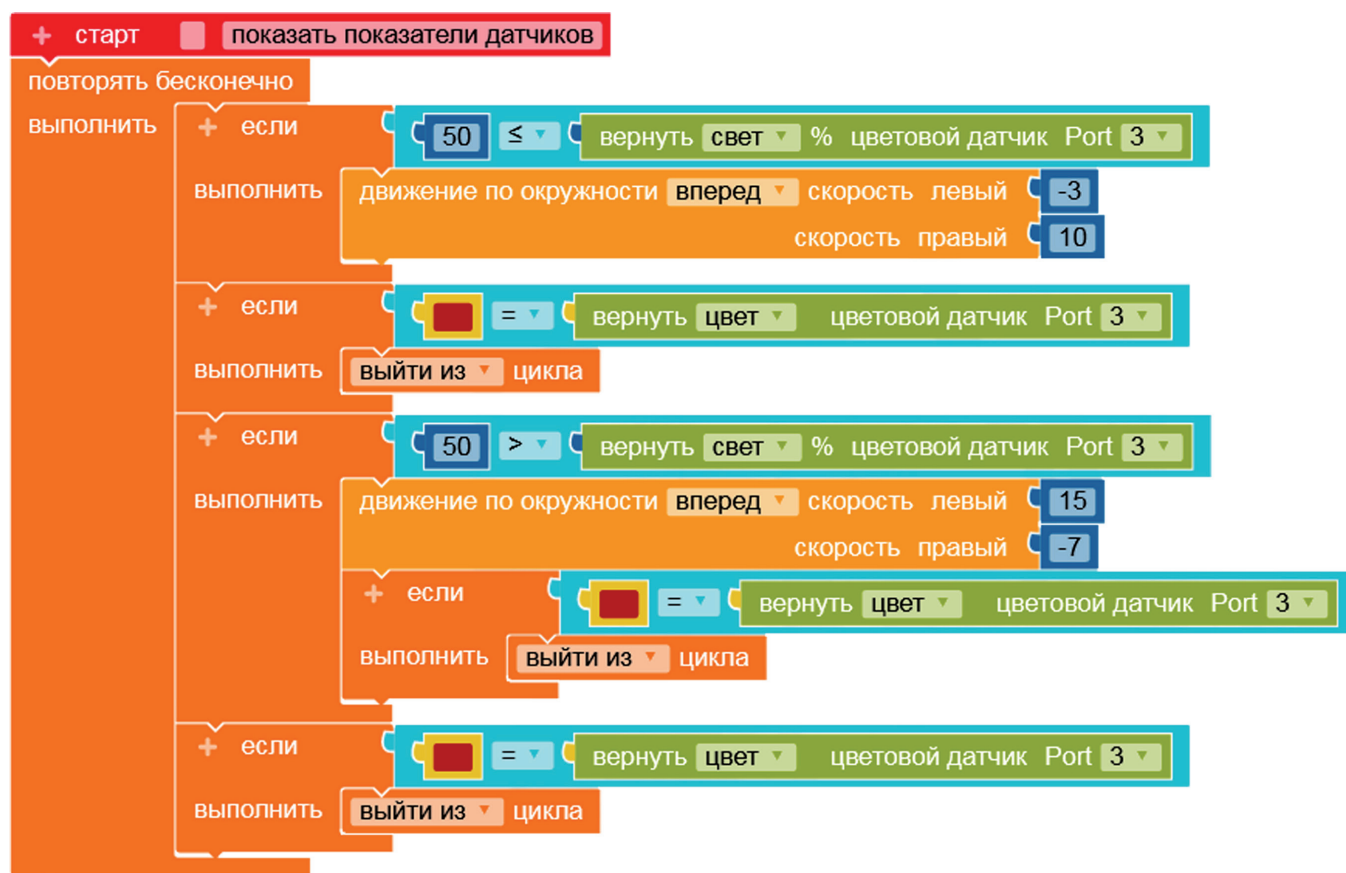


Рис. 11. Пример программы для виртуального робота Mindstorms EV3

5. Примеры задач для решения в робосимуляторе Open Roberta Lab по теме «Сенсоры»

Задача 5. Mindstorms EV3. Следование по линии с двумя датчиками цвета.

Условие задачи.

Подключить к роботу два датчика цвета. Написать программу движения по замкнутой линии с использованием двух датчиков цвета (рис. 12).

Решение задачи представлено на рисунке 12.

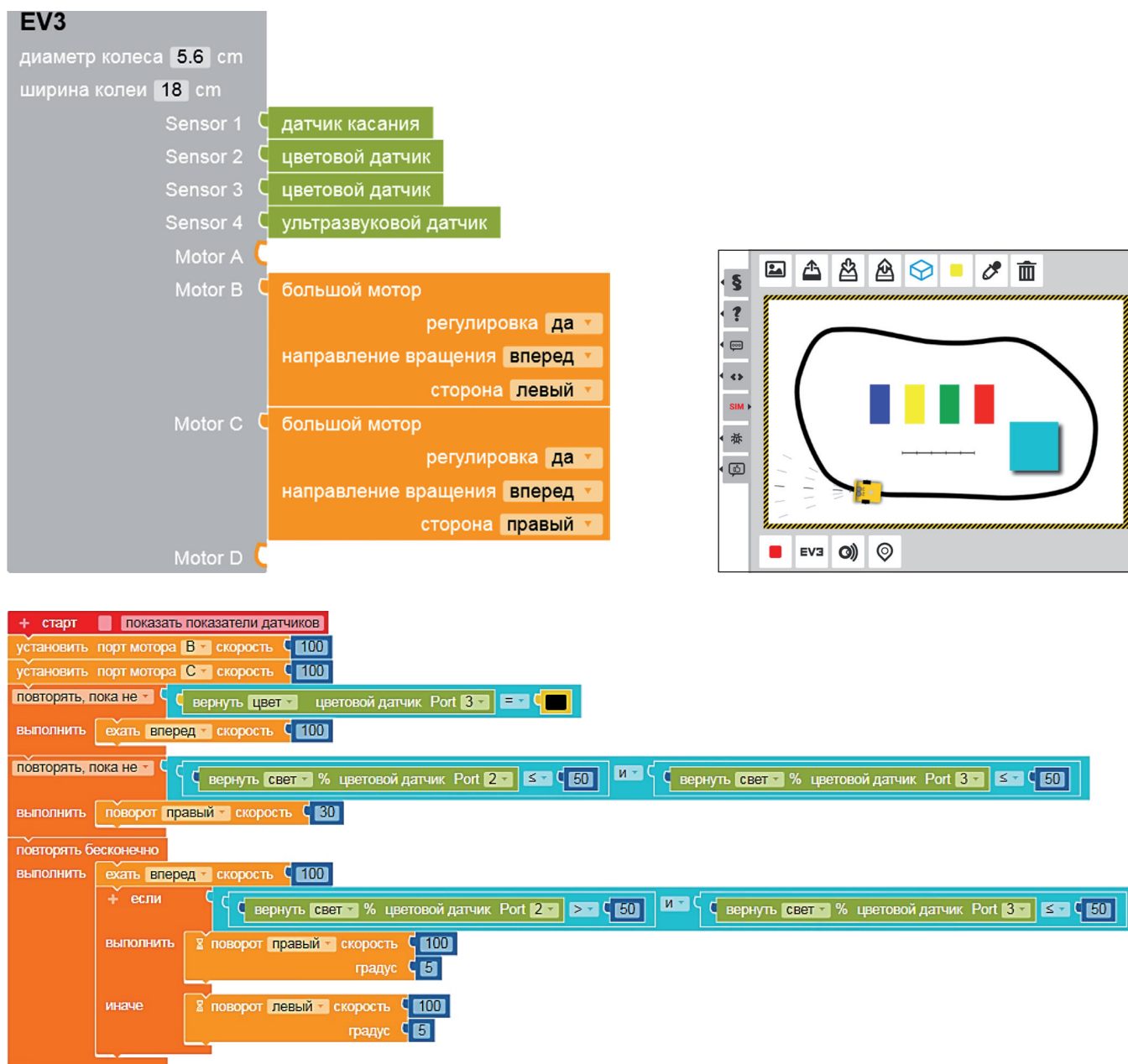


Рис. 12. Решение задачи 5

Задача 6. mBot. Избегание столкновения.

Условие задачи.

Фон симуляции — белый. Расставить на фоне симуляции препятствия. Написать программу, которая позволит роботу не сталкиваться с встречаемыми препятствиями.

Решение задачи представлено на рисунке 13.

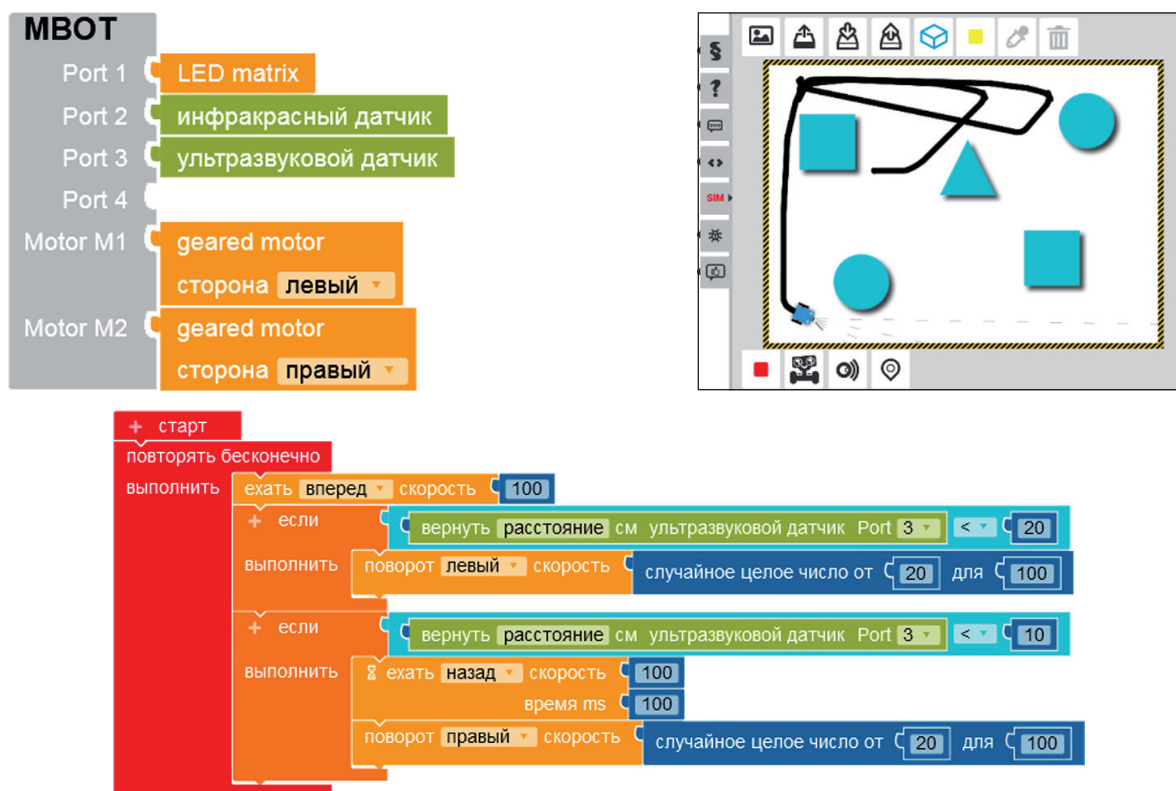


Рис. 13. Решение задачи 6

Задача 7. mBot. Движение в ограниченном пространстве.

Условие задачи.

Расставить на авторском фоне симуляции препятствия. Написать программу, которая позволит роботу перемещаться внутри области, очерченной черной линией и не сталкиваться с встречаемыми препятствиями.

Решение задачи представлено на рисунке 14.

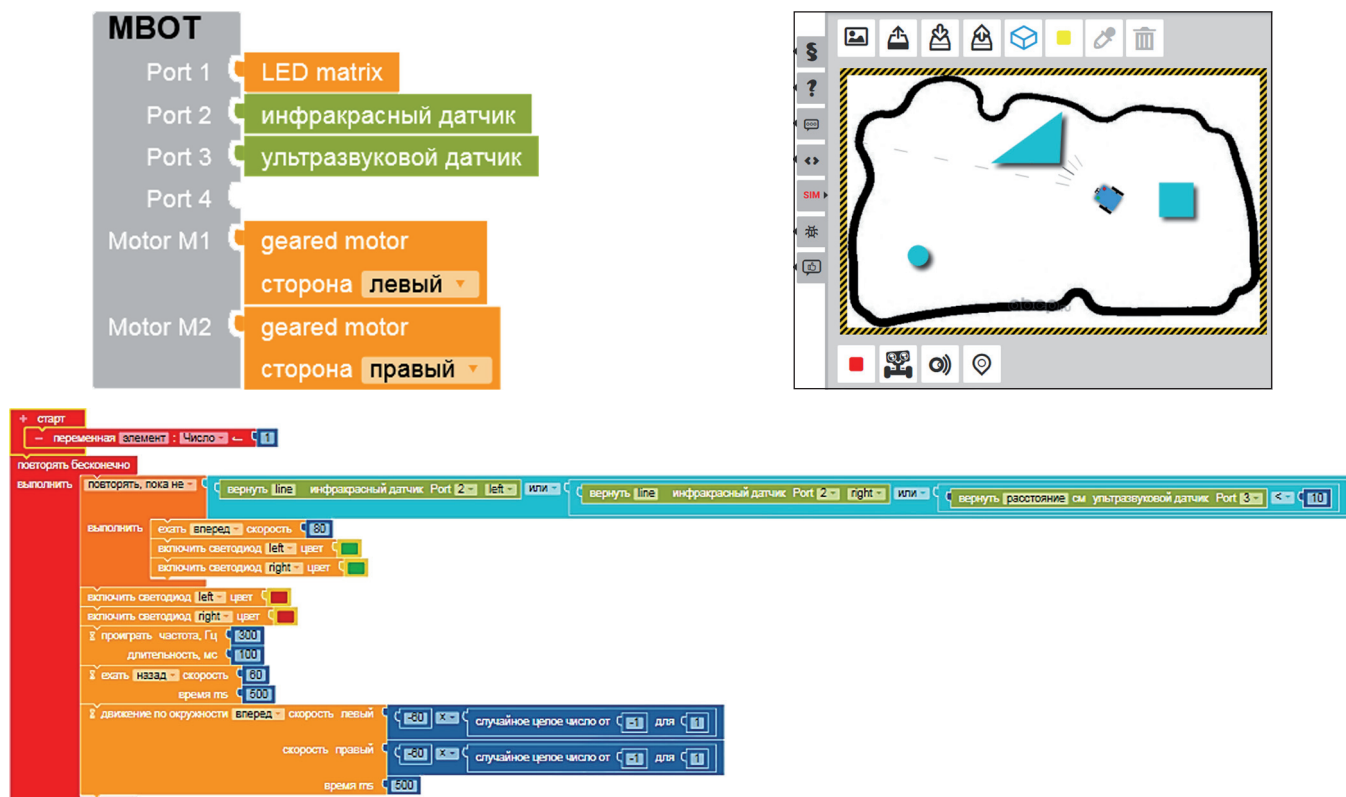


Рис. 14. Решение задачи 7

Задача 8. Calliope mini. Мелодия по выбору.

Условие задачи.

Написать программу, которая позволит роботу проигрывать мелодии в зависимости от нажатия кнопок pin A и pin B [4].

Решение задачи представлено на рисунках 15, 16.

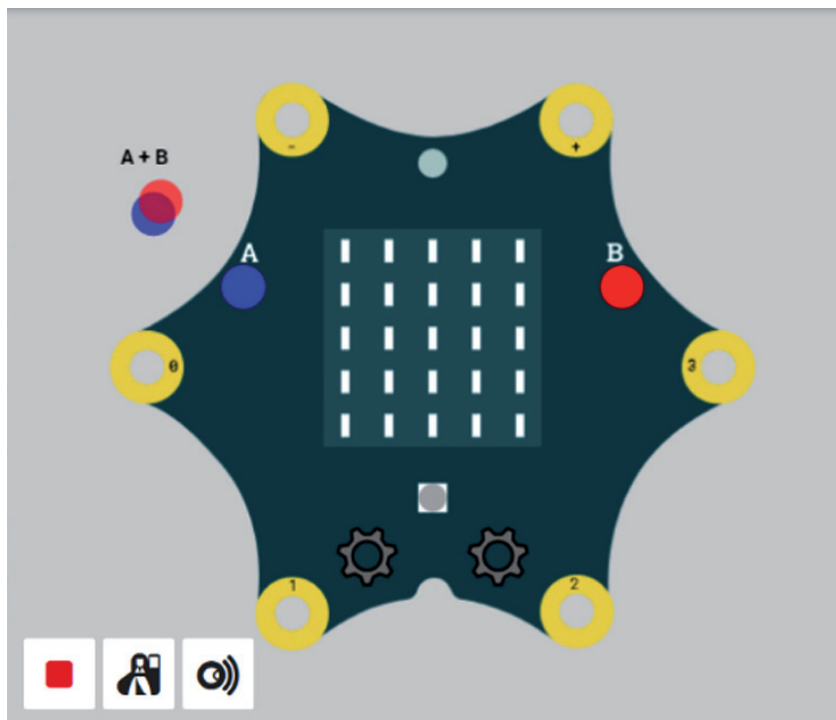


Рис. 15. Робот ждет нажатия пина

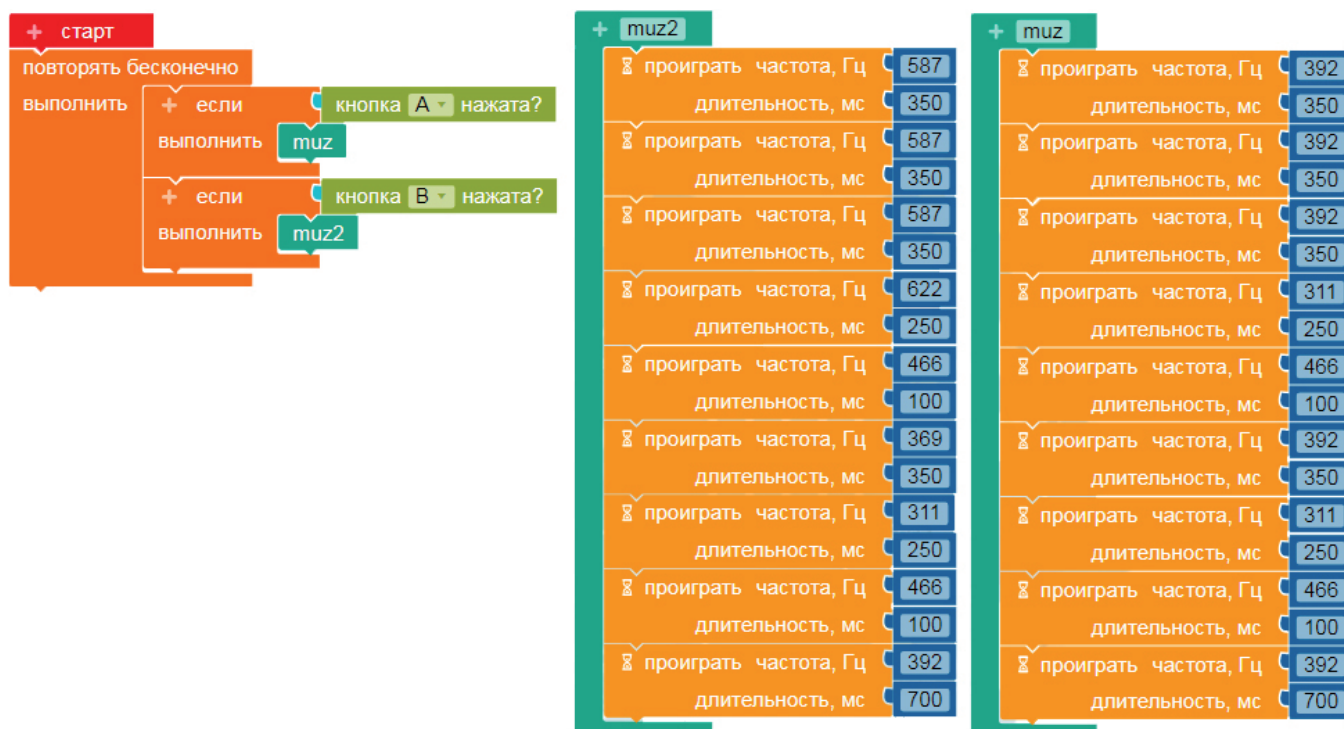


Рис. 16. Решение задачи 8

Задача 9. Calliope mini. Голосовое общение.

Условие задачи.

Написать программу, которая позволит роботу при разных уровнях громкости микрофона реагировать миганием светодиода, показывать изображение, проигрывать мелодию [4].

Решение задачи представлено на рисунках 17, 18.

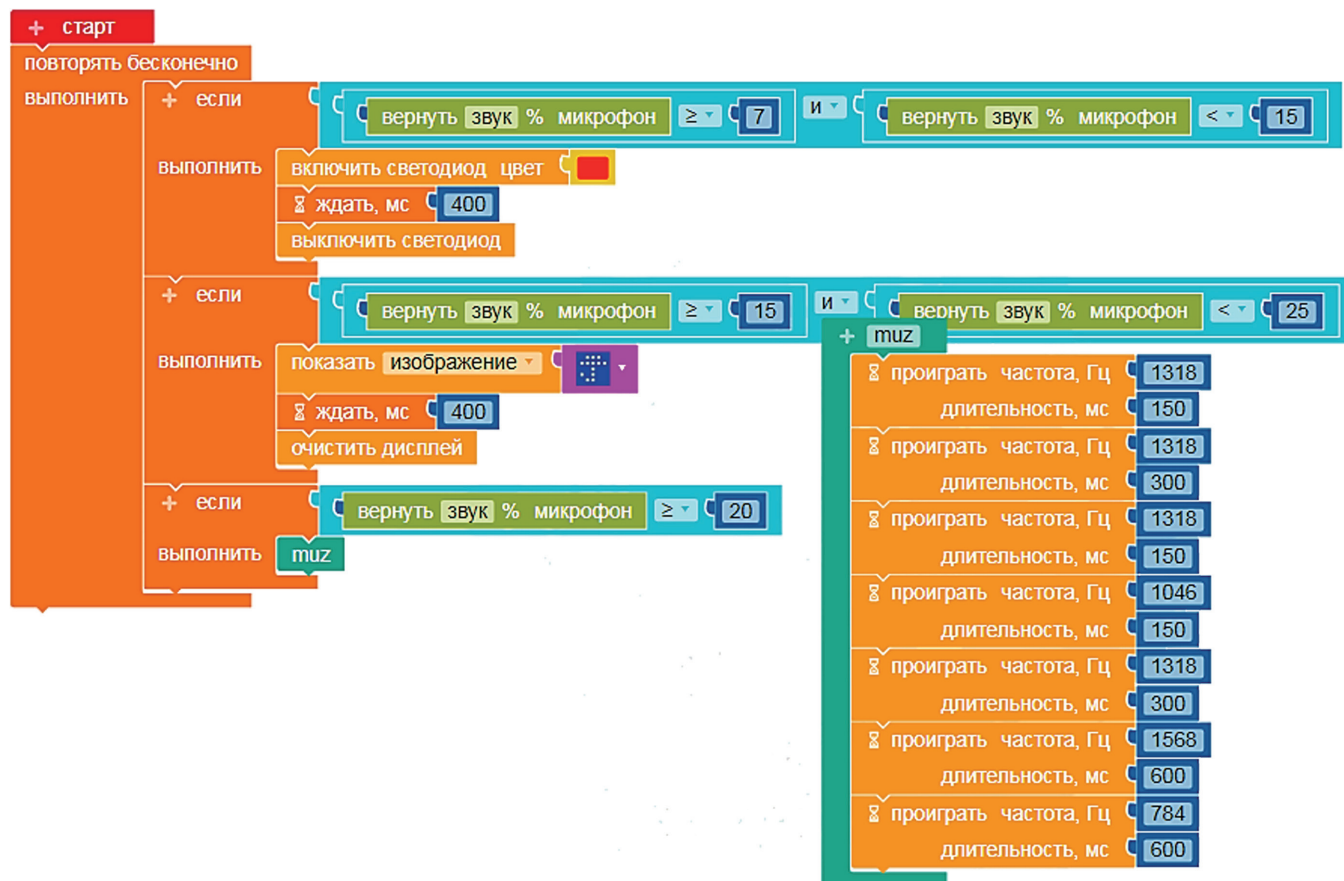


Рис. 17. Решение задачи 9

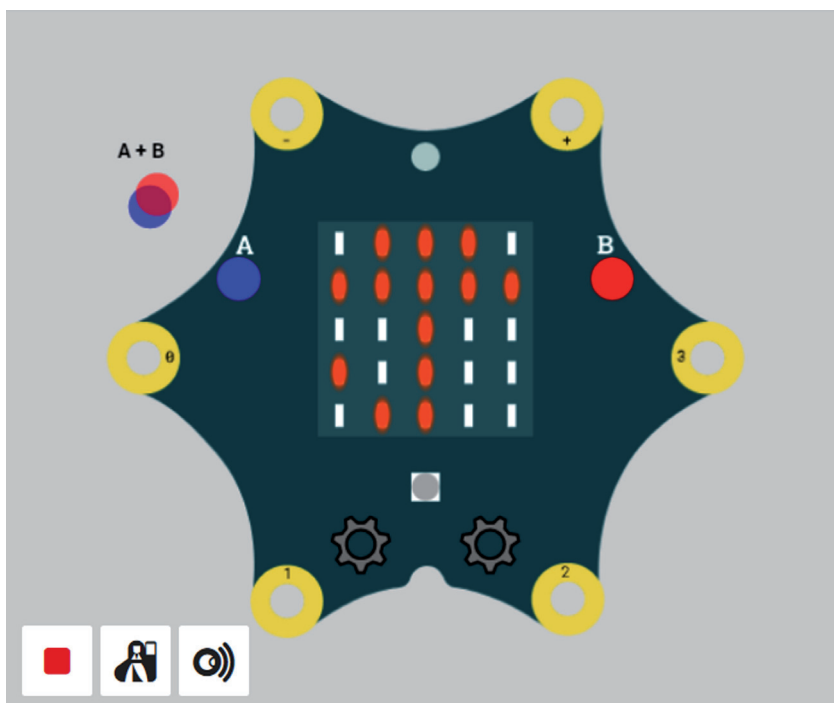


Рис. 18. Реакция робота на уровень громкости голоса

6. Заключение

Использование программ виртуальной робототехники удобно в разных ситуациях: при отсутствии робототехнического оборудования в школе; когда необходимо решать задачи дома или на дистанционном обучении; при подготовке к соревнованиям и конкурсам; а также для развития алгоритмического, вычислительного и креативного мышления школьников, начиная с младших классов.

Рассмотренные в данной статье облачные платформы позволяют в модельных условиях исследовать поведение различных виртуальных роботов на множестве полей, научиться управлять поведением роботов через программирование, развивать способности в области программирования и технических дисциплин.

Список источников

1. Векслер В. А. Виртуальная робототехника с Open Roberta Lab // Актуальные проблемы методики обучения информатике и математике в современной школе. Материалы конференции МПГУ (г. Москва, апрель 2021 г.). <http://news.scienceland.ru/2021/04/18/виртуальная-робототехника-с-open-roberta-lab/>
2. Виртуальная робототехника. Видеокурс. <https://digitalschool.msk.ru/vexcodevr?>
3. Курносенко М. В., Мацаль И. И. Реализация дополнительной общеобразовательной программы по тематическому направлению «Программирование роботов» с использованием оборудования центра цифрового образования детей «IT-куб»: методическое пособие / под ред. С. Г. Григорьева. М.: Центр естественно-научного и математического образования, 2021. 110 с. https://tochkarosta.68edu.ru/wp-content/uploads/2021/06/IT-куб_Программирование-роботов.pdf
4. Мелодии на Arduino. Проекты. Arduino // Arcade PUB. <https://arcadepub.ru/2020/03/20/мелодии-на-ардуино/>
5. Чупин Д. Ю., Ступин А. А., Ступина Е. Е., Классов А. Б. Образовательная робототехника: учебное пособие для студентов педагогических вузов. Новосибирск: Сибпринт, 2019. 114 с. <https://repo.nspu.ru/bitstream/nspu/3632/1/obrazovatel'naya-robototekhnika-uc.pdf>
6. Якимчук Н. В. Применение виртуального робототехнического эмулятора VEXcode VR в дистанционном обучении // Педагогика информатики. 2021. № 2. С. 1–5. <http://sibupk.su/science/research/mconfer2021/1/ПРИМЕНЕНИЕ ВИРТУАЛЬНОГО РОБОТОТЕХНИЧЕСКОГО ЭММУЛЯТОРА VEXCODE VR В ДИСТАНЦИОННОМ ОБУЧЕНИИ.pdf?ysclid=l36bp8lhvc>
7. Open Roberta. Lerne spielerisch programmieren — wie Millionen andere auch! // Open Roberta. <https://www.open-roberta.org/>
8. Open Roberta Lab. Робосимулятор. <https://lab.open-roberta.org/>
9. Teach computer science and coding with VEXcode VR // nooby.tech. <https://nooby.tech/en/content/27-vexcode-vr>
10. VEXcode VR. <https://vr.vex.com/>

DOI: 10.32517/2221-1993-2022-21-3-27-35

В. О. Батракова, С. Н. Веснин

дипломанты Всероссийского конкурса «Школьная информатика: вчера, сегодня, завтра» в номинации «Школьная информатика: вчера, сегодня, завтра», предметно-языковая школа «Дуплекс», г. Пермь, Россия

А. А. Корепанова, Т. Д. Лаптева

дипломанты Всероссийского конкурса «Школьная информатика: вчера, сегодня, завтра» в номинации «Школьная информатика: вчера, сегодня, завтра», Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия

А. И. Саврасов

дипломант Всероссийского конкурса «Школьная информатика: вчера, сегодня, завтра» в номинации «Школьная информатика: вчера, сегодня, завтра», предметно-языковая школа «Дуплекс», г. Пермь; Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия

ЭЛЕКТИВНЫЙ КУРС «ПРИМЕНЕНИЕ МЕТОДОВ МОДЕЛИРОВАНИЯ СИСТЕМ ПРИ РАБОТЕ С ТЕКСТОВОЙ ИНФОРМАЦИЕЙ»

Контактная информация

Батракова Виктория Олеговна, учитель информатики, предметно-языковая школа «Дуплекс», г. Пермь, Россия; *адрес:* 614068, Россия, г. Пермь, ул. Толмачева, д. 18; *e-mail:* viktoriyabatrakova20@gmail.com

Веснин Сергей Николаевич, учитель информатики, предметно-языковая школа «Дуплекс», г. Пермь, Россия; *адрес:* 614068, Россия, г. Пермь, ул. Толмачева, д. 18; *e-mail:* vesnin-sn@duplex.permedu.ru

Корепанова Алла Александровна, студентка 1-го курса магистратуры направления подготовки 44.04.01 «Педагогическое образование», профиль «Инновационные процессы в образовании и науках», Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия; *адрес:* 614990, Россия, г. Пермь, ул. Сибирская, д. 24; *e-mail:* allo4kinapo4ta@mail.ru

Лаптева Татьяна Дмитриевна, студентка 1-го курса магистратуры направления подготовки 44.04.01 «Педагогическое образование», профиль «Управление цифровой школой», Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия; *адрес:* 614990, Россия, г. Пермь, ул. Сибирская, д. 24; *e-mail:* tanyal1998@mail.ru

Саврасов Антон Игоревич, учитель информатики, предметно-языковая школа «Дуплекс», г. Пермь, Россия; *адрес:* 614068, Россия, г. Пермь, ул. Толмачева, д. 18;

студент 1-го курса магистратуры направления подготовки 44.04.01 «Педагогическое образование», профиль «Инновационные процессы в образовании и науках», Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия; *адрес:* 614990, Россия, г. Пермь, ул. Сибирская, д. 24;

e-mail: greencoil.as@gmail.com

V. O. Batrakova, S. N. Vesnin

School "Duplex", Perm, Russia

A. A. Korepanova, T. D. Lapteva

Perm State Humanitarian Pedagogical University, Perm, Russia

A. I. Savrasov

School "Duplex", Perm;

Perm State Humanitarian Pedagogical University, Perm, Russia

THE ELECTIVE COURSE "APPLICATION OF METHODS OF THE MODELING OF SYSTEMS WHILE WORKING WITH TEXT INFORMATION"

Abstract

The article discusses the concepts of linear and spatial text, describes the advantages and disadvantages of the latter. The elective course aimed at developing students' skills in processing text information using various approaches to modeling of systems by using modeling methods, including the use of graphic languages GNOM, MOLNIYA, GRAF, DRAGON, the use of the American standard for modeling production processes IDEF-0 and DFD Data Flow Diagrams is described in the article. The structure and content line of the course are described. Examples of students' work performed as part of the study of each of the sections of the course "Application of methods of the modeling of systems while working with text information" are given (the target audience of the course is students of the 10th grade of a comprehensive school who study informatics at a basic level). Based on the experience of implementing the described course, a number of conclusions have been formulated that characterize the results achieved.

Keywords: modeling, text, training, information, summary, graphic language, informatics.

Аннотация

В статье рассмотрены понятия линейного и пространственного текста, описаны преимущества и недостатки последнего. Представлена разработка элективного курса, направленного на формирование у обучающихся навыков обработки текстовой информации с применением различных подходов к моделированию систем путем использования методов моделирования, среди которых: применение графических языков ГНОМ, МОЛНИЯ, ГРАФ, ДРАКОН, использование американского стандарта моделирования производственных процессов IDEF-0 и диаграммы потоков данных DFD. Описаны структура и содержательная линия курса. Приведены примеры работ учащихся, выполненных в рамках изучения каждого из разделов курса «Применение методов моделирования систем при работе с текстовой информацией» (целевой аудиторией курса являются ученики X класса общеобразовательной школы, изучающие информатику на базовом уровне). На основе опыта реализации описанного курса сформулирован ряд выводов, характеризующих достигнутые результаты.

Ключевые слова: моделирование, текст, обучение, информация, конспект, графический язык, информатика.

1. Введение

В современном обществе информация как ресурс имеет высокую ценность и значение. Умения и навыки работы с ней во многом определяют успешность человека в любой сфере деятельности [6, с. 326]. Поскольку одной из основных форм представления информации является текст, у обучающихся необходимо развивать умения работы с ним.

По форме представления исследователи выделяют линейный и пространственный тексты (рис. 1) [1].



Рис. 1. Виды текста

Линейный текст — это сплошной массив последовательно выстроенных предложений, который преобладает в книгах, учебниках, статьях и других информационных источниках. Пространственный текст, в свою очередь, представляет собой логическую схему, демонстрирующую основные понятия, их свойства и связи между ними [8, с. 69].

Пространственный текст имеет как достоинства, так и недостатки (см. табл.).

Эффективным методом создания пространственного текста является моделирование, которое может служить основой стандартизации конспекта путем при-

менения подходящей нотации, связывающей конкретный элемент текста (понятие, свойство понятия, шаг процесса и т. д.) и визуальный образ — блок диаграммы (прямоугольник, овал и т. д.).

Поскольку создание пространственного текста требует тренировки и знакомства с различными средствами и способами его построения, можно предложить обучающимся элективный курс, посвященный изучению различных методов моделирования применительно к созданию пространственного текста.

Приведем пример организации такого элективного курса, разработанного и реализуемого в предметно-языковой школе «Дуплекс» города Перми учителями информатики С. Н. Весниным (автор идеи курса), В. О. Батраковой и А. И. Саврасовым.

2. Характеристика элективного курса «Применение методов моделирования систем при работе с текстовой информацией»

2.1. Цель курса

Целью элективного курса «Применение методов моделирования систем при работе с текстовой информацией» является формирование у обучающихся умения работать с информацией, применяя методы моделирования. Широкие возможности методов и подходов к моделированию позволят обучающимся увидеть межпредметные связи. Применение моделирования для работы с текстами разных тематик из различных

Таблица

Достоинства и недостатки пространственного текста

Достоинства	Недостатки
Четко выделенная и логически выстроенная содержательная структура, основой которой является блочное деление информации на небольшие взаимосвязанные порции	Ограниченность используемого пространства
Простота визуального восприятия	При составлении пространственного текста, содержащего большой объем существенной информации, возникает риск создания громоздких, излишне детализированных схем, затрудняющих поиск информации
Более высокая относительно линейного текста скорость поиска информации	Составление конспекта в форме пространственного текста требует тренировки
Рассмотрение текста с позиции не только содержания, но и возможностей расположения в пространстве (на листе бумаги или на экране компьютера)	

научных областей создает новую учебную ситуацию, в которой учащиеся смогут овладеть умениями и навыками моделирования. Кроме того, новая ситуация повышает познавательную мотивацию обучающихся и их вовлеченность в процесс обучения [9].

2.2. Задачи курса

- Знакомство учащихся с понятием «моделирование» и подходами к моделированию систем;
- расширение представлений обучающихся о моделировании как инструменте работы с информацией;
- формирование умения применять методы моделирования при работе с текстом;
- формирование межпредметных связей;
- развитие умения совершать логические операции при работе с информацией;
- расширение кругозора учащихся посредством работы с текстами разных тематик.

2.3. Место и роль курса в образовательном процессе

Предлагаемый элективный курс является межпредметным с преобладанием одной дисциплины (информатики) [3]. Он разработан для учащихся X класса базового уровня обучения. Курс позволяет удовлетворить познавательные потребности учащихся. В процессе изучения данного элективного курса школьники овладевают новыми знаниями и способами деятельности, получают возможность практического применения своих интеллектуальных способностей. Курс является средством перехода от обучения учащихся к их самообразованию. Разработанный курс рассчитан на любого ученика старшей школы. Дополнительных знаний и умений по предмету для успешного освоения его программы не требуется.

2.4. Сроки реализации

Курс рассчитан на две четверти и включает в себя 16 занятий по 45 минут.

2.5. Методы, формы, режим занятий

Элективный курс позволяет использовать различные формы занятий — лекции, лабораторные работы, а также тематические задания по подгруппам.

2.6. Предполагаемые результаты обучения

Освоение материалов курса предполагает формирование у обучающихся следующих личностных, метапредметных и предметных результатов обучения:

личностные:

- формирование ответственного отношения к учению, готовности и способности к саморазвитию и самообразованию на основе мотивации к обучению и познанию [9];
- формирование умений и навыков работы с текстом;

метапредметные:

- умение определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации, устанавливать причинно-следственные связи, строить логические рассуждения, умозаключения и делать выводы;

- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для представления различной информации;
- смысловое чтение [4, с. 9];

предметные:

- знание подходов к моделированию;
- умение применять различные подходы при работе с текстом;
- умение выбирать наиболее эффективный подход в заданных условиях.

2.7. Инструментарий для оценивания

Оценивание достижения планируемых результатов состоит из промежуточного и итогового контроля, которые проводятся не в виде выполнения теста и письменных работ, а в виде выполнения продуктивных заданий и создания проектов.

Работы оцениваются по следующим критериям:

- самостоятельность выполнения работы;
- логически верное построение мысли;
- умение представить текст в пространстве, используя изученные методы и схемы;
- целесообразность выбора метода моделирования применительно к конкретному тексту.

Промежуточный контроль достижений обучающихся осуществляется после изучения каждого раздела курса в формате индивидуальной работы.

Итоговый контроль проходит в форме конкурса работ учащихся.

3. Тематическое планирование курса

№ п/п	Раздел элективного курса	Кол-во акад. часов
1	Основы моделирования	2
2	Основы создания графического конспекта	2
3	Объектно-ориентированный подход к моделированию	3
4	Функциональный подход к моделированию	3
5	Моделирование потоков данных	3
6	Процессно-ориентированный подход к моделированию	3
	ИТОГО:	16

4. Содержание курса

Элективный курс «Применение методов моделирования систем при работе с текстовой информацией» апробирован на уроках информатики в X классе, поэтому описание его содержания будет сопровождаться демонстрацией работ, выполненных учащимися.

4.1. Основы моделирования

В рамках данного раздела рассматриваются:

- понятия «модель», «моделирование», «оригинал», «система», «декомпозиция», «подсистема»;

- основное свойство модели;
- классификации моделей по различным основаниям;
- способы построения моделей;
- причины построения моделей;
- примеры разнообразных моделей из различных областей человеческой деятельности [2, с. 7–31].

В результате работы на этом этапе учащиеся:

- овладевают основным понятийным аппаратом по теме «Моделирование»;
- учатся классифицировать модели по различным признакам, определять причины создания различных моделей, а также описывать достоинства и недостатки различных моделей в зависимости от предъявляемых к модели требований.

4.2. Основы создания графического концепта

В рамках этого раздела:

- рассматриваются понятия «линейный текст», «пространственный текст», особенности этих ви-

дов текста, их сходства и различия; описываются примеры, достоинства и недостатки этих видов текста;

- вводится понятие «графический концепт», описываются основные правила его построения;
- рассматриваются примеры линейного текста и построенного на его основе графического концепта;
- рассматриваются различные онлайн-сервисы, позволяющие создавать графический концепт и используемые далее в рамках всего курса.

В результате освоения этого раздела учащиеся:

- усваивают понятия «линейный текст», «пространственный текст», «графический концепт»;
- овладевают первичными навыками составления концепта в общем и графического концепта в частности (или развивают такие навыки);
- рассматривают различные программные средства составления графического концепта, их достоинства, недостатки, возможности и ограничения.

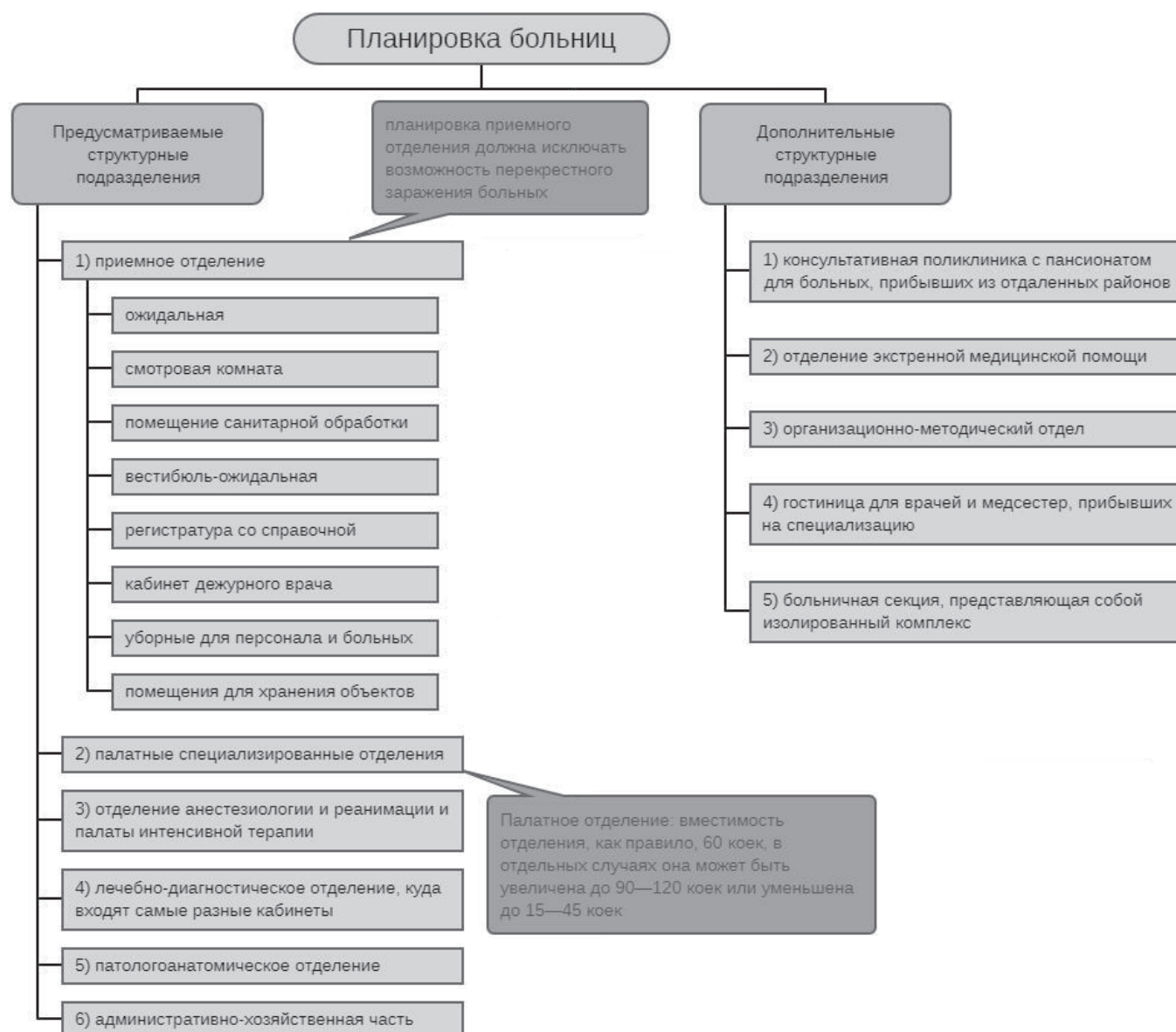


Рис. 2. Использование объектно-ориентированного подхода к моделированию при создании пространственного текста. Пример 1 (разработан в сервисе Drakonhub.com)

4.3. Объектно-ориентированный подход к моделированию

Раздел посвящен:

- рассмотрению семейства графических языков, разработанных советским и российским ученым В. Д. Паронджановым;
- правилам и рекомендациям по созданию графических схем с использованием языков ГНОМ, МОЛНИЯ, ГРАФ;
- их использованию при составлении графических конспектов отдельных понятий, правил, а также учебных, научных и других текстов.

В рамках этого раздела текст рассматривается как система, основными элементами которой выступают понятия — объекты, сущность и взаимосвязь которых раскрывается при помощи графического конспекта.

В результате освоения раздела учащиеся:

- знакомятся с особенностями объектно-ориентированного подхода к моделированию;
- формируют и развивают навыки применения данного подхода к созданию графического конспекта текста;
- изучают возможности использования графических языков при структурировании текстовой информации;
- формируют и развивают навыки использования различных онлайн-сервисов;
- учатся осуществлять целесообразный выбор средств решения задачи (выбор наиболее удобного онлайн-сервиса, а также подходящего под ситуацию графического языка).

На рисунках 2, 3 приведены примеры работ учащихся, демонстрирующих реализацию объектно-ориентированного подхода.

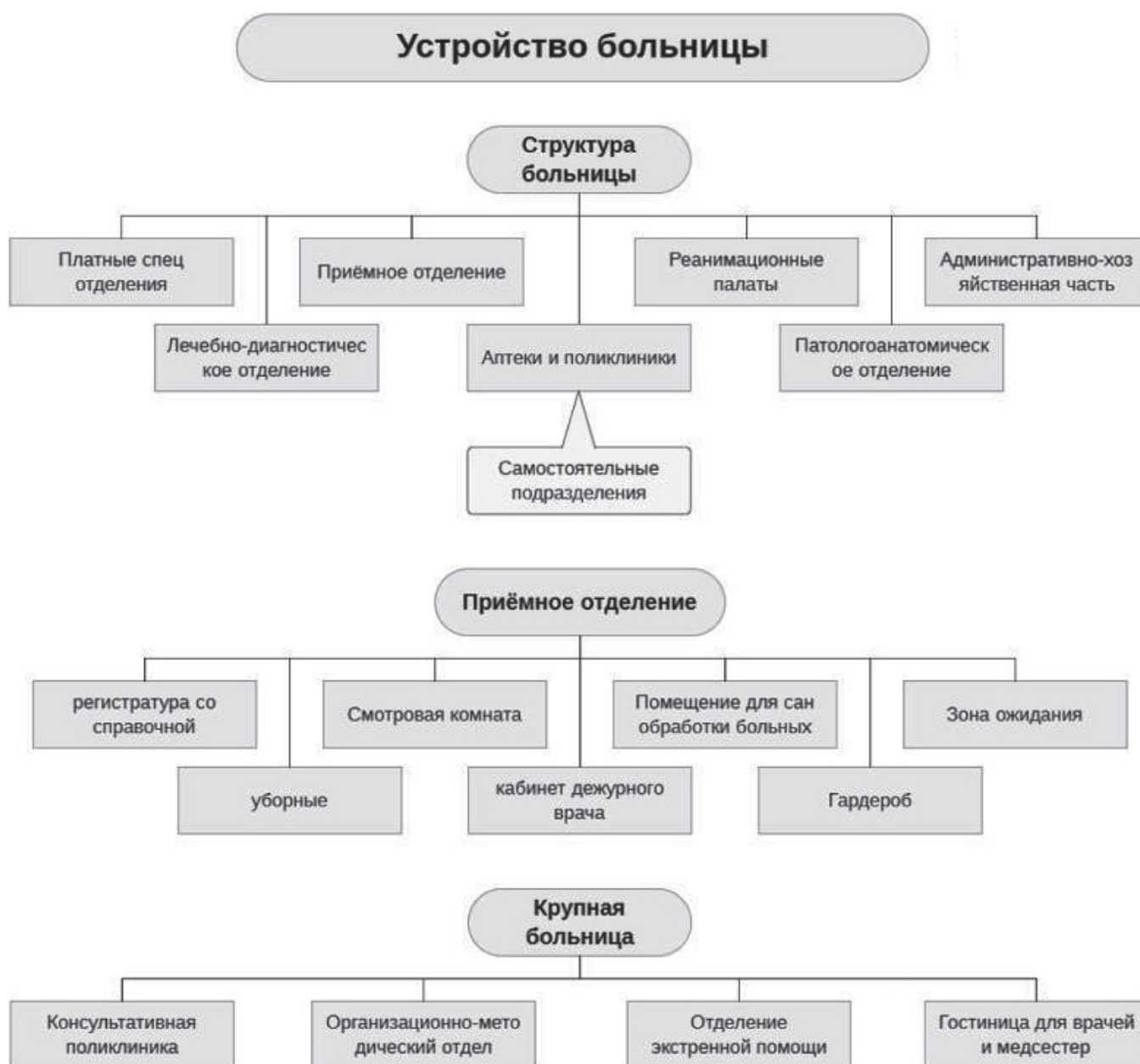


Рис. 3. Использование объектно-ориентированного подхода к моделированию при создании пространственного текста. Пример 2 (разработан в сервисе *Drakonhub.com*)

4.4. Функциональный подход к моделированию

В рамках данного раздела рассматривается сущность функционального подхода к моделированию систем.

В качестве средства для создания концептов изучается и используется стандарт моделирования производственных процессов IDEF-0.

В результате освоения раздела обучающиеся:

- знакомятся с особенностями функционального подхода к моделированию;
- формируют и развивают навыки применения данного подхода к созданию графического концепта текста;

- изучают возможности использования стандарта IDEF-0 [10, с. 15–21] при описании функциональных взаимосвязей элементов системы, описанной в тексте;
- формируют и развивают навыки применения различных онлайн-сервисов;
- учатся осуществлять целесообразный выбор средств решения задачи (выбор наиболее удобного онлайн-сервиса; выбор метода моделирования, подходящего для составления концепта конкретного текста).

На рисунках 4, 5 приведены примеры работ учащихся, демонстрирующих реализацию функционального подхода.

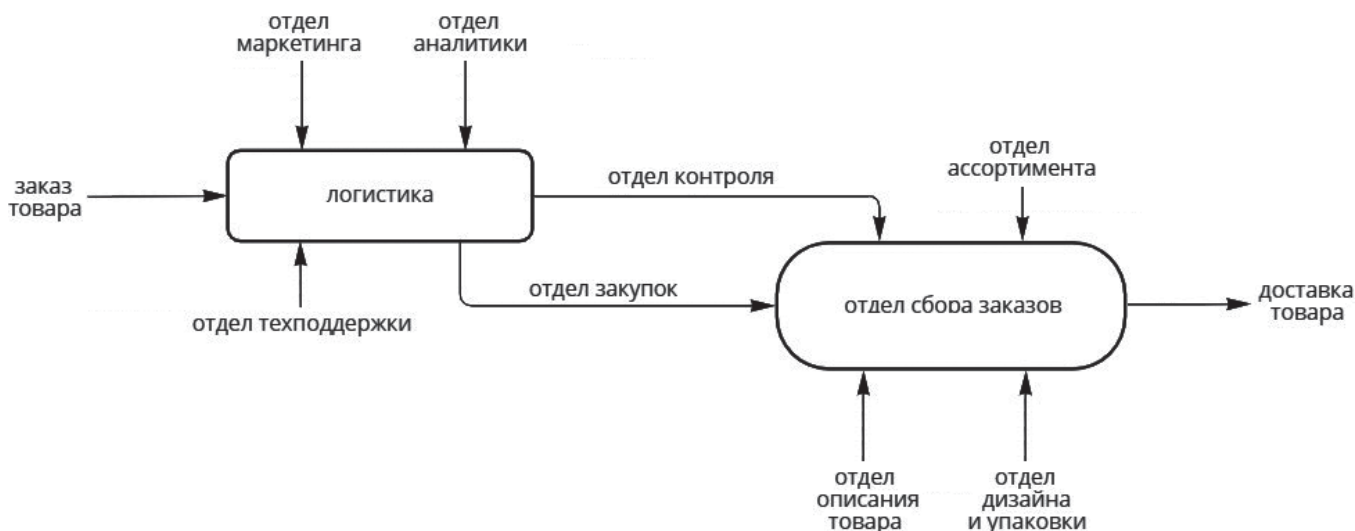


Рис. 4. Использование функционального подхода к моделированию при создании пространственного текста. Пример 1 (разработан в сервисе Lucidchart.com)

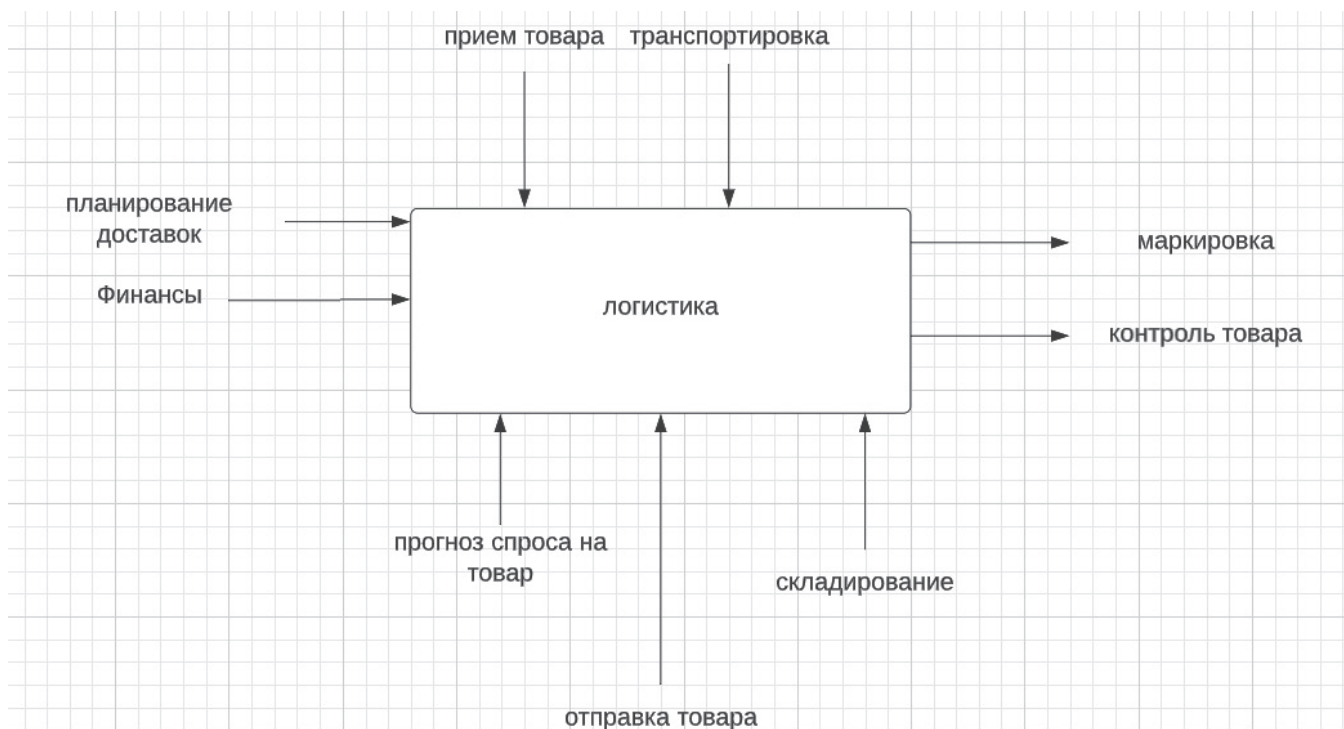


Рис. 5. Использование функционального подхода к моделированию при создании пространственного текста. Пример 2 (разработан в сервисе Lucidchart.com)

4.5. Моделирование потоков данных

В данном разделе рассматривается методология DFD (Data Flow Diagrams) [5], описывающая моделируемую систему как черный ящик и акцентирующая внимание на входных и выходных потоках данных, протекающих между ее подсистемами (рис. 6).

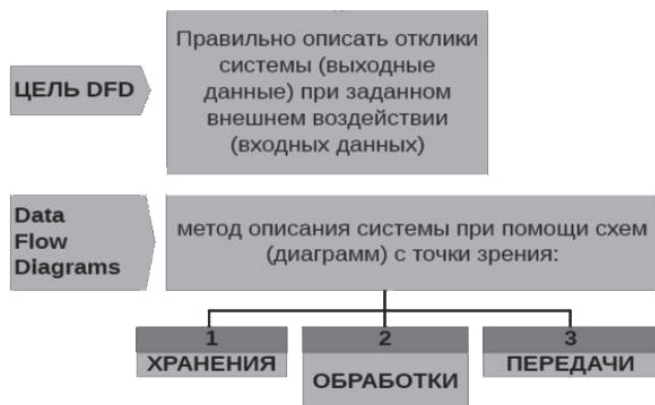


Рис. 6. Краткая характеристика DFD-подхода

В результате освоения раздела обучающиеся:

- знакомятся с особенностями DFD-подхода к моделированию;
- формируют и развивают навыки применения данного подхода к созданию графического конспекта текстов;
- изучают возможности использования различных онлайн-сервисов, формируют и развивают соответствующие навыки;
- учатся осуществлять целесообразный выбор средств решения задачи (выбор наиболее удобного онлайн-сервиса; выбор наиболее подходящего под ситуацию подхода к моделированию текста).

На рисунке 7 приведен пример работы учащегося, демонстрирующий реализацию DFD-подхода.

4.6. Процессно-ориентированный подход к моделированию

Раздел посвящен рассмотрению процессно-ориентированного подхода к моделированию. В качестве средства построения конспекта предлагается использовать разработанный В. Д. Паронджановым графический язык ДРАКОН. Учащиеся знакомятся с правилами и возможностями языка ДРАКОН, рассматривают его в контексте составления графических конспектов текстов и их элементов, описывающих различные процессы.

В рамках этого раздела также:

- освещается вопрос о применении блок-схем при изучении и описании процессов (алгоритмов);
- рассматриваются отличия между применением блок-схем и языка ДРАКОН;
- описываются существенные преимущества языка ДРАКОН по сравнению с блок-схемами, принятыми к использованию, например, при изучении алгоритмов в школьных УМК по информатике [7].

В результате освоения раздела обучающиеся:

- знакомятся с особенностями процессно-ориентированного (процедурного) подхода к моделированию;
- формируют и развивают навыки применения данного подхода к созданию графического конспекта текста;
- изучают возможности использования графического языка ДРАКОН при структурировании текстовой информации;
- формируют и развивают навыки использования различных онлайн-сервисов, а также учатся осуществлять целесообразный выбор средств решения задачи.

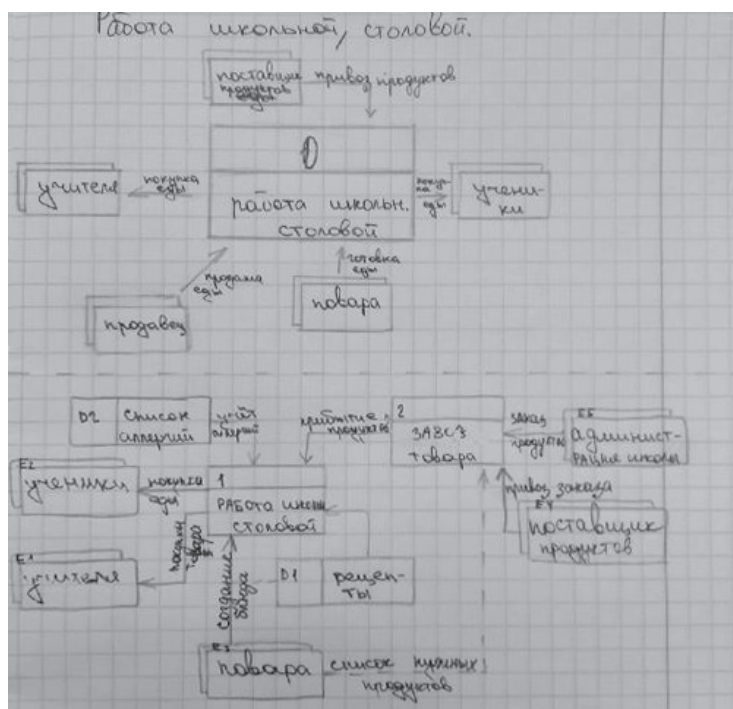


Рис. 7. Использование моделирования потоков данных при создании пространственного текста

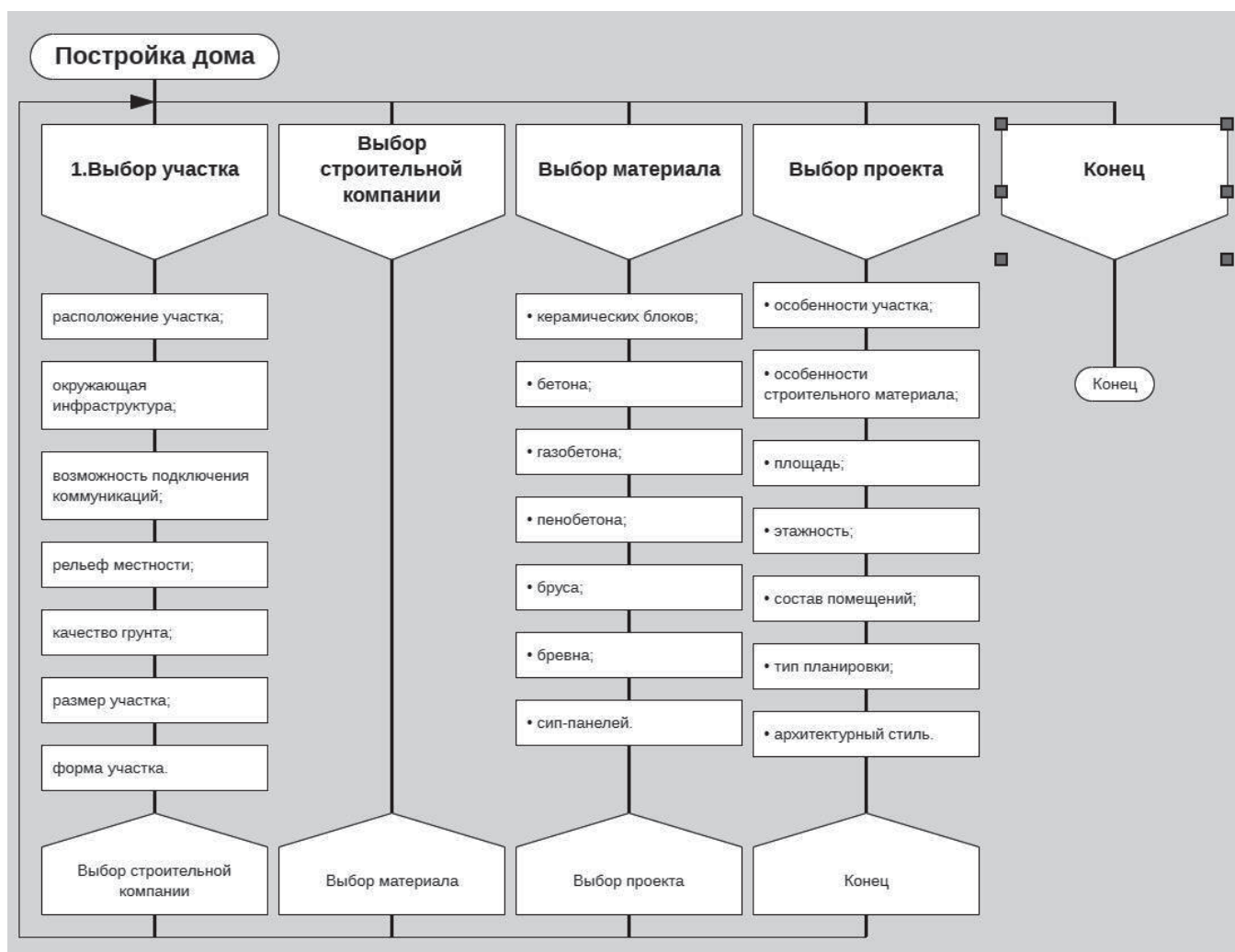


Рис. 8. Использование процессно-ориентированного подхода к моделированию при создании пространственного текста. Пример 1 (разработан в сервисе Draconhub.com)

На рисунках 8, 9 приведены примеры работ учащихся, демонстрирующих реализацию процессно-ориентированного подхода.

5. Выводы

На основе опыта реализации описанного в работе элективного курса для учащихся X класса общеобразовательной школы можно сформулировать следующие выводы:

- использование различных методов моделирования позволяет эффективно структурировать информацию путем преобразования линейного текста к пространственному виду;
- создание пространственного текста позволяет обеспечить условия для повышения и сохранения уровня познавательной мотивации обучающихся при работе с текстовой информацией;
- изучение различных способов моделирования позволяет предоставить учащимся широкий набор средств создания пространственного конспекта текста и сформировать навыки использования этих средств в различных учебных и жизненных ситуациях.

Список источников

1. Борисова С. А. Пространство текста как материального объекта // Вестник Оренбургского государственного университета. 2003. № 4. С. 158–162.
2. Горностаева Т. Н., Горностаев О. М. Математическое и компьютерное моделирование: учебное пособие. М.: Мир науки, 2019. <https://izd-mn.com/PDF/50MNNPU19.pdf>
3. Забусова А. А. Предметно-ориентированные курсы для классов математического профиля // Обучение и воспитание: методики и практика. 2013. № 5. С. 75–79.
4. Казарская Г. Е. Формирование и развитие на уроке универсальных учебных действий // Эксперимент и инновации в школе. 2015. № 3. С. 7–12.
5. Калянов Г. Н. Концептуальная модель DFD-технологии // Открытое образование. 2017. № 4. С. 21–26.
6. Корепанова А. А. Элективные курсы по геометрии Н. И. Лобачевского как форма формирования и развития мышления учащихся // Формирование мышления в процессе обучения естественнонаучным, технологическим и математическим дисциплинам. Материалы Всероссийской научно-практической конференции, приуроченной к юбилею Тамары Николаевны Шамало (Екатеринбург, 26–27 октября 2020 г.). Екатеринбург: УрГПУ, 2020. С. 326–330.
7. Паронджанов В. Д. Алгоритмы и жизнеритмы на языке ДРАКОН. Разработка алгоритмов. Безошибочные алгоритмы. М., 2019. 374 с.

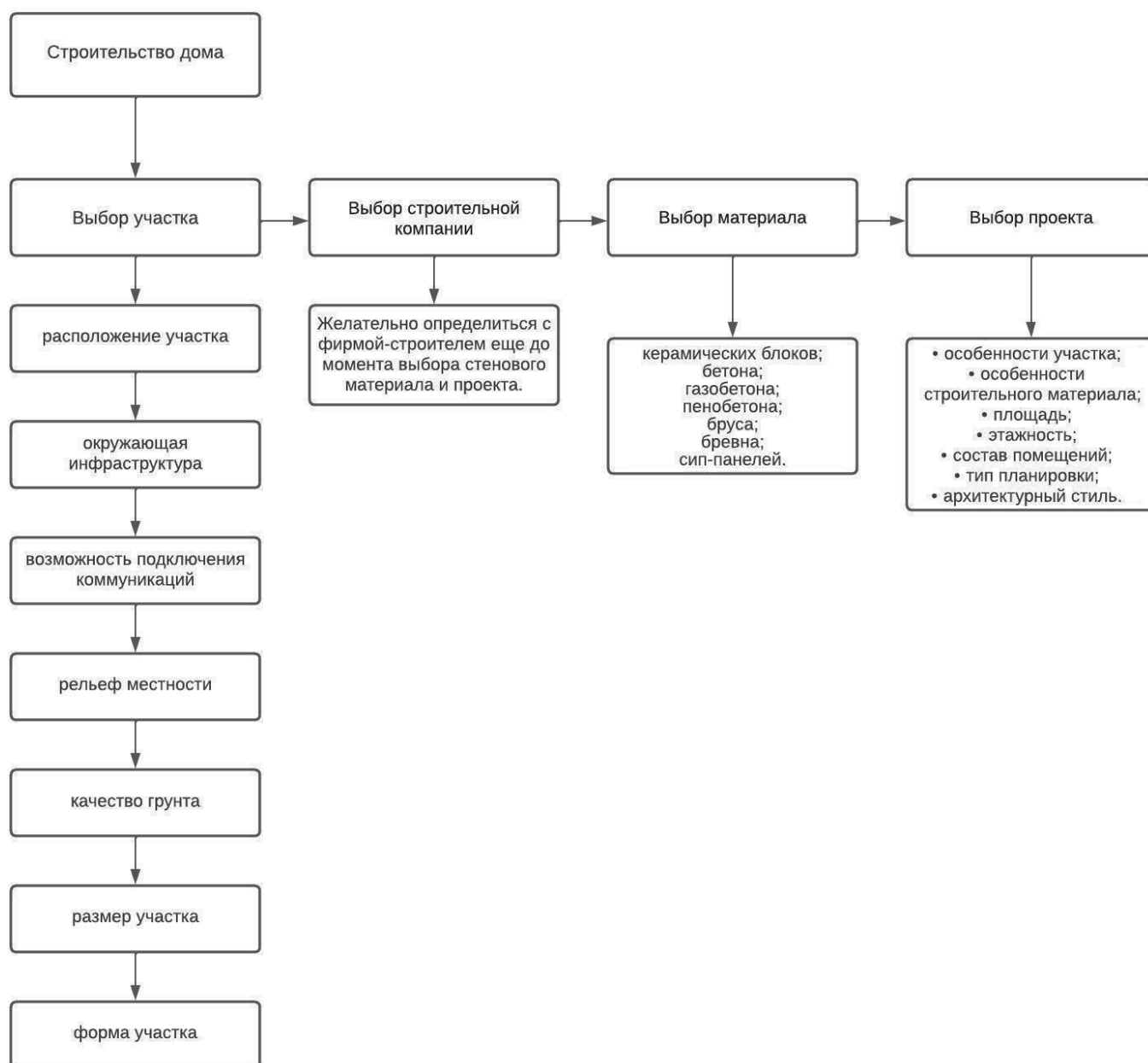


Рис. 9. Использование процессно-ориентированного подхода к моделированию при создании пространственного текста.
Пример 2 (разработан в сервисе Lucidchart.com)

8. Паронджанов В. Д. Как написать хороший учебник для хороших людей. Учебники, о которых мечтают студенты и школьники. М.: ДМК Пресс, 2017. 500 с.

9. Скорнякова А. Ю., Лантева Т. Д. Геймификация как способ повышения мотивации к изучению истории математики //

Сборник трудов международной конференции «Математическое образование 8». Ереван: АГПУ им. Х. Абовяна, 2020. С. 163–166.

10. Цуканова О. А. Методология и инструментарий моделирования бизнес-процессов: учебное пособие. СПб.: Университет ИТМО, 2015. 100 с.

DOI: 10.32517/2221-1993-2022-21-3-36-48

М. А. Плаксин*дипломант Всероссийского конкурса «Школьная информатика: вчера, сегодня, завтра»**в номинации «Школьная информатика: вчера, сегодня, завтра»,**Национальный исследовательский университет «Высшая школа экономики» (Пермский филиал);**Пермский государственный национальный исследовательский университет, г. Пермь, Россия*

ИГРЫ-ХОДИЛКИ ДЛЯ ИЗУЧЕНИЯ БЛОК-СХЕМ*

Аннотация

В статье описывается методика освоения блок-схем с помощью настольных игр-ходилки.

Игровое поле в такой игре представляет собой блок-схему. В ходе игры учащиеся бросают кубик и передвигают фишку по игровому полю в зависимости от выпавшего числа. Но при этом они должны соблюдать правила исполнения входящих в блок-схему алгоритмических конструкций.

Цель предлагаемого подхода — интенсифицировать освоение учебного материала и освободить время для его расширения. Это актуально для всех школьных предметов, но для информатики с ее крайне быстрыми темпами развития — особенно.

Для игры класс разбивается на команды по два—четыре человека. Все команды играют одновременно. В ходе игры каждый участник многократно проходит все изученные алгоритмические конструкции. Таким образом достигается стопроцентный охват учащихся и многократное повторение материала.

За счет этого появляется возможность продвинуться в изучении алгоритмических конструкций достаточно далеко. А именно: кроме традиционных развилок и двух видов циклов (с предусловием и с заданным числом повторений) изучаются команды ввода, подпрограммы (вспомогательные алгоритмы) и параметры (в играх — только входные).

Знакомство с механизмом подпрограмм, причем параметризованных, позволит в будущем рассматривать достаточно сложные алгоритмы и плавно перейти к пропедевтике технологии программирования (наращивание мощности простых исполнителей за счет разработки мощных команд-подпрограмм, проектирование алгоритмов «сверху вниз», построение алгоритмов из крупных заготовок).

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_03-2022/

Контактная информация

Плаксин Михаил Александрович, канд. физ.-мат. наук;

доцент кафедры информационных технологий в бизнесе, Национальный исследовательский университет «Высшая школа экономики» (Пермский филиал), г. Пермь, Россия; *адрес:* 614070, г. Пермь, ул. Студенческая, д. 38;

доцент кафедры математического обеспечения вычислительных систем, Пермский государственный национальный исследовательский университет, г. Пермь, Россия; *адрес:* 614990, г. Пермь, ул. Букирева, д. 15;

e-mail: mapl@list.ru

M. A. Plaksin,

National Research University Higher School of Economics, Perm Branch;

Perm State National Research University, Perm, Russia

BOARD GAMES FOR THE STUDY OF FLOW CHARTS

Abstract

The article presents the description of method of the study of flow charts using board games.

The playing field in such a game is a flow chart. During the game, students roll the dice and move the chip around the playing field, depending on the number that has fallen. But at the same time, they must comply with the rules for the execution of the algorithmic structures included in the flow chart.

The purpose of the proposed teaching method is to intensify the learning of educational material. This is especially important for informatics with its extremely fast rates of development.

For the game, the class is divided into teams of 2–4 people. All teams play at the same time. During the game, each participant repeatedly goes through all the studied algorithmic constructions. Thus, one hundred percent coverage of students and repeated repetition of the material is achieved.

Due to this, it becomes possible to advance far enough in the study of algorithmic constructions. In addition to traditional forks and two types of loops (with a precondition and with a given number of repetitions), input commands, subprograms and parameters are studied.

The methodic were prepared as part of the Perm version of the propaedeutic course of informatics (team of authors: M. A. Plaksin, N. G. Ivanova, O. L. Rusakova; the working title of the course is "TRIZformatika"). But the technique can be applied in any other course.

The article provides a description and playing fields of four board games for mastering the mechanism of flow charts, and also gives a number of methodological comments.

Keywords: informatics, secondary school, primary school, teaching methods, flow chart, algorithmic constructions, board game, TRIZformatika, TRIZformatika.

Предлагаемые методические материалы подготовлены в рамках работ над Пермской версией пропедевтического курса информатики (авторский коллектив: М. А. Плаксин, Н. Г. Иванова, О. Л. Русакова; рабочее название курса — «ТРИЗформатика»). Но методика может применяться в любом другом курсе.

В статье приведены описание и игровые поля четырех игр-ходилок для освоения механизма блок-схем, а также дан ряд методических замечаний.

Ключевые слова: информатика, средняя школа, начальная школа, методика обучения, блок-схема, алгоритмические конструкции, игра-ходилка, ТРИЗформатика.

1. Введение

Стремительное развитие информационных технологий требует постоянного обновления школьного курса информатики и включения в него нового материала. При этом старый материал в значительной мере должен сохраниться. То есть речь идет о содержательном расширении курса без увеличения отводимых на него часов. Сделать это можно только за счет интенсификации образовательного процесса, только научившись давать учащимся за то же время больше учебного материала, научившись прежнее содержание излагать за меньшее количество уроков. Это актуально для всех школьных предметов. Но для информатики с ее чрезвычайно высокими темпами развития — особенно.

Настоящая статья предлагает для изучения такой традиционной темы, как представление алгоритмов в виде блок-схем, использовать нетрадиционный прием — игры-ходилки. Игровое поле в такой игре представляет собой блок-схему. В ходе игры учащиеся бросают кубик и передвигают фишку по игровому полю в зависимости от выпавшего числа. Но при этом они должны соблюдать правила исполнения входящих в блок-схему алгоритмических конструкций.

Цель предлагаемого подхода — интенсифицировать освоение учебного материала.

Для игры класс разбивается на команды по два—четыре человека. Все команды играют одновременно. В ходе игры каждый участник многократно проходит все изученные алгоритмические конструкции. Таким образом достигается стопроцентный охват учащихся и многократное повторение материала.

За счет этого появляется возможность продвинуться в изучении алгоритмических конструкций достаточно далеко. А именно: кроме традиционных развилок и двух видов циклов (с предусловием и с заданным числом повторений) изучаются команды ввода, подпрограммы (вспомогательные алгоритмы) и параметры.

Несколько слов о параметрах. При обсуждении этого понятия в примерах возникают параметры как входные, так и выходные (например, подпрограмме сложения многозначных чисел нужен выходной параметр для передачи результата). Однако внимание на этом не акцентируется. Про деление параметров на входные и выходные и способы передачи параметров ничего не говорится. В играх используются только входные параметры.

Знакомство с механизмом подпрограмм, причем параметризованных, позволит в будущем рассматривать достаточно сложные алгоритмы и плавно перейти к пропедевтике технологии программирования (наращивание мощности простых исполнителей за счет разработки мощных команд-подпрограмм, проектирование алгоритмов «сверху вниз», построение алгоритмов из крупных заготовок).

В данный момент от учащихся не требуется построения блок-схем, нужны только чтение и исполнение алгоритмов. Построение алгоритмов — материал более поздний. (Почему-то при изучении естественных языков кажется нормальным, что ребенок сначала научается читать, а потом в течение многих лет в школе осваивает письмо. А вот при изучении алгоритмики ребенку сразу предлагают писать, хотя он еще читать не научился.)

Предлагаемые методические материалы подготовлены в рамках работ над Пермской версией пропедевтического курса информатики (авторский коллектив: М. А. Плаксин, Н. Г. Иванова, О. Л. Русакова; рабочее название курса — «ТРИЗформатика») [9, 10]. Но применение данной методики с курсом ТРИЗформатики никак не связано.

Далее приводятся описание и игровые поля четырех игр-ходилок для освоения механизма блок-схем и еще несколько методических замечаний.

2. Правила игры «Блок-схема-Бродилка»

«Блок-схема-Бродилка» — игра-ходилка. Цель игры — переместить фишку по шагам от начала до конца блок-схемы. Количество шагов на каждом ходе определяется бросанием кубика. Развилки определяются правилами записи блок-схем: условие (ветвление и цикл ПОКА), цикл ПОВТОРИТЬ.

Блоки перенумерованы. Номера играют чисто информационную роль, служат «именами» блоков и предназначены для однозначного определения позиции.

Играть может любое количество игроков. Разумное число — от двух до четырех.

Каждому игроку полезно завести карточку для ведения записей в ходе игры.

Каждый игрок двигает свою фишку, отличную от других.

Первоначально все фишки ставятся на блок «Начало».

Игроки по очереди бросают кубик и передвигают свою фишку по блокам на выпавшее количество шагов.

Если на пути встречается заголовок цикла ПОВТОРИТЬ (блок-шестиугольник), то фишка должна зайти в тело цикла и повторить цикл указанное число раз. Игрок должен отслеживать (отмечать на своей карточке), сколько повторов цикла он уже выполнил.

Если на пути встречается блок условия (блок-ромб), то число, выпавшее на кубике (количество шагов на данном ходе), подставляется в записанное в ромбе условие, которое и определяет направление дальнейшего движения. Если фишка попала на блок условия на последнем шаге данного хода, то направление движения будет определяться следующим броском кубика (на следующем ходе).

На рисунке 1 приведено игровое поле игры «Бродилка-1». Рассмотрим пример игры.

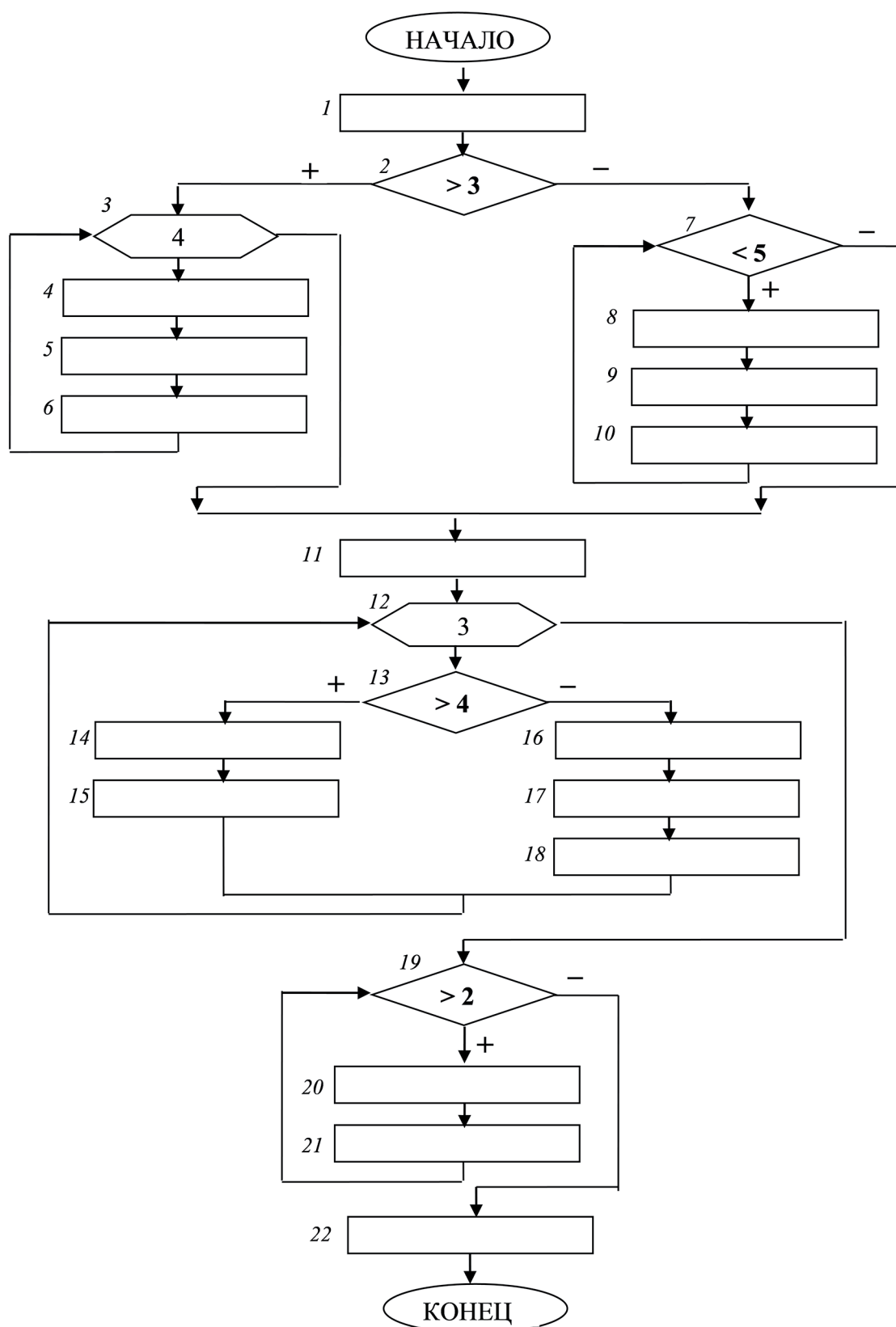


Рис. 1. Игровое поле (блок-схема) игры «Бродилка-1»

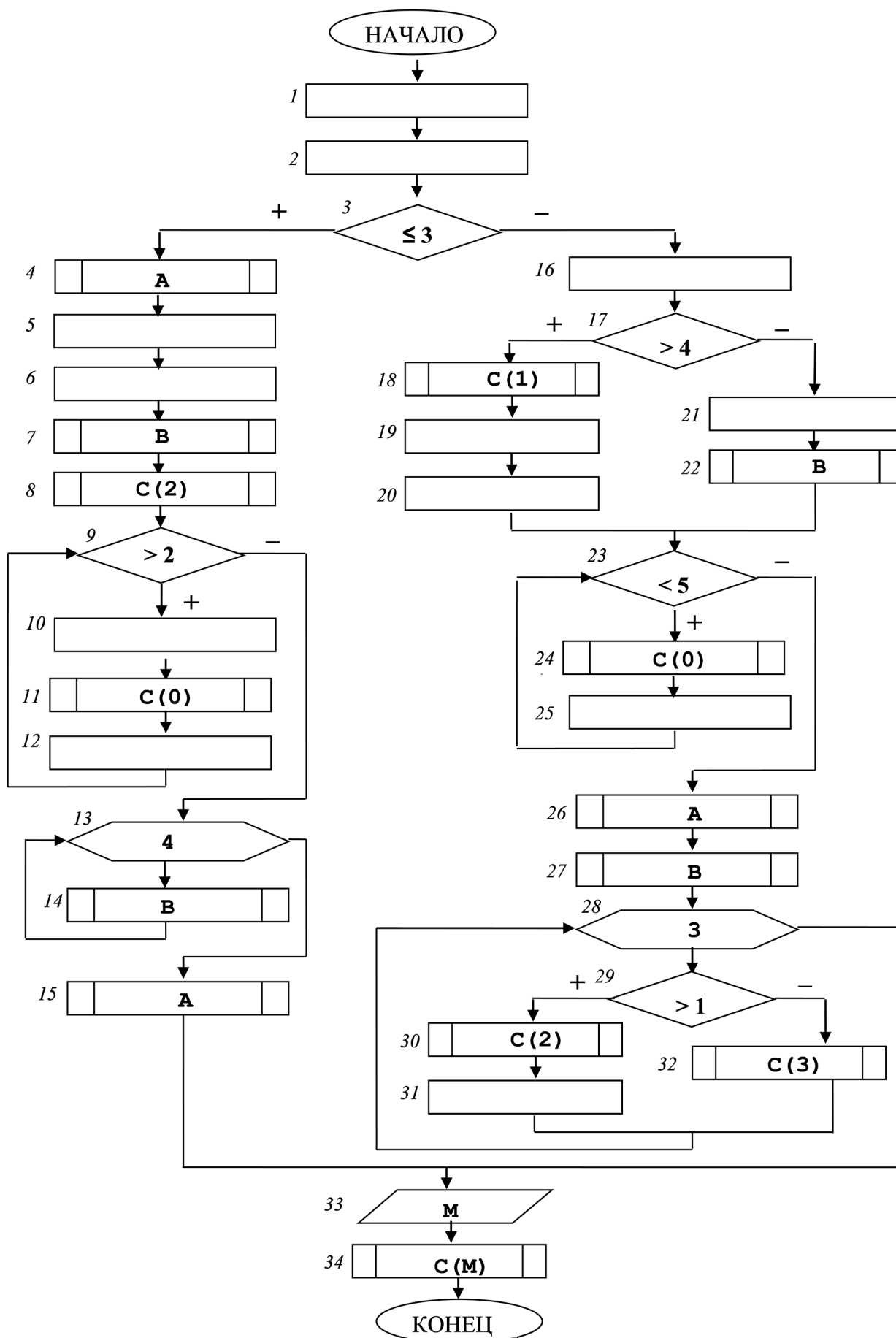


Рис. 2. Игровое поле игры «Бродилка-2». Блок-схема главной программы

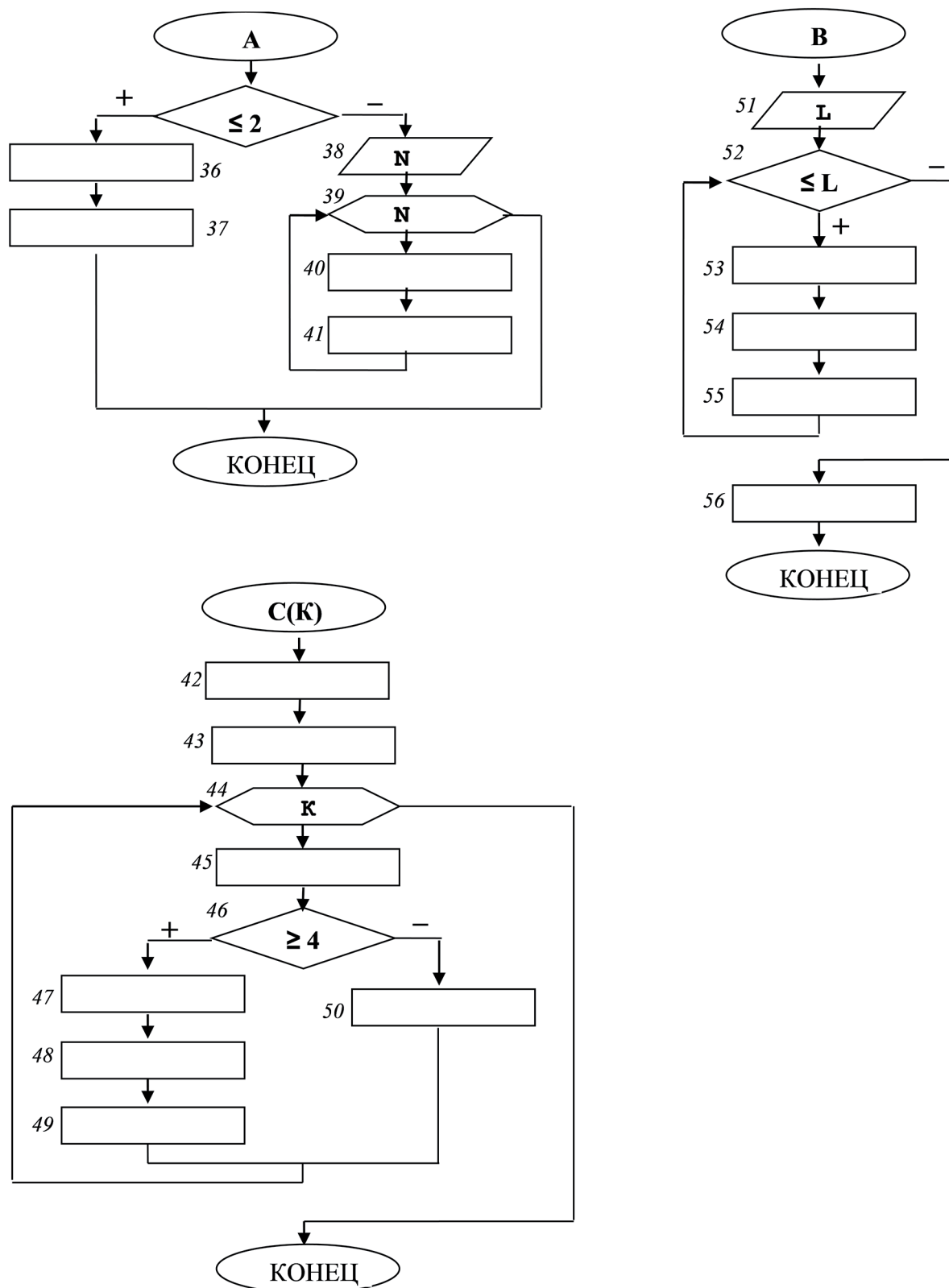


Рис. 3. Игровое поле игры «Бродилка-2». Блок-схемы подпрограмм

Пусть при первом броске на кубике выпало число 3. Игрок передвигает фишку по блоку 1 и доходит до блока 2. В блоке 2 стоит условие « > 3 ». В это условие надо подставить число, выпавшее на кубике, т. е. 3. Получится условие « $3 > 3$ ». Это условие ложно, поэтому движение продолжается по ветке «минус». Третьим шагом фишка встанет на блок 7. Это блок-условие. Фишка попала на него на последнем шаге данного хода. Право хода переходит к другому игроку. Направление движения данной фишки будет определено только на следующем ходе. Если на кубике выпадет меньше 5, фишка должна войти в цикл (на блок 8). Если выпадет больше или равно 5, фишка должна пойти на блок 11 и далее.

Продолжение правил для игры «Бродилка-2».

Если на пути встречается блок вызова подпрограммы (блок-прямоугольник с двойными стенками), то игрок должен увести свою фишку на блок-схему названной подпрограммы, пройти ее от начала до конца и после этого продолжить выполнение с блока, следующего за блоком вызова подпрограммы.

Если подпрограмма параметризована, игроку имеет смысл записать значение параметра на свою карточку. (Подпрограммы вызываются по несколько раз с разными значениями параметров.)

Последний вид блока, который встречается на блок-схеме, это блок ввода данных (блок-параллелограмм). В параллелограмме указано имя переменной, значение которой надо ввести. Для этого игрок бросает кубик. Выпавшее значение становится значением этой переменной и используется в следующем блоке. Имя и значение переменной игрок записывает на свою карточку. Переменная может использоваться для сравнения в блоке-условии, для определения количества повторов цикла ПОВТОРИТЬ и для передачи в подпрограмму в качестве параметра. Обратим внимание, что ввести значение переменной надо даже в том случае, если фишка просто прошла через блок ввода данных, не остановившись на нем. Дело в том, что введенное значение будет задействовано в следующих блоках.

На рисунках 2 и 3 приведено игровое поле игры «Бродилка-2» (на рисунке 2 — блок-схема главной программы, на рисунке 3 — блок-схемы подпрограмм).

3. Правила игры

«Блок-схема-Вычислялка»

«Блок-схема-Вычислялка» — игра-ходилка. Игровое поле представляет собой блок-схему. Цель игры — переместить фишку по шагам от начала до конца блок-схемы. Количество шагов на каждом ходе определяется бросанием кубика. Развилки определяются правилами записи блок-схем: условие (ветвление и цикл ПОКА), цикл ПОВТОРИТЬ.

Играть может любое количество игроков. Разумное число — от двух до четырех.

Каждый игрок должен завести карточку для ведения записей в ходе игры.

Каждый игрок двигает свою фишку, отличную от других.

Первоначально все фишки ставятся на блок «Начало».

Игроки по очереди бросают кубик и передвигают свою фишку по блокам на выпавшее количество шагов.

По ходу игры будут использоваться «переменные величины». Каждая такая величина имеет имя (в этой игре оно обозначается одной латинской буквой) и значение. Значение величины может изменяться по ходу игры (на то она и переменная!). У каждого игрока есть карточка, на которой обязательно записывается значение «величины N». Остальные переменные будут записываться в том случае, когда фишка игрока проходит через блок ввода данных, в котором записано имя соответствующей переменной.

Перед началом игры игрок бросает кубик. Выпавшее число является начальным значением величины N этого игрока.

В блоках-действиях записаны выражения, состоящие из знаков $+/-$ и числа.

В блоках-условиях записаны выражения, состоящие из N, знака сравнения и числа.

Если при движении по игровому полю фишка останавливается на блоке-действии, то указанное действие надо выполнить над величиной N (прибавить к N или отнять от N указанное число). Результат операции становится новым значением числа N.

Если на пути встречается заголовок цикла ПОВТОРИТЬ (блок-шестиугольник), то фишка должна зайти в тело цикла и повторить цикл указанное число раз. Игрок должен отслеживать (отмечать на своей карточке), сколько повторов цикла он уже выполнил.

Если при движении фишка проходит через блок-условие (блок-ромб), то величина N подставляется в операцию сравнения и направление движения определяется результатом сравнения.

На рисунке 4 приведено игровое поле игры «Вычислялка-1». Рассмотрим пример игры.

Пусть начальное значение N, определенное броском кубика, равно 5.

Пусть при первом броске на кубике выпало число 6. Игрок передвигает фишку по блоку 1 и доходит до блока 2. В блоке 2 стоит условие « $N < 10$ ». В это условие надо подставить значение переменной N, т. е. 6. Получится условие « $6 < 10$ ». Это условие истинно, поэтому движение продолжается по ветке «плюс». То есть фишка входит в цикл. Третьим и четвертым шагами фишка проходит блоки, составляющие тело цикла (« $+3$ » и « -1 »). И пятым шагом опять оказывается на заголовке цикла — блоке условия « $N < 10$ ». Поскольку значение переменной N не изменилось, фишка на шестом шаге опять входит в цикл и заканчивает движение на первом блоке тела цикла « $+3$ ». Это блок-действие. Это действие — « $+3$ » — выполняется с переменной N. Ее значение увеличивается на 3 и становится равно 8. На этом ход данного игрока заканчивается. Право броска кубика переходит к следующему игроку.

Продолжение правил для игры «Вычислялка-2».

Если на пути встречается блок вызова подпрограммы (блок-прямоугольник с двойными стенками), то игрок должен увести свою фишку на блок-схему названной подпрограммы, пройти ее от начала до конца и после этого продолжить выполнение с блока, следующего за блоком вызова подпрограммы.

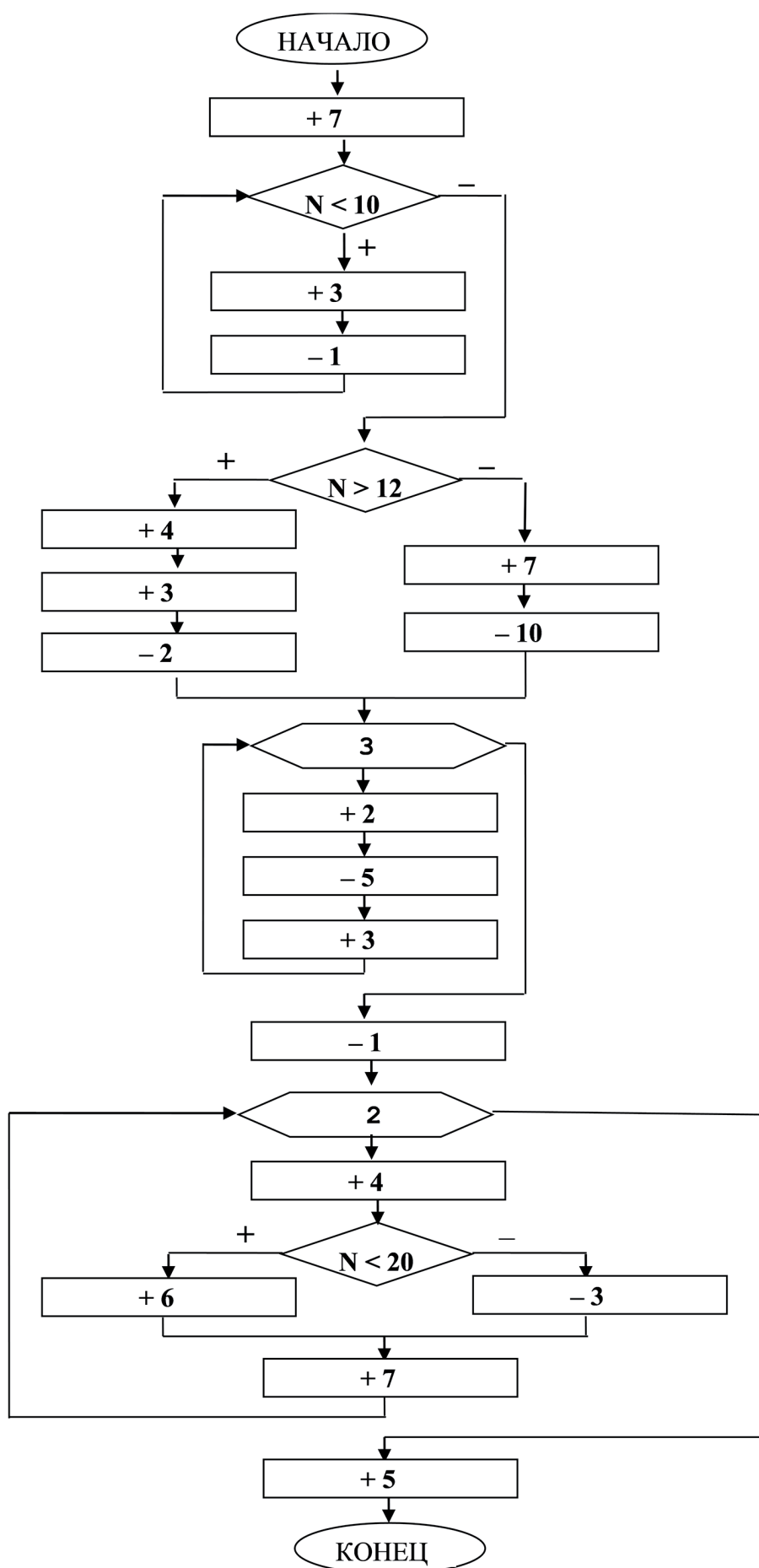


Рис. 4. Игровое поле (блок-схема) игры «Вычислялка-1»

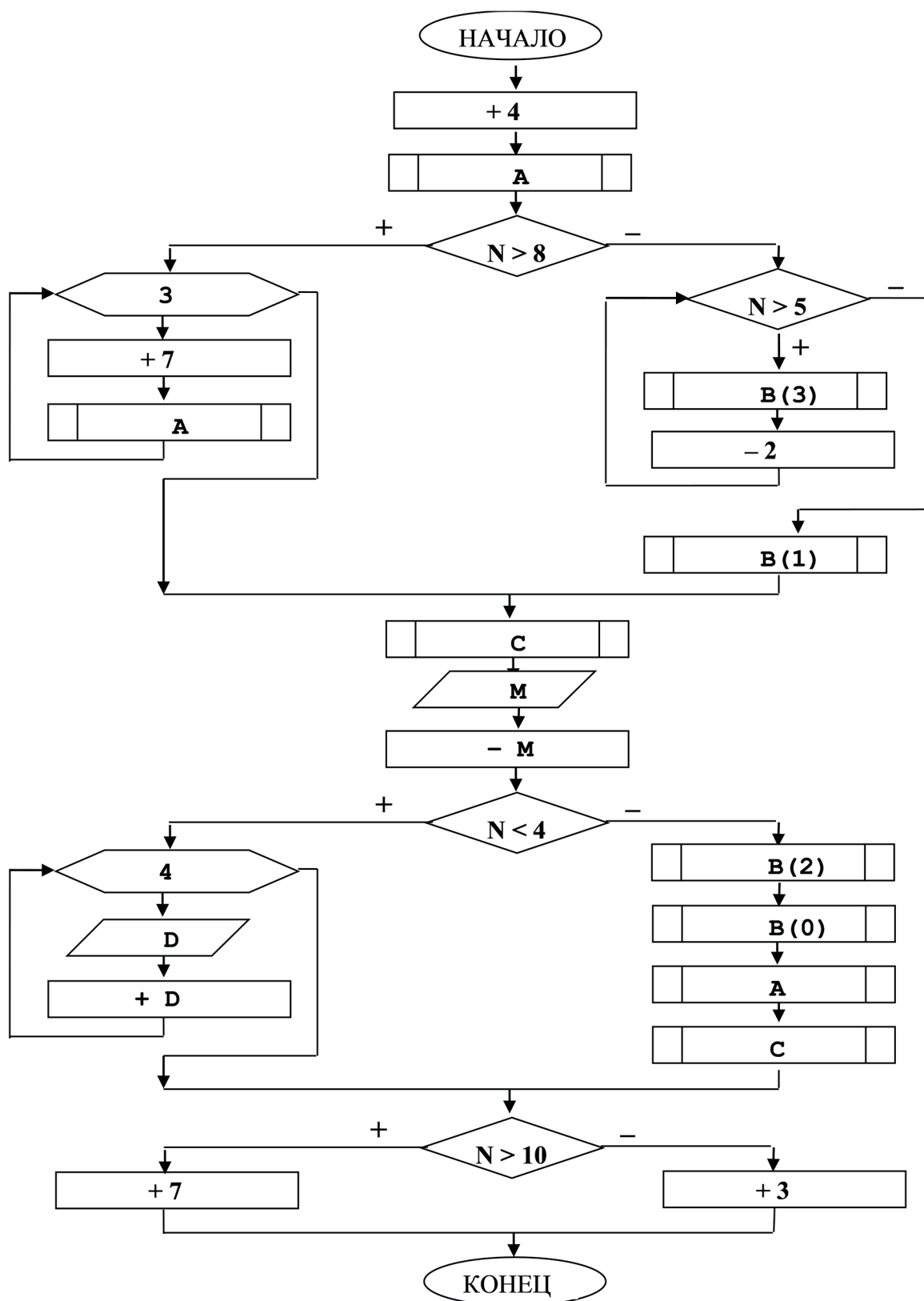


Рис. 5. Игровое поле игры «Вычислялка-2». Блок-схема главной программы

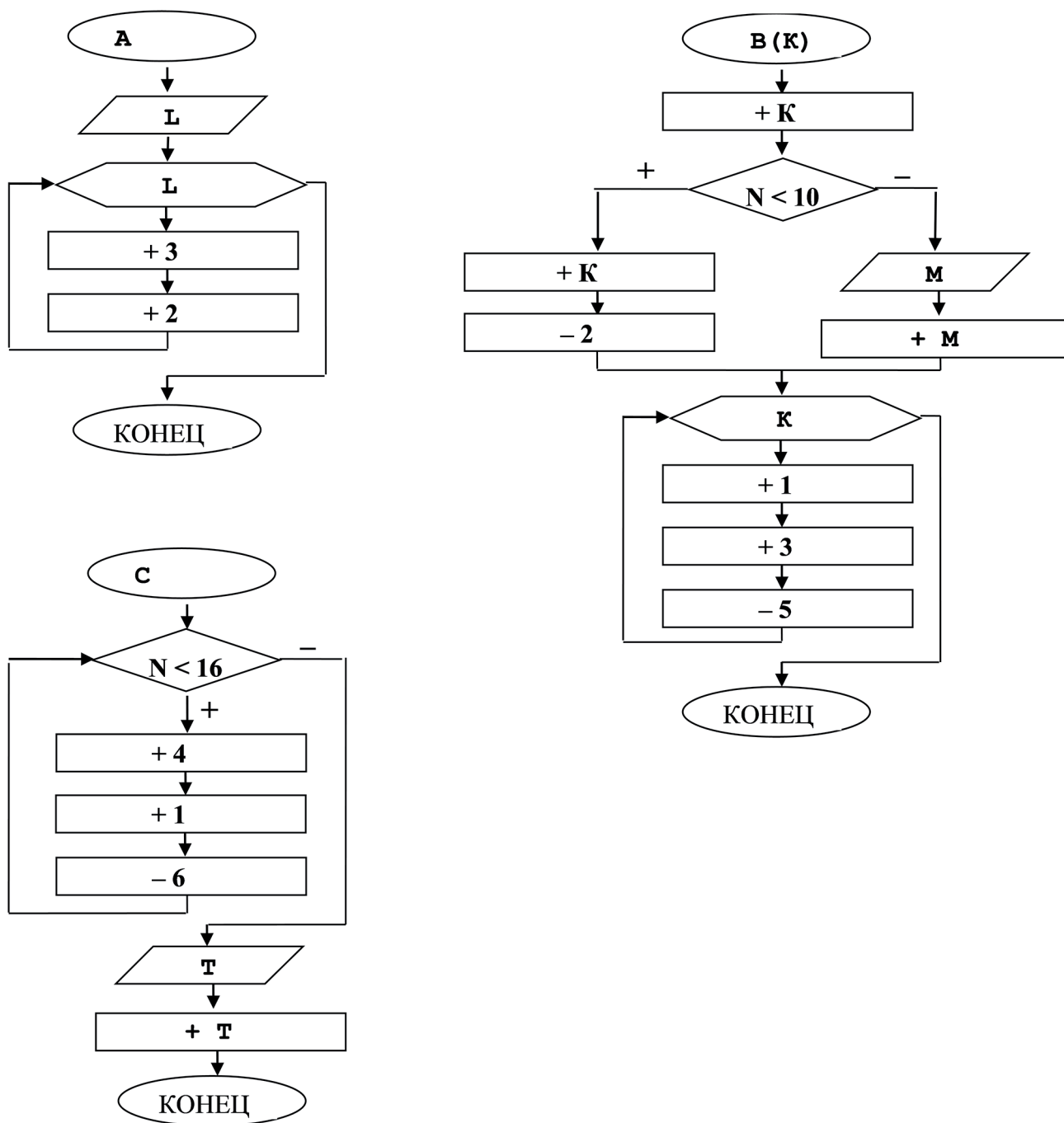


Рис. 6. Игровое поле игры «Вычислялка-2». Блок-схемы подпрограмм

Если подпрограмма параметризована, игроку имеет смысл записать значение параметра на свою карточку. (Подпрограммы вызываются по несколько раз с разными значениями параметров.)

Последний вид блока, который встречается на блок-схеме, это блок ввода данных (блок-параллелограмм). В параллелограмме указано имя переменной, значение которой надо ввести. Для этого игрок бросает кубик. Выпавшее значение становится значением этой переменной и используется в следующем блоке. Имя и значение переменной игрок записывает на свою карточку. Переменная может использоваться для вычисления нового значения переменной N в блоке-действии, для сравнения в блоке-условии, для определения количества повторов цикла ПОВТОРИТЬ и для передачи в подпрограмму в качестве параметра. Обратим внимание, что ввести значение переменной надо даже в том случае, если фишка просто прошла через блок ввода данных, не остановившись на нем. Дело в том, что введенное значение будет задействовано в следующих блоках.

На рисунках 5 и 6 приведено игровое поле игры «Вычислялка-2» (на рисунке 5 — блок-схема главной программы, на рисунке 6 — блок-схемы подпрограмм).

4. Методические замечания

4.1. Разбивка материала по урокам

Всего на тему «Представление алгоритмов с помощью блок-схем» планируется четыре часа:

- Первый урок — «Знакомство с видами алгоритмов, ветвлениями и циклами. Игра “Бродилка-1”». О заикливании речь пока не идет.
- Второй урок — «Знакомство с конструкциями ввода, подпрограммами и параметрами. Продолжение работы с управляющими конструкциями. Игра “Бродилка-2”». (Пока тоже без заикливания.)
- Третий урок — «Понятие заикливания на примере подпрограммы В из игры “Бродилка-2”». Закрепление материала. Игры “Вычислялка-1”, “Вычислялка-2”».
- Четвертый урок — «Повторение. Проверочная работа».

Начинать надо с той игры, которая проще. К сожалению, универсального способа оценить сложность игр не существует. Выше предложено начинать с «Бродилки». Но автор встречался с ситуацией, когда «Вычислялка» оказывалась для аудитории проще.

4.2. Использование «рукописных» игровых полей

Автор проводил занятия с детьми на игровых полях, нарисованных от руки. Специально для данной публикации (для лучшего отображения материала при типографской печати) были подготовлены «машинописные» игровые поля, выполненные средствами компьютерной графики. Автор настоятельно рекомендует использовать для занятий с детьми поля «рукописные» (рис. 7). Причина — та же, по которой рукописные конспекты лекций предпочтительней машинописных. «Рукописные»

поля — «живые». С ними интересно. Каждая блок-схема, каждый блок уникальны, неповторимы. Это позволяет при освоении информации активно задействовать правое полушарие головного мозга (отвечающее за конкретно-образное мышление). «Рукописные» блок-схемы «хорошо видно». При использовании «рукописных» полей эффективность работы возрастает.

«Машинописные» блок-схемы, разумеется, много красивее. Но они — «мертвые», неинтересные. Все они выглядят одинаково. От однообразных изображений попросту рябит в глазах. Воспринимать их приходится левым — абстрактно-логическим — полушарием. Правое используется минимально. С такими блок-схемами дети играть не захотят.

В данном случае мы имеем дело с часто встречающимся противоречием: эффектность или эффективность? Что важнее: красивый вид или доходчивость учебного материала?

Оба комплекта игровых полей — «машинописный» и «рукописный» — можно скачать как дополнительный материал к данной статье на сайте ИНФО.

4.3. Логика исполнения алгоритма vs логика игры-ходилки

В правилах обеих игр возникает противоречие между «логикой игры-ходилки» и «логикой исполнения алгоритма». По логике игры-ходилки «срабатывать» должен только тот блок, на котором фишка остановилась. Блок, через который фишка прошла без остановки, «срабатывать» не должен. Это значит, например, что если фишка не остановилась на блоке вызова подпрограммы, а просто прошла через этот блок, то и заходить в эту подпрограмму она не должна. С точки зрения логики исполнения алгоритма деление продвижения фишки по блок-схеме на ходы — вещь совершенно условная. Поэтому «срабатывать» должен каждый блок, через который проходит фишка. В описанных выше правилах игры вход в подпрограмму выполняется всегда. То есть в этом случае применяется логика исполнения алгоритма. Аналогично всегда выполняется ввод данных при прохождении фишки через блок ввода (введенное значение может потребоваться далее). В то же время в «Вычислялке» предлагается выполнять не все вычислительные блоки (действия с переменной N), а только те, на которых фишка остановится. В этом случае применяется логика игры-ходилки.

4.4. Пропедевтика трассировки переменных

Запись последовательности значений переменной N в игре «Вычислялка», по сути, является трассировкой. Впоследствии ее можно будет вспомнить при изучении трассировочных таблиц.

4.5. Сравнение циклов ПОКА и ПОВТОРИТЬ

В игре «Бродилка-1» дается возможность сравнить циклы с предусловием и с заданным числом повторений. В первой развилке (блок 2) на ветви «плюс» располагается цикл ПОВТОРИТЬ (блоки 3–6), а на ветви «минус» — цикл ПОКА (блоки 7–10). У обоих циклов тело имеет одинаковую длину. В первом цикле количество повторений задано точно — 4. Во втором стоит условие «Пока < 5» (т. е. для ответа «Да» на кубике могут выпасть

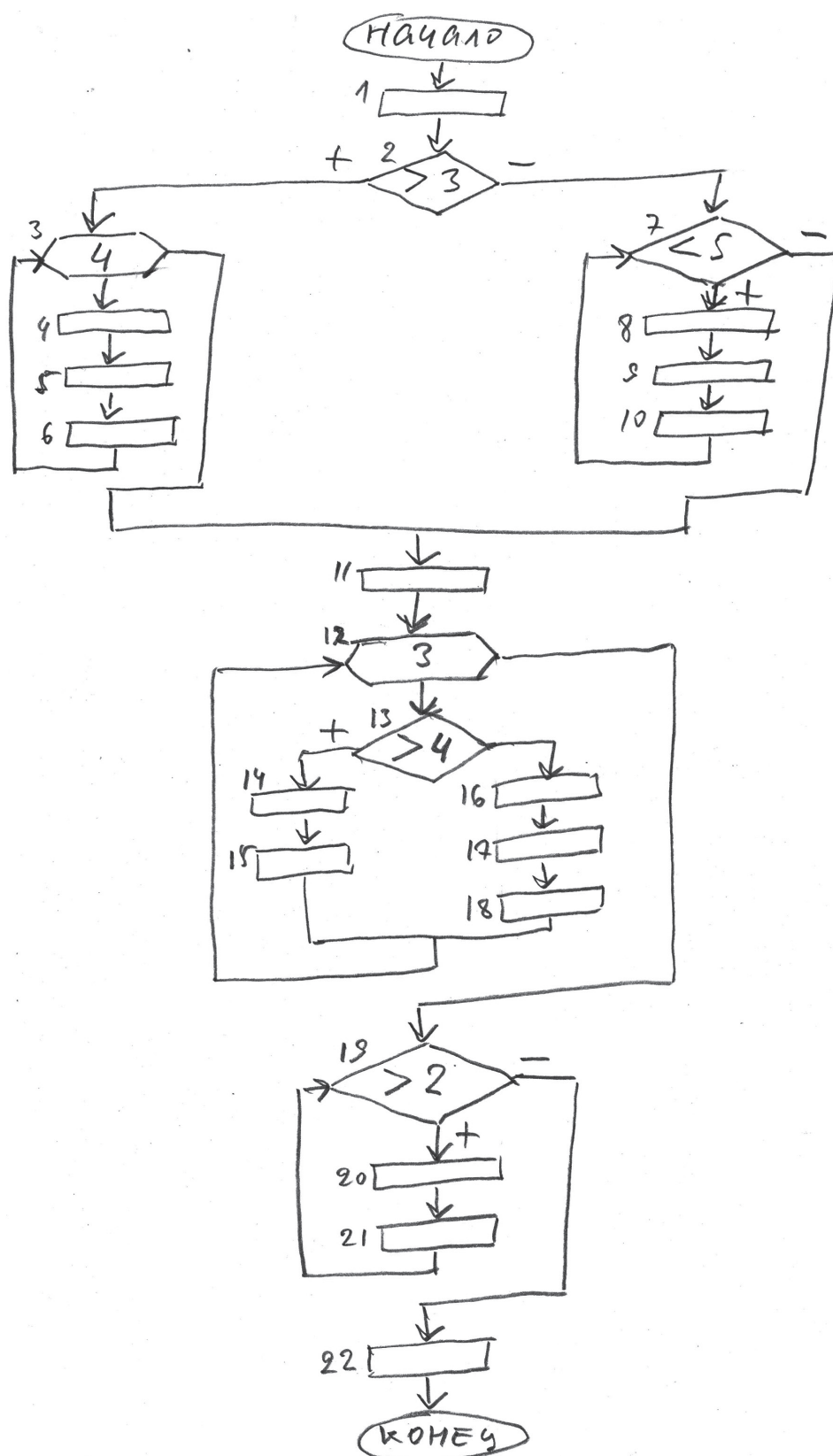


Рис. 7. Игровое поле (блок-схема) игры «Бродилка-1»

числа от 1 до 4). Можно проследить отличие в исполнении этих циклов.

4.6. Зацикливание

В игре «Бродилка-2» в подпрограмме В нарочно допущена возможность зацикливания. Заголовок цикла имеет вид «ПОКА значение на кубике $\leq L$ », где значение переменной L определено предыдущим броском кубика. Если L выпадает равным 6, выйти из цикла невозможно. Хорошо, если такая ситуация реально возникнет в игре. На нее надо обратить внимание и ввести понятие «зацикливание». Если ситуация не возникнет «естественным» путем, следует специально обратить внимание на ее возможность.

Вообще говоря, понятие «зацикливание» в данный момент выглядит некоторым излишеством. Нет никакой необходимости изучать его именно сейчас. Вполне можно отложить знакомство на потом.

4.7. Отсутствие вложенных подпрограмм

Для простоты в описанных играх отсутствуют вложенные вызовы подпрограмм. Подпрограммы вызываются только из главной программы. Обучение манипуляциям со стеклом не входит в задачу данных игр.

4.8. Варьирование правил игры

Предлагаемые правила игр являются результатом нескольких проб, проведенных как на ученической, так и на учительской аудитории. Автор считает полезным поделиться описанием опробованных вариантов. Обсуждались следующие идеи.

Не считать за шаги блоки-ветвления и заголовки циклов, считать только блоки-действия. Опыт показал, что необходимость разделять блоки на те, которые надо считать, и на те, которые считать не надо, излишне усложняет игру. Проще считать все блоки подряд.

Был вопрос: надо ли продолжать движение после входа в подпрограмму, если на данном ходе еще остались неизрасходованные шаги, или движение в подпрограмме надо начинать со следующего броска кубика? Вариант «продолжать движение» проще.

Значительные сложности были связаны с прохождением условий в игре «Бродилка». Один из операндов указан в блоке-условии. Откуда взять второй, который должен принести в игру случайность? Кроме варианта, описанного выше, рассматривались следующие:

- 1) Игрок бросает сразу два кубика: «кубик-шагалку» и «кубик-решалку». Кубик-шагалка определяет количество шагов, на которые должна быть передвинута фишка. Кубик-решалка задает число, которое будет подставлено в блок-условие.
- 2) Игра ведется одним кубиком. Его первоначальный бросок определяет количество шагов. При попадании фишки на блок-условие игрок бросает кубик еще раз. Выпавшее число подставляется в условие. Этот вариант делает игру существенно сложнее.
- 3) Перед началом игры разыгрывается на кубике и записывается ряд случайных чисел. Далее они по очереди будут использоваться в блоках-условиях. При исчерпании ряда его можно либо продолжить, либо зациклить.

4.9. Необходимость специальных мер для контроля знаний

Слабым местом игровой методики является сложность контроля знаний. В тот момент, когда играет весь класс, контроль правильности действий каждого учащегося обеспечить физически невозможно. Может возникнуть ситуация, когда *все* игроки одной команды неправильно поняли какую-то конструкцию либо неправильно понял кто-то один, но этот один сумел убедить в своей правоте всех остальных. В результате окажется, что вся команда затвердила неправильное понимание. Для проверки правильности понимания необходимы домашние задания, причем легкопроверяемые. Лучше всего, если это будут блок-схемы алгоритмов черчения, которые надо выполнить (создать соответствующий чертеж). В этом случае правильность ответов можно оценить «с одного взгляда». В данный момент учащимся еще не введены строго (хотя уже упомянуты) понятия «исполнитель» и «система команд исполнителя» (СКИ). Поэтому углубляться в их строгое описание не стоит.

Самую первую апробацию описанные игры прошли в лицее № 10 города Перми (см. фото). Больше фотографий — в архиве дополнительных материалов к данной статье, который можно скачать на сайте ИНФО.



Фото. Игры-бродилки на уроке информатики в лицее № 10 города Перми

5. Заключение

В заключение несколько слов об оценке оригинальности описанных игр.

Сама по себе идея использования игровых моментов в изучении информатики ведет начало от Сеймура Пейперта с его черепахой Лого. Затем она была подхвачена и реализована в превеликом множестве ползающих, шагающих и прыгающих исполнителей. Вслед за многочисленными версиями Лого (см., например, [7]) были Таркан [4], Муравей [3], Кенгуренок [2]. Предназначались они для освоения алгоритмики. В настоящее время это направление развивается очень активно. Перечислять надо уже не отдельные игры и исполнителей, а обзоры таких игр и исполнителей (см., например, [1, 5, 6, 8, 11]).

Описанные в данной статье игры придуманы в 2013 году и с тех пор многократно использовались при работе как с учащимися, так и учителями информатики. Аналогов автор пока не встречал. Игры оригинальны, во-первых, по теме (освоение блок-схем), во-вторых, по направленности на «чтение алгоритмов», а не их написание, в-третьих, по «настойной реализации». Традиционно считается, что игры для обучения информатике должны иметь компьютерную реализацию. Исключения — редки. Признавая достоинства компьютерных игр, автор считает, что современные дети слишком много времени проводят в мире виртуальном. И любую возможность вернуть их в мир материальный и заставить поработать руками надо использовать.

Список источников

1. *Бордик В.* 15 бесплатных игр и сайтов, которые научат ребенка программировать // Летидор. <https://letidor.ru/obrazovanie/15-besplatnykh-igr-i-saitov-kotorye-nauchat-rebenka-programmirovat.htm>
2. *Вараксин Г. С.* Информатика в начальной школе. Кенгуренок (графический исполнитель). <https://ikt-det.jimdofree.com/алгоритмика/исполнители/графический-исполнитель-кенгуренок/>
3. *Гутман Г. М., Карпилова О. М.* Муравьиные сказки. М.: Просвещение, 1993. 98 с.
4. *Дуванов А. А., Первин Ю. А.* Необыкновенные приключения Пети Кука в Роботландии. М.: Финансы и статистика, 1997. 272 с.
5. Игры с алгоритмами // POLYCENT. <https://polycent.ru/blog/igryi-s-algoritmami/>
6. *Карпов Д.* 15 сайтов и игр, которые научат детей программированию. Почему и как стоит учить информатику с дошкольного возраста // Мел. https://mel.fm/ucheba/fakultativ/5297184-computer_science
7. ЛогоМиры 3.0. Интегрированная творческая среда для освоения азов программирования. <https://www.softslot.com/software-2581-logomiry.html>
8. *Мартинов А.* 10 игр, которые научат ребенка программированию // Лайфхакер. <https://liferhacker.ru/10-igr-nauchat-programmirovat/>
9. *Плаксин М. А.* ТРИЗформатика — метапредмет, объединяющий компьютерные и интеллектуальные технологии работы с информацией (ответ на вызов информационного общества) // Программирование. 2011. № 6. С. 26–32.
10. *Плаксин М. А., Иванова Н. Г., Русакова О. Л.* Информатика: учебник для 4 класса. М.: БИНОМ. Лаборатория знаний, 2012. 128 с.
11. ТОП-12 игр, с которыми ребенок научится программировать // Proglib. <https://proglib.io/p/games-for-kids>

DOI: 10.32517/2221-1993-2022-21-3-49-49

Александра Алексеева

дипломант Всероссийского конкурса «Школьная информатика: вчера, сегодня, завтра»

в номинации «Школьная информатика: заглянем в будущее»,

средняя общеобразовательная школа № 567 Петродворцового района Санкт-Петербурга, Россия

ПУТЕШЕСТВИЕ В ШКОЛУ БУДУЩЕГО

Я решила заглянуть в будущее, чтобы увидеть, как будет выглядеть школа через несколько лет.

Итак, поехали!

При входе в школу меня встретил аппарат — он подтвердил, что я являюсь ученицей этой школы, а также измерил температуру и сообщил, что я здорова и могу присутствовать на уроках. И заодно спросил, как у меня дела! Я уже была удивлена. Что же будет дальше?

Затем я пошла в гардероб — там у каждого ученика есть свой личный шкафчик для вещей, а чтобы его открыть, надо отсканировать свою ладонь, и если это твой шкафчик, то он откроется. Вот это да!

После посещения гардероба я посмотрела расписание уроков в своем дневнике — этот дневник появляется в воздухе в виде маленького светового экрана, когда ты нажимаешь на кнопочку на специальном браслете. В дневник можно «записать» домашнее задание, просто произнеся его в специальный микрофон на браслете.

Так... Смотрим расписание — первым уроком у меня физическая культура. Интересно, что меня встретит в спортивном зале?

При входе в спортивный зал специальный аппарат проверяет наличие спортивной формы. Как хорошо, что я ее сегодня не забыла! В физкультурной раздевалке, как и в гардеробе при входе в школу, у каждого есть свой шкафчик, который тоже открывается с помощью сканирования ладони.

Всё, я надела спортивную форму и готова идти на урок! Вышла из раздевалки и... Ой! А как же так? Где учитель физкультуры? А он, оказывается, сидит в тренерской комнате и ждет, пока каждый ученик подойдет к специальному экрану на стене и подтвердит свое присутствие на уроке. И такой экран есть в каждом учебном кабинете — он передает информацию в электронный журнал, облегчая работу учителя.

Что ж, все одноклассники подтвердили свое присутствие на уроке, вышел учитель, и мы начали заниматься.

В целом урок проходил как обычно, но после каждого выполненного упражнения браслет измерял температуру тела, пульс, а также, если было нужно, сообщал,

что необходимо выпить воды. Вот так здесь заботятся о самочувствии учащихся!

Урок закончился, браслет проверил состояние моего здоровья, я переоделась в школьную форму, попила воды и пошла на следующий урок.

Расписание в браслете подсказало, что вторым уроком у меня будет русский язык. Очень интересно, как же выглядит класс, в котором преподают этот предмет.

Заходим в класс, подтверждаем свое присутствие на уроке и садимся за парты. Пока что ничего необычного. Но тут прозвенел звонок, учитель вошел в класс, и...

Ого! Вместо тетрадей и ручек все достают планшеты и стилусы, чтобы писать на экране планшета! Вместо обычной меловой доски слова учителя выводятся на большом экране в центре класса, но самое интересное — учитель ничего не записывает на доске, он только произносит свою речь, а его слова сами отображаются на экране.

Так же прошли еще два урока: математика и литература.

Остались география и информатика, последнюю я ждала больше всех остальных уроков!

На географии в основном все было так же, как на привычном нам уроке, но всем знакомой карты полушарий Земли не было — вместо нее висел интерактивный экран с картой, и на нем можно было рисовать, писать и делать многое другое, чего нельзя делать на обычной карте.

И вот, наконец, начался урок информатики.

Вместе с другими учениками я вошла в класс и попала в совершенно иную реальность! Всё светится, издает разные звуки, мигает... Места учеников оборудованы новейшими компьютерами, которые очень сильно отличаются от тех, которые мы привыкли видеть в школьных кабинетах информатики.

Учитель дал задание, и ученики начали его выполнять, а для тех, кто не запомнил или не расслышал, задание вывелось на доске, очень здорово!

Вот такое необычное путешествие в будущее у меня получилось. И мне оно очень понравилось!

Надо будет как-нибудь его повторить, но не одной, а с моими друзьями, им тоже будет очень интересно. До новых встреч!

DOI: 10.32517/2221-1993-2022-21-3-50-50

Снежана Жумашева

*дипломант Всероссийского конкурса «Школьная информатика: вчера, сегодня, завтра»
в номинации «Школьная информатика: заглянем в будущее»,
Оренбургский учетно-финансовый техникум, Оренбург, Россия*

ЗАВТРА — УЖЕ СЕГОДНЯ

Сегодня разнообразные гаджеты — смартфон, планшет, ноутбук, настольный компьютер — присутствуют в жизни практически каждого человека. В школах также используются компьютеризированные средства, которые помогают легче освоить учебный материал. Помимо уже перечисленных гаджетов это, например, интерактивные доски, очки виртуальной реальности, 3D-принтеры и т. д. Невозможно представить себе современную школу без широкого применения информационных технологий, а глубокое освоение этих технологий — без изучения информатики, которая за 37 лет своего существования в советской, а затем российской школе стала одним из ведущих учебных предметов в системе образования. Информатика — это такая учебная дисциплина, которая никогда не стоит на месте: вслед за непрерывным появлением новых технологий, новых средств обработки и передачи информации развивается, совершенствуется, постоянно меняется и школьный предмет «Информатика».

В XXI веке информационные технологии присутствуют в нашей жизни повсеместно, а значит, знания, полученные при изучении информатики, тоже нужны нам всегда и везде.

Например, у меня дома есть телевизор, который подключен к интернету, на нем я смотрю фильмы. Домашний компьютер с выходом в интернет позволяет с помощью поисковых систем, таких как Яндекс или Google, найти что угодно — не надо идти в библиотеку и среди множества книг искать нужную информацию, достаточно просто ввести запрос в поисковой системе, и она за секунду выдаст все необходимое. Изучая роман-эпопею Л. Н. Толстого «Война и мир», мы можем найти его в электронной библиотеке и прочитать на экране компьютера. А также можем скачать и послушать аудиозапись чтения романа выдающимися актерами. По математике мы можем пройти тест в Google-форме, по дисциплине «Финансовая грамотность» — просмотреть видео для закрепления изученной на уроке информации. И, конечно, текстовый редактор позволяет нам красиво и грамотно оформить домашнюю работу, доклады и рефераты, а приложение для создания презентаций — подготовить слайд-шоу для выступления на защите проекта.

Из всего этого следует вопрос:

Если уже сейчас люди придумали такие технологии, которые значительно упрощают нашу жизнь, то что будет в будущем? Что еще можно придумать и как это внедрить в школьную программу?

На мой взгляд, в будущем в школах будет очень широко использоваться виртуальная реальность. Только представьте: на уроке географии, изучая какую-нибудь страну, мы сможем оказаться в ней, подробно рассмотреть все достопримечательности, очутиться в ее географической экзотичности, услышать шум водопадов и пение птиц. Можно путешествовать где угодно. Как здорово будет изучать астрономию: захотел — увидел вблизи планету или звезду. На уроках биологии виртуальная реальность даст возможность буквально «изнутри» изучить человеческий организм.

Доступные 3D-принтеры в школе будущего позволят каждому ученику создать то, что ему необходимо для решения своих задач. Например, модели каких-то объектов, распечатанные на 3D-принтере, помогут школьникам понять принцип работы с данными объектами, отточить навыки работы с ними.

Использование вместо обычных тетрадей электронных, хранящихся в облаке, снимет вопрос с привычным «а я забыл тетрадь дома». В электронной тетради можно будет не только выполнять задания, но и сразу отправлять ответы учителю для проверки.

Даже, вполне возможно, появится мультитач-дисплей. Это облегчит работу как учителю, так и ученикам. Данное устройство будет выглядеть как большой стол, обеспечивающий ввод данных сразу несколькими учениками.

И последнее (и, наверно, самое лучшее, что может случиться) — это образование за пределами школы. Благодаря всем информационным технологиям мы сможем обучаться не только в самой школе, но и из дома, из другого города, из другой страны.

И опыт последних лет показал, что все это не такое уж далекое будущее, что информационные технологии уже сейчас способны менять нашу жизнь, в том числе существенно помогать школьникам как в обучении, так и в творчестве, в личностной реализации. Хочется верить, что будущее развитие информационных технологий, а вместе с ним и совершенствование обучения в школе приведут к тому, что для школьников учеба станет самой интересной частью их жизни.

DOI: 10.32517/2221-1993-2022-21-3-51-51

Николай Перминов

дипломант Всероссийского конкурса «Школьная информатика: вчера, сегодня, завтра»

в номинации «Школьная информатика: заглянем в будущее»,

*учебно-консультационный пункт «Верхний Чов» Республиканского центра образования,
г. Сыктывкар, Республика Коми, Россия*

ИНФОРМАТИКЕ БЫТЬ!

В школьной программе информатика появилась относительно недавно (по сравнению, например, с математикой или историей). И с момента ее появления в школе и до сегодняшнего дня очень многие ученики любят этот предмет больше всех остальных. Любят в основном за то, что на уроках информатики можно «сидеть в компьютере» и никто за это не будет ругать, а наоборот, даже поставят оценку (обычно хорошую).

Когда информатика только появилась как школьный предмет, многие дети впервые увидели компьютер именно в кабинете информатики. Сейчас же компьютером не удивишь никого: мы учимся в дистанционном формате, наши родители работают «на удаленке», мы покупаем товары через интернет, смотрим фильмы, общаемся с друзьями, проводим все больше времени в виртуальном пространстве.

И я задумался: а что с информатикой будет дальше? Останется ли она в школе? Или цифровизация нашего общества сместит знакомство с информационными технологиями на первый уровень образования — в детский сад? Нужна ли будет информатика в школе?

Я посоветовался с родителями, с друзьями, со своим учителем информатики и все-таки решил: информатика в школе обязательно будет!

Информатика — это же не только и не столько программирование, сколько наука о способах получения, хранения, обработки и применения информации. Инфор-

матика учит нас логике и алгоритмизации, системности и однозначности. Информатика — это наука, объединяющая лучшие достижения других наук — математики, физики, языков, изобразительного искусства, музыки и многих других. Информатика учит нас интеграции, партнерству и даже, в некотором смысле, лояльности и толерантности.

И я думаю, что информатика будущего — это основной школьный предмет, в рамках которого детей будут учить, как использовать имеющиеся информационные и культурные ресурсы общества, как жить и общаться в быстро меняющемся информационном пространстве. На профильном уровне на уроках информатики будут учить программированию на языках высокого уровня и робототехнике.

Не нужно будет таскать с собой десятикилограммовые сумки с бумажными учебниками, пыльными книгами из библиотек и тетрадями, которые учителя забывают проверять, — все будет в электронном виде. Все ученики будут иметь возможность использовать интерактивные электронные учебники и рабочие тетради, изучать оцифрованные художественные произведения, посещать виртуальные музеи и экскурсии. Никто не будет требовать сдавать гаджеты при входе в класс — ноутбук, нетбук, графический планшет станут обыденными инструментами на уроке.

Такой мне видится информатика будущего, и хочется верить — самого ближайшего будущего!

DOI: 10.32517/2221-1993-2022-21-3-52-52

Евгения Шевчук

дипломант Всероссийского конкурса «Школьная информатика: вчера, сегодня, завтра»
в номинации «Школьная информатика: заглянем в будущее»,
средняя школа № 206, г. Минск, Беларусь

ПУСТЬ МЕНЯ НАУЧАТ!

Я с компьютером дружу,
Это ясно и ежу.
Я с закрытыми глазами
Покажу программы маме —
Word, Excel и школьный чат.
Дома на меня ворчат:
«Папка, почта, файл... Delete!
Целый день в компе сидит.
Может, надо ей к врачу?»
Я тихонько бормочу:
«Я в дизайнеры хочу!
Есть ли что-то круче?
Пусть меня научат!»

Думая о том, какой будет школа в будущем, я верю, что через несколько лет предмет «Информатика» станет самым увлекательным в школе. И каждый ученик на уроках информатики будет изучать то, что интересно *именно ему* и что пригодится *именно в его жизни*, а не в жизни абстрактного школьника или абстрактного взрослого.

Например, мне нравится рисовать. Я почти каждую свободную минуту что-то рисую. И на уроках информатики мне интересно было бы выполнять такие задания, где нужно что-то изображать на экране монитора или придумывать что-то, связанное с рисунками, анимацией, видео. В этом я вижу пользу для себя. Я не планирую стать программистом, поэтому те уроки, где нас учат программировать, для меня трудные. И даже скучные.

Я бы за это время создала увлекательные мультфильмы, которые радовали бы не только меня, но и других. Но если задания на программирование, на составление алгоритмов или на какие-то другие темы по информатике будут связаны с рисованием, анимацией, т. е. *с моими интересами*, возможно, это могло бы увлечь меня и помочь освоить данные темы.

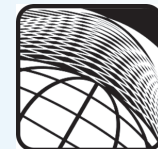
Я очень надеюсь, что в будущем информатика будет учитывать увлечения и наклонности *каждого* ученика.

Сначала школьники будут изучать те программы, которые потребуются в жизни абсолютно всем. Конечно, это, прежде всего, работа с текстом.

А затем изучение информатики будет разделяться по направлениям. Это значит, что будущие художники и музыканты не будут мучиться над непонятным для них программированием, а будут углубленно практиковаться в графических и музыкальных редакторах, воплощать свои художественные задумки на экранах мониторов и, соответственно, будут творчески относиться к урокам информатики.

А будущие программисты не будут рисовать на компьютере «каляки-маляки», а направят свои умственные способности именно на программирование, на участие в олимпиадах и конференциях по информатике.

И тогда наверняка
Станет для ученика
Информатика наукою чудесной!



DOI: 10.32517/2221-1993-2022-21-3-53-66

М. В. Маркушевич
Школа № 1352, г. Москва, Россия

ЭЛЕМЕНТЫ МЕТОДИКИ ПРЕПОДАВАНИЯ КОМПЬЮТЕРНОЙ ТРЕХМЕРНОЙ ГРАФИКИ НА УРОВНЕ ОСНОВНОГО ОБЩЕГО ОБРАЗОВАНИЯ С ИСПОЛЬЗОВАНИЕМ КОМПАС-3D LT V12

Аннотация

Трёхмерная компьютерная графика не является обязательной к изучению на уровне основного общего образования информационной технологией. Её изучение имеет отношение скорее к предпрофильному обучению по углубленным программам, используемым в таких образовательных проектах, как «Математическая вертикаль». В связи с этим наблюдается некоторый дефицит методик обучения данной теме, использующих в качестве программных инструментов российское бесплатное или свободное программное обеспечение.

В качестве наилучших вариантов программного обеспечения для поддержки преподавания темы «Трёхмерная компьютерная графика» предлагается рассматривать как отечественную бесплатную для некоммерческих и образовательных целей систему трёхмерного моделирования КОМПАС-3D LT V12, так и свободную систему трёхмерного параметрического моделирования FreeCAD. Выбор именно КОМПАС-3D LT V12 обусловлен прежде всего необходимостью использовать отечественное программное обеспечение в условиях постоянного ужесточения экономических санкций в отношении Российской Федерации со стороны ряда недружественных западных государств. Эти санкции оказывают негативное влияние в том числе и на информационную инфраструктуру российских образовательных организаций.

Авторская методика преподавания темы «Трёхмерная компьютерная графика» для основной школы обладает следующими характерными особенностями: может использоваться как для дистанционной, так и для очной формы обучения; базируется на отечественном программном обеспечении, в связи с чем инфраструктура российских образовательных организаций не попадает под угрозу немотивированного отказа в лицензиях на использование иностранного проприетарного ПО. Кроме того, применение в учебном процессе бесплатных версий программного обеспечения оптимизирует государственные расходы на образование.

Ключевые слова: КОМПАС-3D LT V12, FreeCAD, трёхмерная компьютерная графика, отечественное программное обеспечение, методика обучения.

1. Введение

Одной из разновидностей компьютерной графики, которая может изучаться в основной школе наряду с век-

торной и растровой графикой, является 3D-графика. Отметим, что в ФГОС основного общего образования, утвержденном Приказом Министерства просвещения РФ от 31 мая 2021 года № 287 [20], нет явных указаний

Контактная информация

Маркушевич Михаил Владимирович, учитель информатики, школа № 1352, г. Москва, Россия; адрес: 107497, Россия, г. Москва, ул. Байкальская, д. 44; e-mail: mihael11@yandex.ru

M. V. Markushevich
School 1352, Moscow, Russia

ELEMENTS OF THE METHODOLOGY OF TEACHING COMPUTER 3D GRAPHICS AT THE LEVEL OF BASIC GENERAL EDUCATION USING KOMPAS-3D LT V12

Abstract

3D computer graphics is information technology that is not mandatory for studying at the level of basic general education. Its study is more related to pre-profile training in advanced programs used in such educational projects as the Mathematical Vertical. In this regard, there is a certain shortage of teaching methods for this topic, using Russian free or open source software as software tools.

As the best software to support the teaching of the topic "3D computer graphics", it is proposed to consider both the domestic free 3D modeling system KOMPAS-3D LT V12 for non-commercial and educational purposes, and the free 3D parametric modeling system FreeCAD. The choice of KOMPAS-3D LT V12 is primarily due to the need to use domestic software in the face of constant tightening of economic sanctions against the Russian Federation by a number of unfriendly Western states. These sanctions also have a negative impact on the information infrastructure of Russian educational institutions.

The author's method of teaching the topic "3B computer graphics" in basic school has the following characteristic features: it can be used both for distance learning and for full-time education; is based on domestic software, in connection with which the infrastructure of Russian educational organizations does not fall under the threat of an unmotivated refusal of licenses to use foreign proprietary software. In addition, the use of free versions of software in the educational process optimizes government spending on education.

Keywords: KOMPAS-3D LT V12, FreeCAD, 3D computer graphics, domestic software, teaching methods.

на обязательность изучения 3D-графики на уровне основного общего образования. Однако в данном документе можно обнаружить формулировки, указывающие в том числе на наличие потенциальной возможности организации изучения трехмерной графики. Так, например, в числе универсальных учебных познавательных действий в разделе «Работа с информацией» говорится о необходимости учащимся овладеть навыками самостоятельно выбирать оптимальную форму представления информации и иллюстрировать решаемые задачи несложными схемами, диаграммами, иной графикой и их комбинациями. Под иной графикой вполне можно понимать несложную трехмерную графику, так как в различных учебных предметах, таких как физика, геометрия, биология и т. д., изучаемых в основной школе, школьники имеют дело с объемными фигурами (прямоугольный параллелепипед, куб, шар, цилиндр, конус и т. д.).

Кроме того, в числе предметных результатов по учебному предмету «Информатика» как на базовом, так и на углубленном уровне присутствует владение умениями и навыками использования информационных и коммуникационных технологий для обработки и анализа различных видов информации, среди которых вполне разумно было рассматривать также и трехмерную графическую информацию.

Максимальную актуальность преподавание трехмерной компьютерной графики приобретает в рамках предпрофильного обучения в основной школе по углубленным программам, используемым в таком проекте российского образования, как «Математическая вертикаль» [16]. В данном случае можно говорить о подготовке школьников к продолжению образования в профильных инженерных и ИТ-классах в старшей школе. Для обучения в этих классах пространственное мышление является необходимой метапредметной компетенцией, а освоение трехмерной компьютерной графики прежде всего развивает именно пространственное мышление и воображение. Важность развития пространственного воображения и мышления у учащихся, в том числе с использованием системы трехмерного проектирования КОМПАС-3D, особенно в ситуации, когда черчение не является обязательным предметом к изучению в основной школе, отмечается также в статье М. С. Александрова [2].

При неопределенном положении 3D-графики в содержании основного общего образования большинство ведущих авторских коллективов, разрабатывающих учебные пособия для основной школы, не видят необходимости в написании соответствующих разделов учебников, посвященных рассматриваемой нами теме курса информатики.

Таким образом, с точки зрения автора, компьютерная трехмерная графика на уровне основного общего образования из курса информатики смещается в область внеурочной деятельности — по тематике «Компьютерное черчение», «Компьютерная графика» или «3D-графика», а также дополнительного образования — в основном для учащихся, обучающихся в предпрофильных классах по программам «Математическая вертикаль». Еще один возможный путь внедрения элементов трехмерной компьютерной графики в содержание обучения по

информатике в основной школе — это проведение полипрограммных интегрированных уроков, технология проектирования которых для старшей школы подробно рассмотрена автором в работе [15].

2. Выбор оптимального типа программного обеспечения для поддержки преподавания темы «Компьютерная трехмерная графика» в основной школе

Выбор конкретного типа программного обеспечения для поддержки преподавания той или иной темы курса информатики в основной школе является непростым вопросом, особенно в современных условиях, когда крупные иностранные вендоры ИТ-индустрии, такие как Microsoft Corporation, Adobe Inc., Autodesk, уходят с российского рынка программного обеспечения в результате введения и последовательного ужесточения экономических санкций в отношении России со стороны ряда недружественных западных государств [24]. В данной ситуации рационально обратить внимание, прежде всего, на свободное программное обеспечение или на бесплатное программное обеспечение российской разработки.

Здесь важно понимать, что *свободное ПО не эквивалентно бесплатному проприетарному ПО, но с точки зрения использования в образовательных организациях их объединяет общее свойство, а именно возможность для пользователей легально и бесплатно применять данное программное обеспечение в учебном процессе.*

Бесплатность не единственное свойство программного обеспечения, определяющее его привлекательность для применения в сфере образования. **Необходимо учитывать следующие характеристики:**

- *страна происхождения фирмы — разработчика ПО:* определяет уровень риска использования данного ПО в учебном процессе основной школы ввиду возможности экономических санкций и ограничений со стороны иностранной фирмы — разработчика ПО;
- *является ПО офлайновым программным обеспечением или онлайн-сервисом, требующим постоянного высокоскоростного подключения к интернету:* определяет возможность или невозможность использования в образовательных организациях, не имеющих высокоскоростного подключения;
- *открытость программного кода:* определяет возможность скомпилировать программу самостоятельно, независимо от разработчика ПО;
- *кроссплатформенность:* определяет список операционных систем, под которыми возможно установить и применять данное ПО;
- *минимальные системные требования:* определяют парк компьютеров, на которых возможно установить и применять данное ПО;
- *доступный функционал пользователя:* определяет список возможных действий над информацией, обрабатываемой данным ПО;

- *уровень сложности интерфейса пользователя и алгоритмов создания обрабатываемого в данном ПО вида информации*: определяет минимальный возрастной порог учащихся, с которого можно начинать знакомство с данным ПО.

Переходя непосредственно к нашей ситуации, а именно к выбору программного обеспечения для поддержки темы «Компьютерная трехмерная графика» на уровне основного общего образования, важно сразу отказаться от использования в учебном процессе импортных проприетарных систем автоматизированного проектирования американской компании Autodesk Inc., таких как AutoCAD, Fusion 360, а также САПР SolidWorks разработки SolidWorks Corporation (Франция). Несмотря на то что для данного программного обеспечения предусмотрен вариант бесплатной образовательной лицензии, что очень удобно для общеобразовательных российских школ, *существует высокий риск прекращения оказания услуг компанией Autodesk Inc. для клиентов на российском рынке ввиду ужесточения экономических санкций в отношении России рядом западных государств* [24].

Также, по мнению автора, было бы разумно отказаться от использования в учебном процессе основной школы, осуществляемом в очной форме, онлайн-овых 3D графических редакторов, таких как 3D-редактор Tinkercad, расположенный на портале <https://www.tinkercad.com>, а также бесплатной версии редактора SketchUp — SketchUp Free, доступной пользователю в сети по адресу: <https://www.sketchup.com/ru/plans-and-pricing/sketchup-free>. Действительно, как ранее отмечалось автором, онлайн-овые сервисы хороши для реализации учебного процесса в дистанционной форме в крупных городах, обеспеченных высокоскорост-

ным подключением к сети Интернет [14], но вряд ли являются оптимальным вариантом для поддержки преподавания информационных технологий в очной форме обучения для общеобразовательных российских школ, некоторые из которых не имеют стабильного подключения к интернету. Здесь важно понимать, что онлайн-овый 3D-редактор Tinkercad является проектом компании Autodesk Inc., о действиях которой на российском рынке программного обеспечения в условиях постоянно расширяющихся экономических санкций говорилось выше в этой статье.

Таким образом, после отказа от импортных, как онлайн-овых, так и офлайн-овых, проприетарных систем автоматизированного проектирования **мы можем остановиться на следующих двух возможных вариантах программного обеспечения**:

- российская бесплатная для некоммерческих и образовательных целей система трехмерного моделирования КОМПАС-3D LT V12 разработки ООО «АСКОН — Системы проектирования» [23];
- свободные системы автоматизированного проектирования, среди которых наиболее популярной является кроссплатформенная система трехмерного параметрического моделирования FreeCAD [22].

Некоторые аспекты применения FreeCAD в учебном процессе при преподавании геометрии были рассмотрены автором (в соавторстве с А. Ю. Федосовым) в работе [25]. В настоящей статье мы попытаемся сравнить два рассматриваемых варианта программного обеспечения по приведенным выше характеристикам (табл. 1).

Другие свободные системы автоматизированного проектирования, такие как LibreCAD, OpenSCAD, SolveSpace, BricsCAD, QCAD [5], не будут рассматри-

Таблица 1

Сравнение КОМПАС-3D LT V12 и FreeCAD

№ п/п	Характеристика ПО	КОМПАС-3D LT V12	FreeCAD
1	Страна происхождения фирмы — разработчика ПО	Россия	Интернациональное сообщество разработчиков
2	Открытость программного кода	Закрытый код	Открытый код
3	Кроссплатформенность	Windows, Linux (к 2023 году) [6]	Windows, Linux, Mac OS
4	Минимальные системные требования	Минимально возможная конфигурация компьютера для установки и запуска системы соответствует минимальным системным требованиям для соответствующих операционных систем Windows [11], а именно: • процессор: 1 ГГц и старше или SoC (System-on-a-Chip); • ОЗУ: 1 ГБ (для 32-разрядных систем) или 2 ГБ (для 64-разрядных систем); • место на диске: 16 ГБ (для 32-разрядных систем) или 20 ГБ (для 64-разрядных систем); • видеокарта: DirectX 9 или выше с драйвером WDDM 1.0; • разрешение экрана: 800×600 [18]	В отличие от большинства программ 3D CAD, FreeCAD может без проблем работать на самых скромных компьютерах — известно, что он работает на процессорах Pentium IV и Intel Core2 Solo. Если на компьютере установлена текущая операционная система (Windows, Linux, Mac OS), то FreeCAD будет работать [10]. Иными словами, системные требования для FreeCAD под Windows совпадают с системными требованиями Windows 10 [18] и, следовательно, КОМПАС-3D LT V12

Окончание табл. 1

№ п/п	Характеристика ПО	КОМПАС-3D LT V12	FreeCAD
5	Доступный функционал пользователя	• Создание двумерных чертежей; • автоматическое создание чертежа из трехмерной модели; • трехмерное моделирование деталей с использованием операций выдавливания, вращения, кинематической операции; • поверхностное моделирование	• Создание двумерных чертежей; • автоматическое создание чертежа из трехмерной модели; • трехмерное моделирование деталей; • изучение движения роботов в графическом интерфейсе; • широкие возможности импорта/экспорта в стандартные форматы; • экспорт с помощью модуля рендеринга 3D-объектов для рендеринга во внешние программы; • наличие архитектурного модуля
6	Уровень сложности интерфейса пользователя	Средний	Выше среднего

ваться в данной статье в связи с тем, что они являются менее известными, чем FreeCAD, что подтверждается малым числом их упоминания в отечественной научной и научно-методической литературе, посвященной вопросам выбора программного обеспечения для поддержки преподавания тем «Трехмерная компьютерная графика» или «Системы автоматизированного проектирования» в общеобразовательной школе.

Отметим, что ООО «АСКОН — Системы проектирования» не единственный отечественный разработчик систем трехмерного проектирования, в данном направлении работает также ООО «Нанософт разработка» [17], в ее линейке программных продуктов nanoCAD присутствует бесплатный дистрибутив для использования в образовательных целях, но, к сожалению, его минимальные системные требования к оперативной памяти компьютера в 4 гигабайта и рекомендуемое значение оперативной памяти в 16 гигабайт заставляют сомневаться в рациональности использования данного программного обеспечения в учебном процессе основной школы.

Сравнительный анализ двух альтернативных САПР, приведенных в таблице 1, не дает возможности определить очевидного лидера. Оба рассматриваемых варианта имеют достаточное количество достоинств, в связи с чем окончательное решение в пользу использования именно КОМПАС-3D LT V12 определяется, прежде всего, отечественным происхождением данной системы трехмерного моделирования, что достаточно важно как в плане обеспечения устойчивости информационно-инфраструктуры российских образовательных организаций к рискам, заключающимся в отказе в лицензиях на использование иностранного проприетарного программного обеспечения в условиях постоянного ужесточения экономических санкций со стороны ряда недружественных западных государств, так и в плане гражданско-патриотического воспитания учащихся [13].

Кроме указанных выше факторов выбор отечественной системы трехмерного моделирования КОМПАС-3D LT V12 для преподавания трехмерной графики в основной школе обусловлен следующими ее положительными

характеристиками в плане применения для образовательных целей:

- возможность бесплатного использования для образовательных и личных некоммерческих целей;
- достаточно скромные минимальные системные требования (2 гигабайта оперативной памяти для 64-разрядных систем);
- средний уровень сложности пользовательского интерфейса, позволяющий использовать данное программное обеспечение на уровне основного общего образования;
- насыщенный функционал программного обеспечения, позволяющий создавать достаточно сложные, разнообразные по формам трехмерные модели;
- наличие существенной методической базы в виде учебной и учебно-методической литературы, среди которой можно выделить учебные пособия Н. Б. Ганина [6, 7] и А. А. Герасимова [8], а также интернет-ресурсов, посвященных методике применения системы трехмерного проектирования КОМПАС-3D в учебном процессе, например, обучающие скринкасты на официальном сайте проекта, расположенные в сети по адресу: <https://kompas.ru/publications/video/>, или бесплатный учебник «Азбука КОМПАС-3D» [1].

Отметим также положительный опыт применения КОМПАС-3D LT V12 в учебном процессе общеобразовательной школы, например, описанный В. Н. Барановым в статье [3].

3. Поурочное планирование по теме «Компьютерная трехмерная графика» с использованием системы трехмерного моделирования КОМПАС-3D LT V12

В таблице 2 приведены темы уроков по курсу «Компьютерная трехмерная графика» с использованием системы трехмерного моделирования КОМПАС-3D LT V12 и их основное содержание, а также указано, в какой форме проводится урок.

Таблица 2

Поурочное планирование по теме «Компьютерная трехмерная графика»

№ урока	Тема и основное содержание урока	Тип урока
1	Официальный сайт системы трехмерного моделирования КОМПАС-3D. Получение дистрибутива КОМПАС-3D LT V12. Лицензия на использование. Установка и запуск КОМПАС-3D LT V12 под операционной системой Windows	Семинар
2	Основные элементы интерфейса пользователя (дерево модели, основные панели инструментов, панель свойств, рабочая область трехмерного моделирования, трехмерная система координат — оси координат, плоскости)	Семинар
3	Трехмерная система координат, оси координат, координатные плоскости. Изменение ориентации системы координат в пространстве (кнопка <i>Ориентация</i> панели инструментов <i>Вид</i>). Активация режима <i>Эскиз</i> в одной из выбранных координатных плоскостей. Создание фигуры (контура) в режиме <i>Эскиз</i> с помощью инструментов панели <i>Геометрия</i>	Семинар, практическая работа
4	Активация и параметры операции <i>Выдавливание</i> (выдавливаемый контур, направления выдавливания, расстояние выдавливания, активация уклона наружу или внутрь). Инструменты поворота и перемещения трехмерной модели в пространстве, масштабирование модели	Семинар, практическая работа
5	<i>Самостоятельная практическая работа по теме «Создание трехмерных моделей основных геометрических фигур (цилиндр, конус, призма и т. п.)»</i>	Практическая работа
6	Создание эскизов в различных плоскостях, в том числе в плоскостях граней какой-либо трехмерной фигуры. Использование данных эскизов для последующего применения операций <i>Выдавливание</i> или <i>Вырезать выдавливанием</i>	Семинар, практическая работа
7	<i>Самостоятельная практическая работа по теме «Создание трехмерной модели грузового автомобиля»</i>	Практическая работа
8	Создание трехмерных фигур методом вращения контура вокруг оси	Семинар, практическая работа
9	<i>Самостоятельная практическая работа по теме «Создание трехмерных моделей молекул»</i>	Практическая работа
10	Создание трехмерных фигур методом движения сечения вдоль траектории (кинематическая операция). Черчение сечения и траектории в эскизах, расположенных в различных плоскостях	Семинар, практическая работа
11	<i>Самостоятельная практическая работа по теме «Создание трехмерной модели цепи»</i>	Практическая работа
12	Создание сложных трехмерных фигур с использованием различных методов формирования 3D-моделей. Создание вспомогательных плоскостей различными способами (параллельно заданной плоскости, перпендикулярно заданной плоскости, проходящей через определенную точку и т. п.)	Семинар, практическая работа
13	<i>Самостоятельная практическая работа по теме «Создание трехмерной модели чашки»</i>	Практическая работа
14	<i>Контрольная практическая работа по теме «Создание трехмерной модели космического корабля»</i>	Практическая работа
15	Анализ контрольной практической работы по теме «Создание трехмерной модели космического корабля»	Конференция

4. Основное содержание обучения по теме «Компьютерная трехмерная графика» с использованием системы трехмерного моделирования КОМПАС-3D LT V12

4.1. Этап 1

Урок 1. Официальный сайт системы трехмерного моделирования КОМПАС-3D. Получение дистрибутива КОМПАС-3D LT V12. Лицензия на использо-

вание. Установка и запуск КОМПАС-3D LT V12 под операционной системой Windows.

Урок 2. Основные элементы интерфейса пользователя (дерево модели, основные панели инструментов, панель свойств, рабочая область трехмерного моделирования, трехмерная система координат — оси координат, плоскости).

Получение дистрибутива системы трехмерного моделирования КОМПАС-3D LT V12 осуществляется на официальном сайте проекта, расположенном в сети Интернет по адресу: <https://kompas.ru/kompas-3d-lt/>

download/, после заполнения формы, содержащей фамилию, имя, страну, город и адрес электронной почты лица, заинтересованного в использовании данного программного обеспечения.

Перед установкой на персональный компьютер необходимо ознакомиться на официальном сайте проекта с лицензионным соглашением на использование программного продукта КОМПАС-3D LT V12, в котором указано, что данное программное обеспечение может быть *использовано исключительно для личного некоммерческого использования (в ознакомительных и учебных целях (выполнять собственные домашние задания, курсовые и дипломные проекты, другие учебные работы); или в образовательных организациях исключительно в соответствии с уставными целями для осуществления образовательного процесса и только в образовательных целях* [12].

С подробным описанием интерфейса пользователя КОМПАС-3D LT V12 можно ознакомиться в учебных пособиях А. А. Герасимова [8], Н. Б. Ганина [6, 7], а также в статье И. В. Прояевой, К. Е. Цветковой [21].

4.2. Этап 2

Урок 3. *Трехмерная система координат, оси координат, координатные плоскости. Изменение ориентации системы координат в пространстве (кнопка **Ориентация** панели инструментов **Вид**). Активация режима **Эскиз** в одной из выбранных координатных плоскостей, создание фигуры (контура) в режиме **Эскиз** с помощью инструментов панели **Геометрия**.*

Урок 4. *Активация и параметры операции **Выдавливание** (выдавливаемый контур, направления*

выдавливания, расстояние выдавливания, активация уклона наружу или внутрь). Инструменты поворота и перемещения трехмерной модели в пространстве, масштабирование модели.

Урок 5. *Самостоятельная практическая работа по теме «Создание трехмерных моделей основных геометрических фигур (цилиндр, конус, призма и т. п.)».*

На данном этапе учащиеся создают основные простые трехмерные геометрические фигуры, такие как прямоугольный параллелепипед (рис. 1), цилиндр (рис. 2), пятиугольная призма (рис. 3).

Затем задействуется функционал выдавливания с активацией угла внутрь, применение которого дает возможность создавать трехмерные модели таких геометрических фигур, как пирамида (рис. 4), усеченный конус (рис. 5), шестиугольная усеченная пирамида (рис. 6).

4.3. Этап 3

Урок 6. *Создание эскизов в различных плоскостях, в том числе в плоскостях граней какой-либо трехмерной фигуры. Использование данных эскизов для последующего применения операций **Выдавливание** или **Вырезать выдавливанием** с целью получения либо дополнительной трехмерной фигуры, либо отверстия в существующей фигуре.*

Урок 7. *Самостоятельная практическая работа по теме «Создание трехмерной модели грузового автомобиля».*

Для отработки навыков владения операциями **Выдавливание** и **Вырезать выдавливанием** в КОМПАС-3D LT V12 учащимся может быть предложена практическая

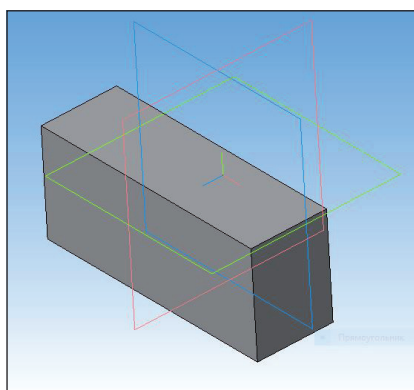


Рис. 1. Прямоугольный параллелепипед

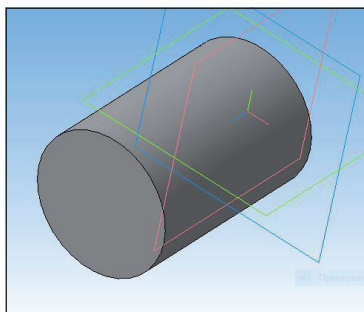


Рис. 2. Цилиндр

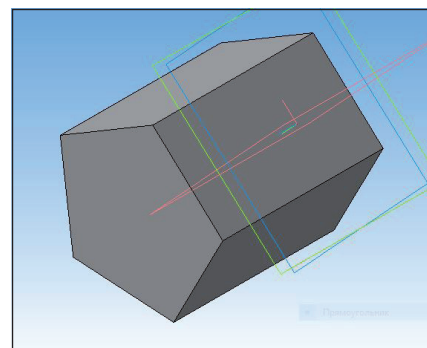


Рис. 3. Пятиугольная призма

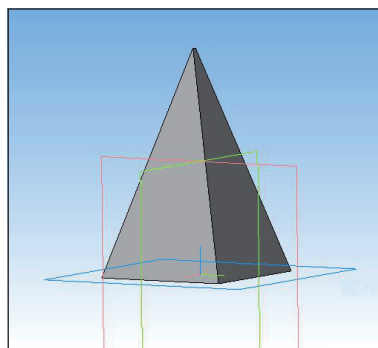


Рис. 4. Пирамида

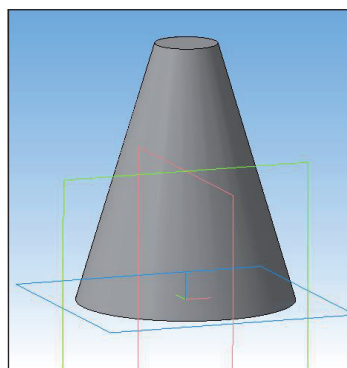


Рис. 5. Усеченный конус

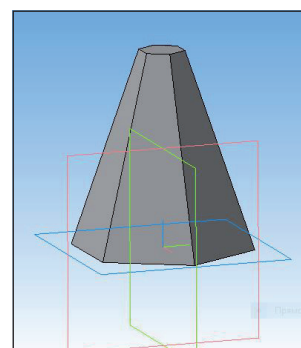


Рис. 6. Шестиугольная усеченная пирамида

работа по созданию трехмерной модели грузового автомобиля, которая выполняется по следующему алгоритму:

1. Создание кузова автомобиля (форма — прямоугольный параллелепипед, см. рис. 1).
2. Создание в торцевой грани кузова эскиза прямоугольника и последующее выдавливание второго прямоугольного параллелепипеда (кабина автомобиля) (рис. 7).
3. Создание в боковой грани кабины эскиза прямоугольника и вырезание им (операция *Вырезать выдавливанием*) ненужной части кабины (рис. 8).
4. Создание боковых окон в кабине автомобиля путем черчения и последующего вырезания выдавливанием прямоугольника в эскизе в боковой плоскости кабины (рис. 9).
5. Создание лобового окна кабины автомобиля путем черчения прямоугольника и последующего вырезания выдавливанием в эскизе в верхней передней плоскости кабины автомобиля (рис. 10).
6. Создание передней оси колес автомобиля путем черчения и последующего выдавливания в два направления окружности в эскизе в боковой плоскости кабины автомобиля (рис. 11).

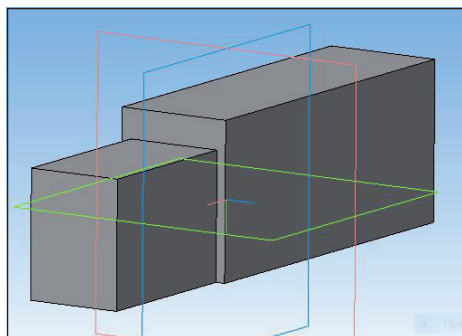


Рис. 7. Создание кабины автомобиля

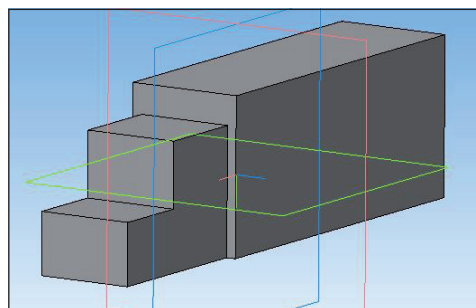


Рис. 8. Вырезание ненужной части кабины

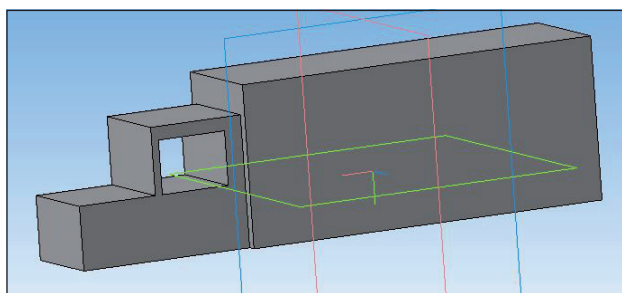


Рис. 9. Создание боковых окон кабины

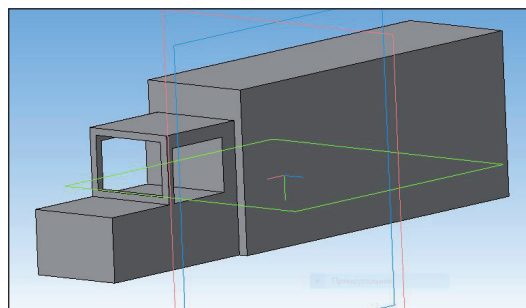


Рис. 10. Создание лобового окна автомобиля

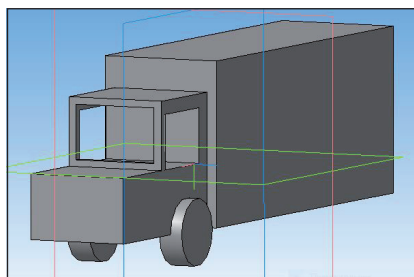


Рис. 11. Создание передней оси колес автомобиля

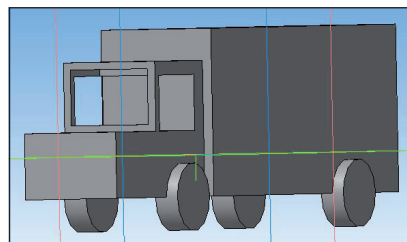


Рис. 12. Создание задней оси колес автомобиля

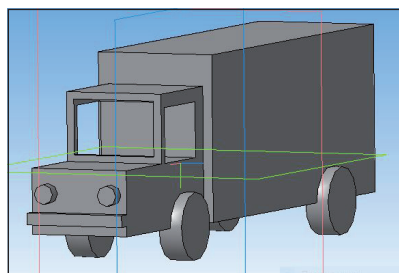


Рис. 13. Создание переднего бампера и фар

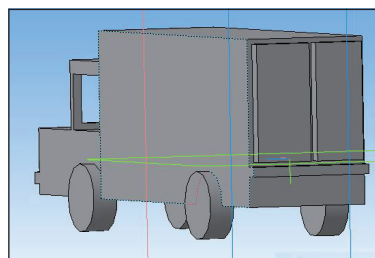


Рис. 14. Создание заднего бампера и дверей кузова

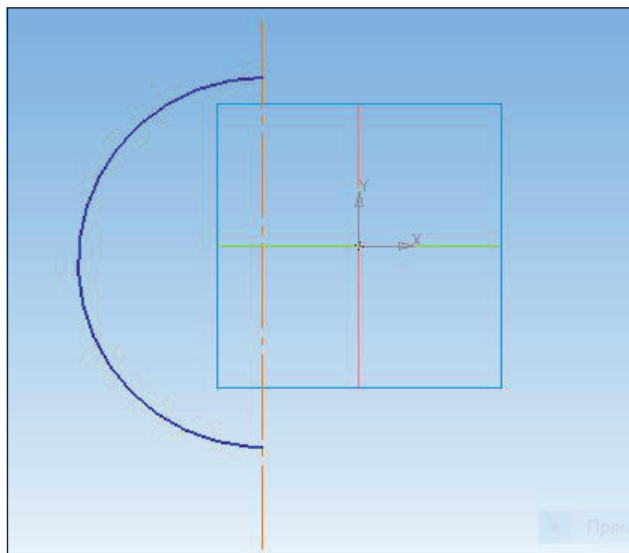


Рис. 15. Ось вращения и контур (дуга окружности)

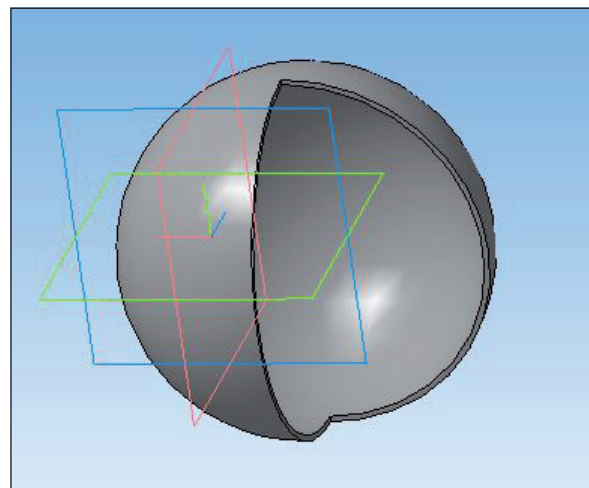


Рис. 16. Результат выполнения операции вращения с углом поворота контура 270°

7. Создание задней оси колес автомобиля путем черчения и последующего выдавливания в два направления окружности в эскизе в боковой плоскости кузова автомобиля (рис. 12).
8. Создание переднего бампера и фар путем черчения и последующего одновременного выдавливания прямоугольника и двух окружностей в эскизе в нижней передней плоскости кабины (рис. 13).
9. Создание заднего бампера и дверей путем черчения и последующего одновременного выдавливания трех прямоугольников в эскизе в задней плоскости кузова автомобиля (рис. 14).

4.4. Этап 4

Урок 8. Создание трехмерных фигур методом вращения контура вокруг оси.

Урок 9. Самостоятельная практическая работа по теме «Создание трехмерных моделей молекул»

Технология создания трехмерных фигур методом вращения в КОМПАС-3D LT V12 достаточно проста. Проиллюстрируем ее суть на примере создания шара, при этом будет реализован следующий алгоритм:

1. Аналогично процессу создания 3D-фигур методом выдавливания контура прежде всего выбирается плоскость, в которой будут лежать ось вращения и контур, потом в выбранной плоскости создается эскиз.
2. Далее в выбранной для создания эскиза плоскости чертится ось вращения, для чего используется инструмент *Отрезок* панели инструментов *Геометрия*, причем на панели свойств выбирается стиль линии *Осевая*.
3. Затем в плоскости эскиза чертится контур, который предполагается вращать вокруг оси. В нашем случае с помощью инструмента *Дуга* панели инструментов *Геометрия* была начерчена половина окружности таким образом, что начальная и конечная точки дуги лежат на оси вращения. Для дуги выбирается стиль линии *Основная* (рис. 15).

4. На *Компактной панели* активируется инструмент *Операция вращения*, для завершения которой необходимо указать угол, на который будет повернут контур (дуга окружности) вокруг оси вращения. В нашем случае был выбран угол 270° . Результат выполнения операции вращения показан на рисунке 16.

Для закрепления навыков учащихся в области создания трехмерных фигур методом вращения контура вокруг оси можно предложить следующие задания:

1. Создание трехмерной модели вазы. Для выполнения задания используется вращение кривой Безье вокруг оси вращения. Данная идея заимствована автором из статьи С. В. Зенкиной, М. В. Савченко-вой [9]. Стадии разработки модели вазы показаны на рисунках 17 и 18.
2. Создание трехмерной молекулы воды. Стадии разработки модели показаны на рисунках 19 и 20.

4.5. Этап 5

Урок 10. Создание трехмерных фигур методом движения сечения вдоль траектории (кинематическая операция). Черчение сечения и траектории в эскизах, расположенных в различных плоскостях.

Урок 11. Самостоятельная практическая работа по теме «Создание трехмерной модели цепи».

Метод создания трехмерных фигур движением сечения вдоль траектории является самым сложным из используемых в системе трехмерного проектирования КОМПАС-3D LT V12. Его сложность заключается в том, что направляющая, вдоль которой движется контур, и сам контур лежат в различных плоскостях, обычно перпендикулярных друг другу, что требует от учащихся наличия уже достаточно развитого пространственного мышления. В связи с этим автор предлагает изучение рассматриваемого метода оставить на финальную стадию обучения по теме «Компьютерная трехмерная графика».

В качестве примера задания, которое учащиеся выполняют для овладения навыками использования кине-

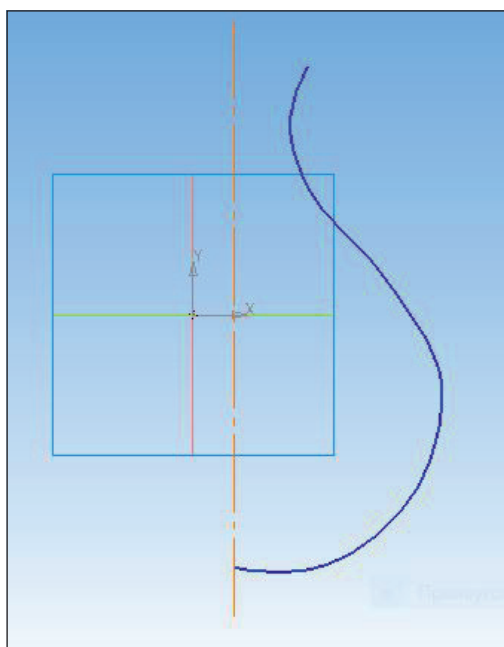


Рис. 17. Ось вращения и контур для создания модели вазы

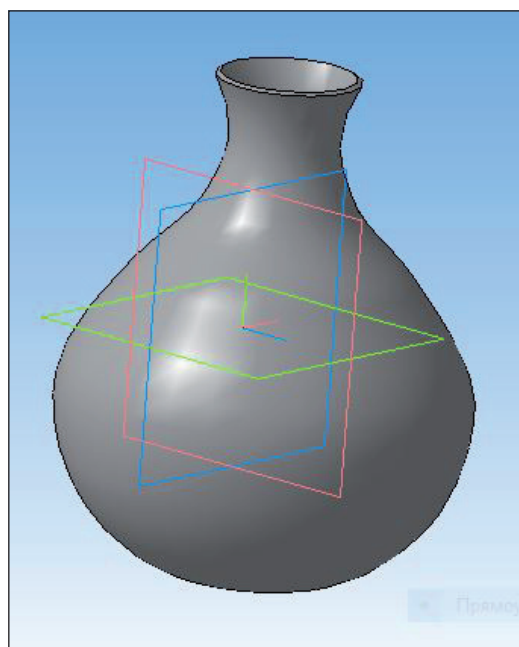


Рис. 18. Завершенная модель вазы

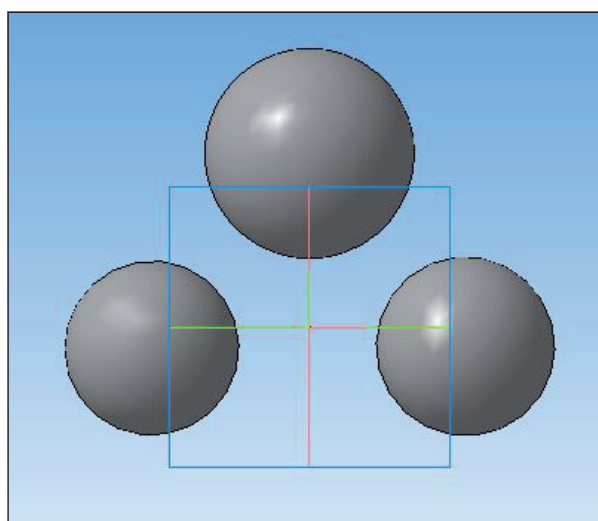


Рис. 19. Начальная стадия создания модели молекулы воды (три атома)

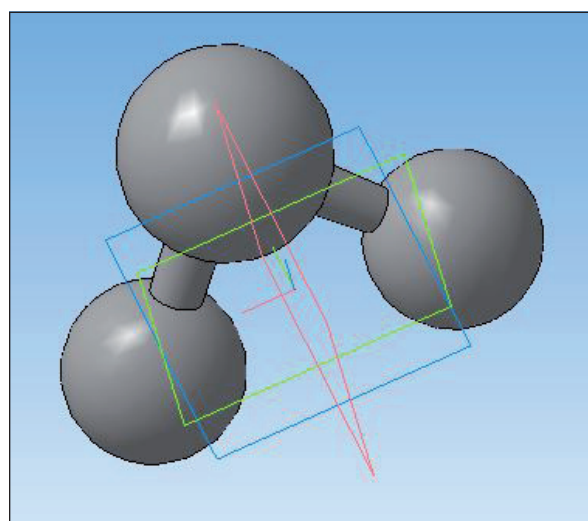


Рис. 20. Завершенная модель молекулы воды

матической операции, можно предложить создание трехмерной модели цепи, состоящей из нескольких звеньев.

Рассмотрим примерный алгоритм создания модели цепи:

1. Выберем одну из трех координатных плоскостей и создадим в ней эскиз. Далее в выбранной плоскости начертим эллипс, задав некое определенное значение большой и малой полуосей.
2. Перейдем в плоскость, перпендикулярную плоскости эллипса, создадим в ней эскиз и начертим окружность таким образом, чтобы центр окружности лежал на эллипсе, а ее радиус был значительно меньше малой полуоси эллипса (в несколько раз). На рисунке 21 траектория (эллипс) показана красным цветом, а сечение (окружность) — темно-синим.
3. С помощью инструмента *Редактирование детали* активируем *Кинематическую операцию*,

укажем, что сечением является окружность, а траекторией его движения — эллипс, и получим первое звено цепи (рис. 22).

4. Создадим эскиз в координатной плоскости, показанной на рисунке 22 зеленым цветом, и начертим в ней эллипс с размерами, равными размерам первого эллипса.
5. Выберем координатную плоскость, показанную на рисунке 22 светло-синим цветом и начертим в ней окружность того же радиуса, что и первая окружность (рис. 23).
6. Активируем *Кинематическую операцию*, укажем в качестве сечения окружность, а в качестве траектории — эллипс и получим второе звено цепи (рис. 24).
7. Дальнейшие действия по созданию третьего звена цепи полностью аналогичны ранее описанным.

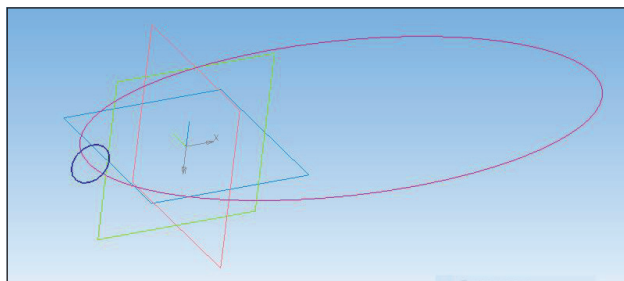


Рис. 21. Создание направляющей (эллипса) и контура (окружности) для первого звена цепи

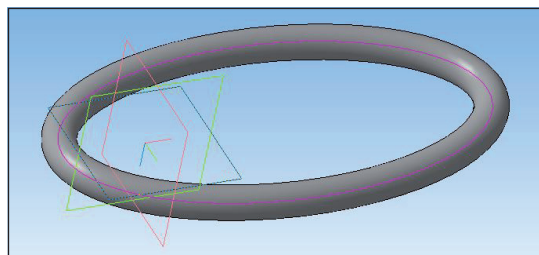


Рис. 22. Первое звено создаваемой цепи

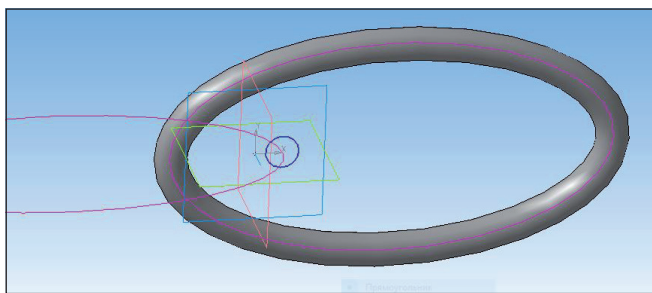


Рис. 23. Создание сечения и траектории для второго звена цепи

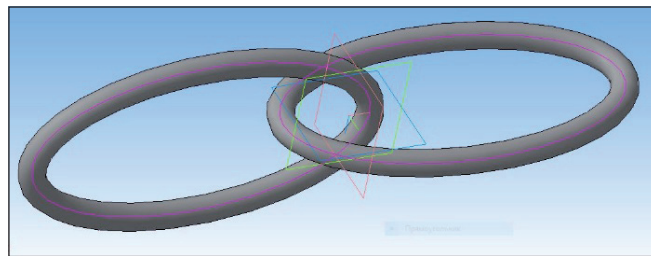


Рис. 24. Два звена создаваемой цепи

4.6. Этап 6

Урок 12. Создание сложных трехмерных фигур с использованием различных методов формирования 3D-моделей. Создание вспомогательных плоскостей различными способами (параллельно заданной плоскости, перпендикулярно заданной плоскости, проходящей через определенную точку и т. п.).

Урок 13. Самостоятельная практическая работа по теме «Создание трехмерной модели чашки».

Урок 14. Контрольная практическая работа по теме «Создание трехмерной модели космического корабля».

Урок 15. Анализ контрольной практической работы по теме «Создание трехмерной модели космического корабля».

В качестве примера сложной трехмерной фигуры, которая создается с помощью различных методов, можно рассмотреть модель кофейной чашки, которая может быть реализована в системе трехмерного проектирования КОМПАС-3D LT V12 с помощью следующего примерного алгоритма:

1. Прежде всего создадим ручку чашки. Для этого выберем одну из координатных плоскостей, создадим в ней эскиз и начертим траекторию будущей ручки с помощью инструмента *кривая Безье* панели инструментов *Геометрия*.
2. Перейдем в плоскость, перпендикулярную первой плоскости, создадим в ней эскиз и начертим окружность (показана темно-синим цветом на рисунке 25), центр которой лежит на траектории (показана фиолетовым цветом на рисунке 25).
3. Активируем *Кинематическую операцию*, укажем в качестве сечения окружность, а в качестве траектории — *кривую Безье* и создадим ручку чашки (рис. 26).

4. Далее перейдем к созданию тела чашки. Для этого выберем координатную плоскость, показанную на рисунке 26 голубым цветом, создадим в ней эскиз и начертим *кривую Безье*, которая будет формировать тело чашки с помощью операции *Вращение*, а также ось вращения (рис. 27).

5. Активируем *Операцию вращения*, в параметрах данной операции укажем, что мы создаем тонкую стенку у тела вращения, и подберем оптимальную толщину этой стенки. В результате получим чашку, показанную на рисунке 28. Полученная нами чашка не имеет дна, так как кривая Безье, создаваемая на 4-м шаге данного алгоритма, не доходит до оси вращения.

6. Для создания дна чашки нам придется воспользоваться вспомогательной плоскостью, лежащей в плоскости будущего дна. Используем инструмент *Смещенная плоскость* панели *Вспомогательная геометрия*. На рисунке 29 вспомогательная плоскость показана коричневым цветом и параллельна плоскости, показанной ярко-красным цветом.

7. Далее во вспомогательной плоскости создается эскиз, в котором чертится окружность диаметром, несколько большим диаметра отверстия в дне чашки. Затем данная окружность выдавливается, в результате чего формируется дно чашки. Законченная модель чашки со сформированным дном показана на рисунке 30.

Еще одним вариантом практической работы, направленной на закрепление навыков учащихся в области применения различных способов создания сложных объемных фигур, может быть создание трехмерной модели космического корабля. Технология выполнения предлагаемой работы показана на рисунках 31–44.

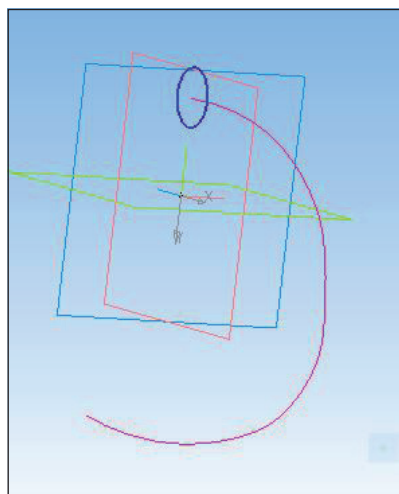


Рис. 25. Траектория и контур для создания ручки чашки с помощью **Кинематической операции**

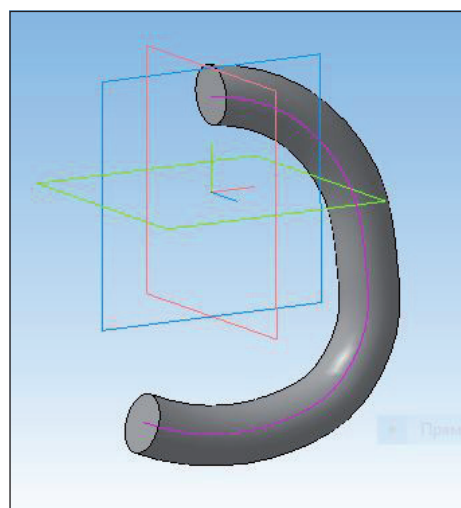


Рис. 26. Ручка чашки, полученная с помощью **Кинематической операции**

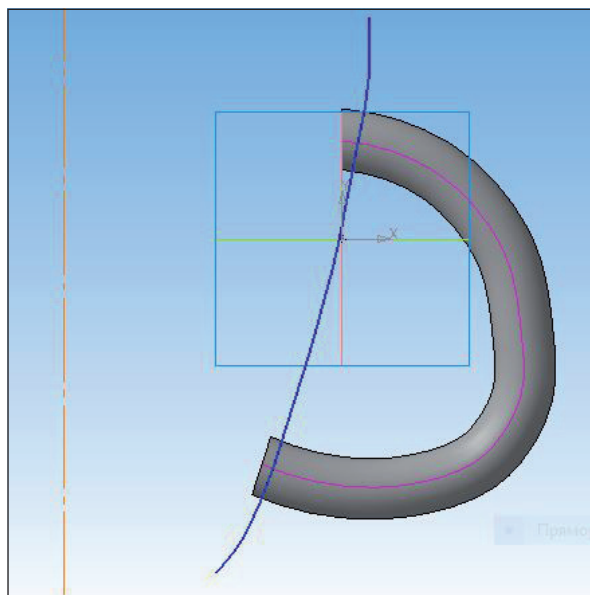


Рис. 27. **Кривая Безье** и ось вращения для создания будущего тела чашки

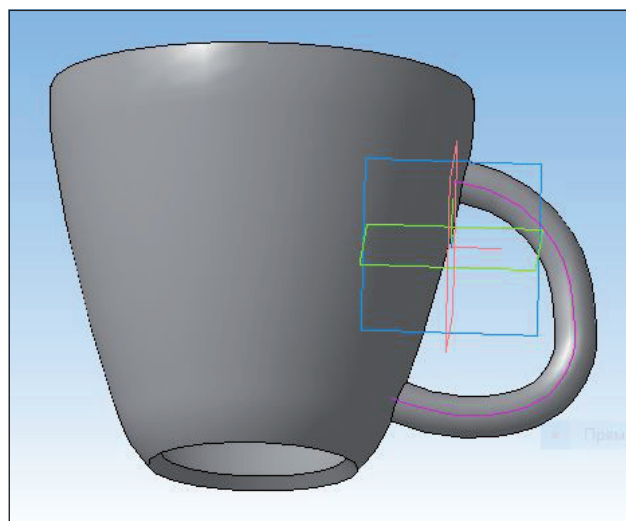


Рис. 28. Сформированная чашка без дна

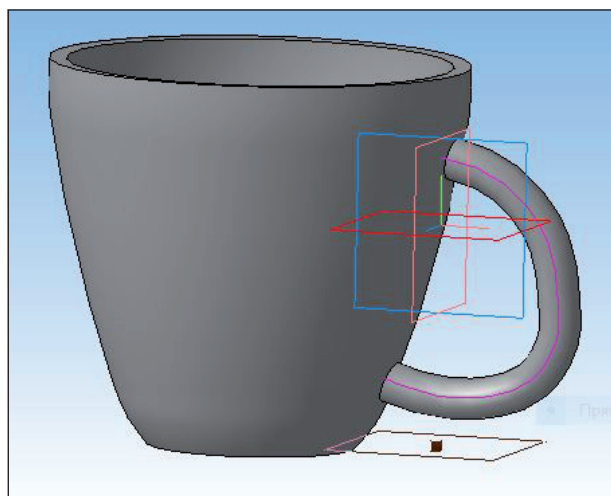


Рис. 29. Создание вспомогательной плоскости, параллельной заданной плоскости

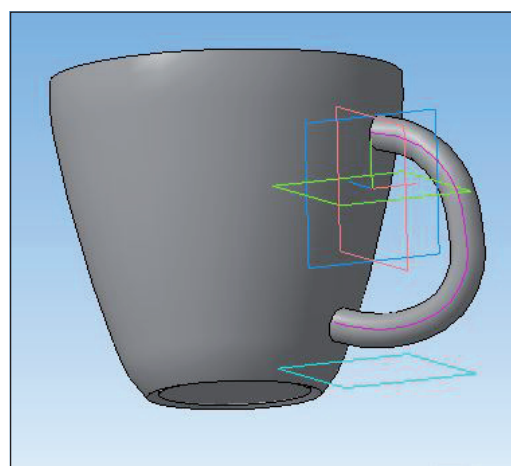


Рис. 30. Законченная модель чашки

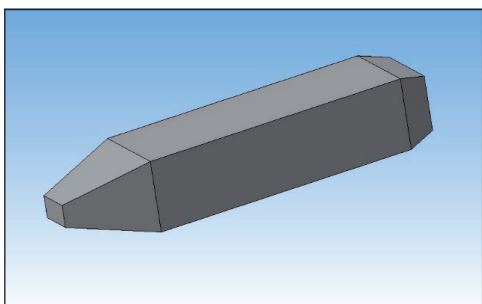


Рис. 31. Корпус космического корабля

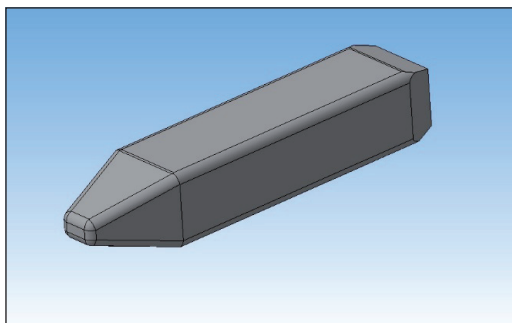


Рис. 32. Корпус корабля со скругленными формами

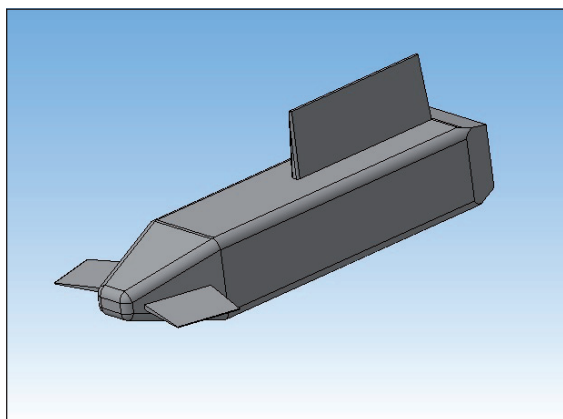


Рис. 33. Корпус корабля с установленными на него стабилизаторами

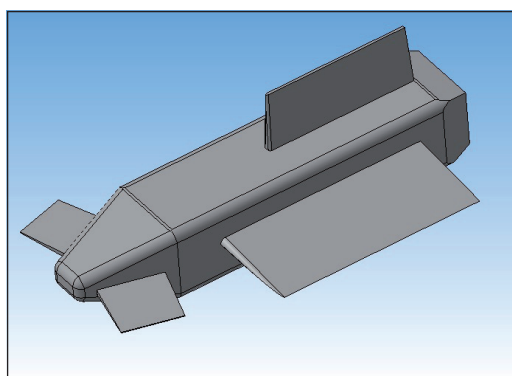


Рис. 34. Установлено левое основное крыло

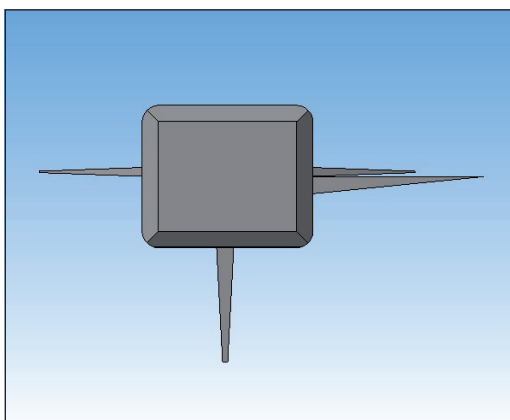


Рис. 35 Форма основного крыла (вид сзади)

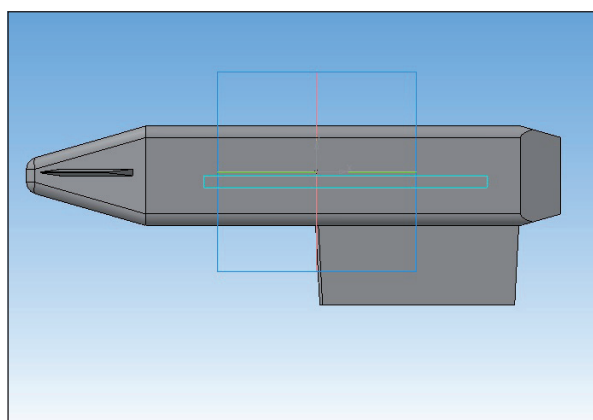


Рис. 36. Технология создания правого основного крыла, одинакового по геометрическим размерам с левым крылом

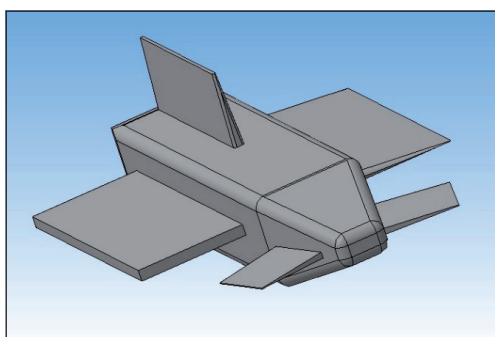


Рис. 37. Правое основное крыло без вырезанной клиновидной части

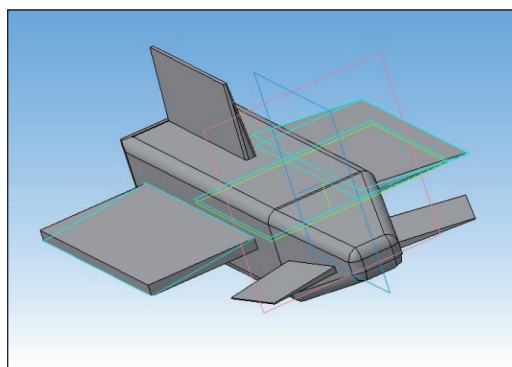


Рис. 38 Корабль с показанными вспомогательными плоскостями

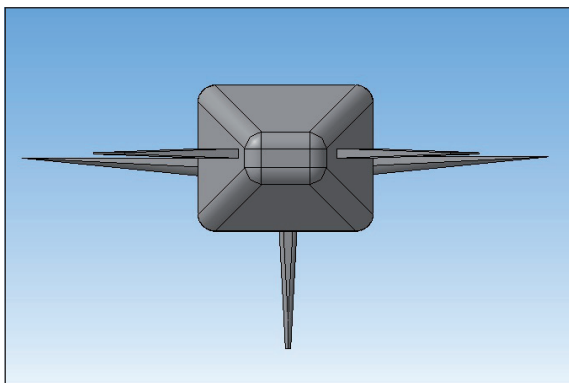


Рис. 39. Корабль с законченными основными крыльями (вид спереди)

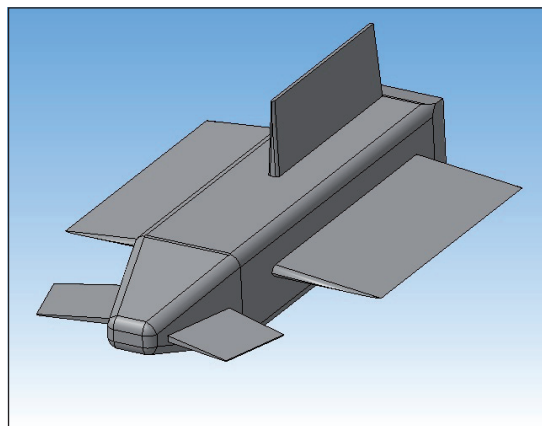


Рис. 40. Корабль с законченными основными крыльями (вид в изометрии)

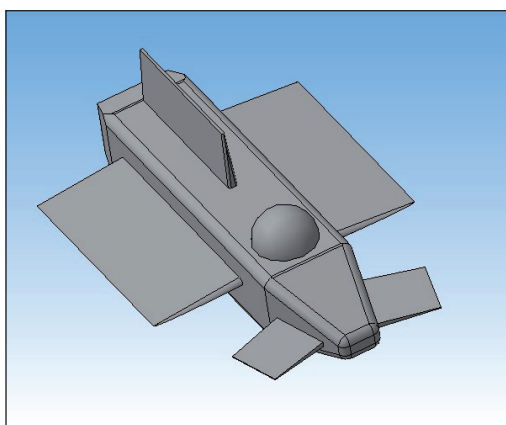


Рис. 41. Корабль с кабиной пилотов

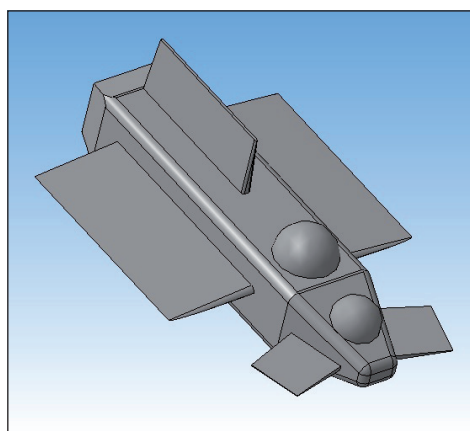


Рис. 42. Корабль с дополнительной кабиной

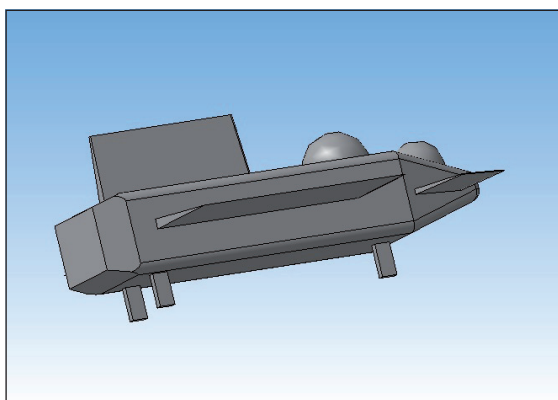


Рис. 43. Корабль с добавленными стойками шасси

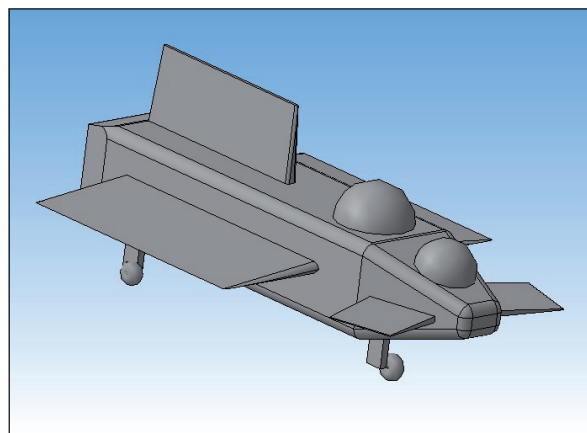


Рис. 44. Корабль с установленными на шасси колесами

5. Выводы

В статье предложена методика использования бесплатной версии отечественной системы трехмерного проектирования КОМПАС-3D LT V12 для поддержки преподавания темы «Трехмерная компьютерная графика» в курсе информатики основной школы или в рамках внеурочной деятельности (объединения дополнительного образования) по теме «Компьютерная графика».

Главной целью разработки и последующего использования данной методики является развитие объемного мышления и воображения учащихся основной школы.

При разработке предлагаемого в авторской методике порядка подачи учебного материала и соответствующего ему поурочного планирования по теме «Трехмерная компьютерная графика» в первую очередь учитывалось наличие в КОМПАС-3D LT V12 четырех типов операций создания объемных фигур, которые были следующим

образом расположены по мере возрастания сложности алгоритмов их реализации:

- 1) операция выдавливания некоторой плоской геометрической фигуры в пространстве;
- 2) операция вырезания выдавливанием некоторой плоской геометрической фигурой части большей трехмерной фигуры;
- 3) операция вращения некоторой плоской геометрической фигуры вокруг оси вращения;
- 4) кинематическая операция, заключающаяся в формировании трехмерной фигуры путем движения некоторой плоской геометрической фигуры вдоль траектории движения.

Пятым вариантом создания трехмерной фигуры в КОМПАС-3D LT V12 является построение модели по нескольким ее сечениям. Данная технология не рассматривается в статье в связи с некоторым избыточным для учащихся основной школы уровнем ее сложности, как и другой определенный функционал КОМПАС-3D LT V12 (сечение объемных фигур вспомогательными плоскостями и т. д.). Тем не менее преподаватели, планирующие использовать предлагаемый автором каркас методики, всегда могут дополнять его технологиями создания трехмерных объектов повышенной сложности, применяя для этого учебные пособия Н. Б. Ганина [6, 7] и А. А. Герасимова [8], а также «Азбуку КОМПАС-3D» [1].

Выбор отечественной системы трехмерного моделирования КОМПАС-3D LT V12 был обусловлен внушительным набором положительных характеристик для поддержки учебно-воспитательного процесса на уровне основного общего образования, как в техническом, так и в методическом плане.

Список источников

1. Азбука КОМПАС-3D // КОМПАС-3D. Официальный сайт САПР КОМПАС. <https://kompas.ru/source/info-materials/2018/Azbuka-KOMPAS-3D.pdf>
2. Александров М. С. Развитие пространственного мышления учащихся на элективных курсах посредством трехмерного графического редактора КОМПАС-3D // Информационные технологии в социальной сфере. Материалы V международной заочной научно-практической конференции. Самара: Самарский государственный социально-педагогический университет, 2017. С. 96–99.
3. Баранов В. Н. О применении системы Компас 3D в школе для обучения инженерной графике // Роль и место инженерных знаний в структуре общего образования. Сборник статей V межрегиональной очно-заочной научно-практической конференции с международным участием. СПб.: Лингвистический Центр «Тайкун», 2017. С. 146–148.
4. Большаков В. П. КОМПАС-3D для студентов и школьников. Черчение, информатика, геометрия. СПб.: БХВ-Петербург, 2010. 304 с.
5. Все альтернативы САПР AutoCAD для Linux // LinuxTheBest. <https://linuxthebest.net/top-luchshikh-cad-programm-dlya-linux-v-2018-godu/>
6. Ганин Н. Б. Автоматизированное проектирование в системе КОМПАС-3D V12. М.: ДМК Пресс, 2010. 360 с.
7. Ганин Н. Б. Создаем чертежи на компьютере в КОМПАС-3D LT. М.: ДМК Пресс, 2005. 184 с.
8. Герасимов А. А. Самоучитель КОМПАС-3D V12. СПб.: БХВ-Петербург, 2011. 464 с.
9. Зенкина С. В., Савченкова М. В. Использование 3D редакторов в урочной и во внеурочной деятельности // Информ. 2018. № 1 (2). С. 45–53.
10. Каковы системные требования для запуска FreeCAD? https://wiki.freecadweb.org/Frequently_asked_questions/ru
11. КОМПАС-3D LT. Системные требования // КОМПАС-3D. Официальный сайт САПР КОМПАС. <https://kompas.ru/kompas-3d-lt/restriction/>
12. Лицензионное соглашение на использование программного продукта КОМПАС-3D LT // КОМПАС-3D. Официальный сайт САПР КОМПАС. https://kompas.ru/source/documents/License_LT_ru_2012.pdf
13. Маркушевич М. В. Использование отечественного программного обеспечения в учебном процессе общеобразовательной школы как фактор гражданско-патриотического воспитания учащихся // Информатика в школе. 2018. № 7. С. 13–16.
14. Маркушевич М. В. Организация дистанционного преподавания робототехники на базе микроконтроллера Arduino Uno в виртуальной среде Autodesk Tinkercad // Информатика в школе. 2020. № 8. С. 12–20.
15. Маркушевич М. В. Полипрограммные интегрированные уроки на базе свободного программного обеспечения как средство контроля ИКТ-компетенций старших школьников // Информатика в школе. 2017. № 4. С. 12–17.
16. Математическая вертикаль — Ресурсный центр «Интеллектуал». <https://vertical.sch-int.ru/>
17. Нанософт разработка. Наши продукты // Нанософт разработка. <https://www.nanocad.ru/products/>
18. Операционная система Windows 10. Минимальные системные требования. <https://system-requirements.ru/os/3-sistemnye-trebovaniya-windows-10.html>
19. Перевод КОМПАС-3D на ОС Linux. Текущая ситуация и планы // АСКОН. https://ascon.ru/news_and_events/news/3314/
20. Приказ Министерства просвещения РФ от 31 мая 2021 г. № 287 «Об утверждении федерального государственного образовательного стандарта основного общего образования». <https://www.garant.ru/products/ipo/prime/doc/401333920/#1000>
21. Прояева И. В., Цветкова К. Е. Особенности использования геометрии в компьютерном моделировании // Проблемы и перспективы внедрения инновационных телекоммуникационных технологий. Сборник материалов VII Международной научно-практической очно-заочной конференции. Оренбург: Оренбургский филиал ФГБОУ ВО «Поволжский государственный университет телекоммуникаций и информатики», 2021. С. 347–356.
22. Свободная программа трехмерного параметрического моделирования FreeCAD // FreeCAD. <https://www.freecadweb.org/?lang=ru>
23. Система трехмерного моделирования КОМПАС-3D // КОМПАС-3D. Официальный сайт САПР КОМПАС. <https://kompas.ru/kompas-3d-home/buy/>
24. Тютина В. Поставщик ПО для строительства Autodesk приостановил работу в России // РБК. 4 марта 2022 года. <https://www.rbc.ru/politics/04/03/2022/6221b6209a7947f98925c8d7>
25. Федосов А. Ю., Маркушевич М. В. Использование свободной системы автоматического проектирования FreeCAD при преподавании стереометрии // XV Колмогоровские чтения: Сборник статей участников Международной научно-практической конференции, посвященной памяти профессора М. И. Зайкина (Арзамас, 10–13 сентября 2019 года). Арзамас: Арзамасский филиал федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский Нижегородский государственный университет им. Н. И. Лобачевского», 2019. С. 147–151.



DOI: 10.32517/2221-1993-2022-21-3-67-74

Н. Н. Самылкина*Московский педагогический государственный университет, г. Москва, Россия***И. А. Калинин***Московский государственный лингвистический университет, г. Москва, Россия*

РЕШЕНИЕ ЗАДАЧ ОТБОРОЧНОГО И ОСНОВНОГО ЭТАПОВ ПЕРВОЙ ВСЕРОССИЙСКОЙ ОЛИМПИАДЫ ПО ИСКУССТВЕННОМУ ИНТЕЛЛЕКТУ ДЛЯ УЧАЩИХСЯ VIII—XI КЛАССОВ

Аннотация

В статье рассматриваются подходы к разработке системы олимпиадных заданий, особенности проведения мероприятия, а также решения заданий двух этапов первой всероссийской олимпиады по искусственному интеллекту для VIII—XI классов, проведенной в октябре–ноябре 2021 года Министерством просвещения Российской Федерации в рамках Федерального проекта «Искусственный интеллект» национальной программы «Цифровая экономика Российской Федерации». Тематика заданий олимпиады — обработка текста на естественном языке (Natural Language Processing) и компьютерное зрение (Computer Vision). На всех этапах олимпиады использовался язык программирования Python со специализированными библиотеками для решения задач искусственного интеллекта. Материал с разбором заданий будет полезен при обучении информатике в школе на углубленном уровне, в том числе при подготовке школьников к участию в следующей олимпиаде по искусственному интеллекту в индивидуальном формате.

Ключевые слова: всероссийская олимпиада, искусственный интеллект, Python, специализированные библиотеки языка Python, углубленное изучение информатики.

1. Актуальность подготовки школьников в области искусственного интеллекта

Осенью 2021 года Министерством просвещения Российской Федерации в рамках Федерального проекта «Искусственный интеллект» национальной программы

«Цифровая экономика Российской Федерации» [5] была проведена первая всероссийская олимпиада по искусственному интеллекту (сайт олимпиады: <https://olimp.edsoo.ru/>). В России традиционно сильная олимпиадная подготовка школьников по многим предметам, в том числе по информатике. Минпросвещения России,

Контактная информация

Самылкина Надежда Николаевна, доктор пед. наук, доцент, профессор кафедры теории и методики обучения математике и информатике, Институт математики и информатики, Московский педагогический государственный университет, г. Москва, Россия; *адрес:* 119991, Россия, г. Москва, ул. Малая Пироговская, д. 1, стр. 1; *e-mail:* nsamylkina@yandex.ru

Калинин Илья Александрович, канд. пед. наук, доцент, начальник управления информатизации, доцент кафедры международной информационной безопасности, Институт информационных наук, Московский государственный лингвистический университет, г. Москва, Россия; *адрес:* 119034, Россия, г. Москва, ул. Остоженка, д. 38, стр. 1; *e-mail:* llyakalinin@mail.ru

N. N. Samylkina

Moscow Pedagogical State University, Moscow, Russia

I. A. Kalinin

Moscow State Linguistic University, Moscow, Russia

SOLVING THE PROBLEMS OF THE PRELIMINARY AND MAIN STAGES OF THE FIRST ALL-RUSSIAN OLYMPIAD IN ARTIFICIAL INTELLIGENCE FOR STUDENTS OF 8TH—11TH GRADES

Abstract

The article discusses approaches to developing a system of Olympiad tasks, features of the event, as well as solving tasks of two stages of the first All-Russian Olympiad in artificial intelligence for 8th—11th grades, held in October–November 2021 by the Ministry of Education of the Russian Federation as part of the Federal Project "Artificial Intelligence" of the National Program "Digital Economy of the Russian Federation". The topics of the Olympiad tasks were Natural Language Processing and Computer Vision. At all stages of the Olympiad, the Python programming language with specialized libraries for solving artificial intelligence problems was used. The material with the analysis of tasks will be useful when teaching informatics at an advanced level at school, including when preparing schoolchildren to participate in the next Olympiad in artificial intelligence in an individual format.

Keywords: All-Russian Olympiad, artificial intelligence, Python, specialized Python libraries, advanced study of informatics.

инициируя новую олимпиаду, ставило целью создание условий для поддержки одаренных школьников, а именно: выявление и развитие творческих способностей, развитие познавательного интереса обучающихся, повышение у них мотивации к углубленному изучению информатики и математики как основе предпрофессиональной подготовки в области информационных технологий.

Тематика искусственного интеллекта как сквозной цифровой технологии поддерживается государственными структурами и активно отражается в научных публикациях и учебной литературе (см., например, [1–4, 6, 7, 9, 10]). Представление этой темы расширилось в федеральных государственных образовательных стандартах общего образования всех уровней. Готовится учебно-методическая литература для обучающихся и педагогов по искусственному интеллекту. Тематика искусственного интеллекта включена в обязательную подготовку будущих учителей информатики на уровне как бакалавриата, так и магистратуры. Несколько лет проводятся вузовские олимпиады по искусственному интеллекту для школьных команд. Появляются доступные для школьников цифровые ресурсы для подготовки к олимпиадам по искусственному интеллекту.

Всероссийская олимпиада по искусственному интеллекту для VIII–XI классов общеобразовательных организаций проводилась в октябре–ноябре 2021 года впервые, и льготы для ее победителей при поступлении в вузы еще не предусматривались.

Разработчики заданий выбрали наиболее актуальные в настоящее время направления искусственного интеллекта — обработка текста на естественном языке (Natural Language Processing) и компьютерное зрение (Computer Vision).

На всех этапах олимпиады использовался язык программирования Python. Это пока единственный язык программирования, имеющий специализированные библиотеки для решения задач искусственного интеллекта.

2. Особенности проведения олимпиады. Система заданий

Первая всероссийская олимпиада по искусственному интеллекту для VIII–XI классов **состояла из трех этапов и проводилась в дистанционном режиме.**

Для обеспечения прозрачности контингента участников предполагалась их регистрация через образовательные организации, которые своим приказом направляют своих обучающихся для участия в олимпиаде. При этом допускалась возможность доступа для учащихся к контенту олимпиады на площадках образовательных организаций при консультативной поддержке учителей информатики.

В олимпиаде рассматривались индивидуальные решения заданий, т. е. это были **индивидуальные соревнования.**

Решенные задания участников проходили **полностью автоматизированную проверку**, при этом участник не видел результата выполнения своей программы, а видел только оценку соответствия ее результатов предусмотренным критериям по одному из двух вариантов:

- 1) количество правильно пройденных тестовых наборов данных (входные данные, предусмотренные в условии задания);
- 2) процентное соотношение результатов работы программы реальному набору данных.

Содержательно олимпиадные задания всех этапов были тематически и инструментально взаимосвязаны, что позволяло каждому участнику нарастить свои тематические и программистские навыки.

На первом, отборочном, этапе олимпиады предлагалось 10 заданий для проверки навыков программирования на Python. Проверялись навыки:

- ввода данных и вывода результатов при помощи файлов;
- обработки числовой и текстовой информации, в том числе работы с различными кодировками;
- планирования структуры хранения данных для последующей обработки этих данных.

Можно было использовать только стандартные библиотеки языка.

Более половины заданий были ориентированы на использование регулярных выражений при обработке текста — именно этот ключевой инструмент требуется при решении множества задач на обработку текста. Поэтому на первом этапе проверялись навыки использования регулярных выражений пока без использования специализированных библиотек [8].

Задания можно было решать в любом порядке и направлять для проверки в любое время в течение пяти дней.

На втором, основном, этапе олимпиады предлагалось пять заданий на использование специализированных библиотек: Numpy, Pandas, Pillow, sklearn, NLTK, Apyory.

Тематика заданий расширилась — к заданиям на обработку текста добавились задания на компьютерное зрение и анализ данных.

Проверялись навыки грамотного использования указанных библиотек, ориентированных на работу с числовыми, текстовыми и графическими данными, с ориентацией на подготовку и анализ наборов данных для последующего решения задач интеллектуальной обработки.

Все алгоритмы задач требовали не только наличия у участников программистских навыков, но и вдумчивого анализа, подбора и настройки методов выделения и анализа данных.

К тому же изменился регламент работы: решения необходимо было направить на проверку в любой из пяти соревновательных дней, но в один непрерывный временной промежуток — четыре часа.

Таким образом, **к третьему, заключительному, этапу** участники были подготовлены к необходимости самостоятельной разработки программной модели обработки данных в соответствии с заданием.

Участникам было предложено два задания по темам «Компьютерное зрение» (сегментация изображений) и «Обработка текста на естественных языках» (задача оценки тональности).

Подготовка и обучение каждой из моделей могли занимать по три часа, поэтому участникам было дано достаточное время для необходимой корректировки мо-

делей — 12 часов непрерывной работы в течение любых пяти соревновательных дней.

Перед проведением очередного этапа олимпиады и после завершения соревновательных дней проводились вебинары для участников олимпиады. На этих встречах разработчики заданий рассматривали типовые задания, рациональные приемы решения и отвечали на вопросы участников. Условия всех задач и записи вебинаров можно посмотреть на сайте олимпиады.

В данной статье мы постарались разобрать задания, решения к которым не представлены на сайте олимпиады.

3. Разбор и решение задач отборочного и основного этапов олимпиады

Рассмотрим решение двух задач первого, отборочного, этапа (задачи 1 и 2).

Задача 1.

Лена и Вася собираются в Красивый Город на восемь дней и имеют на руках некоторую сумму денег (пока они не знают, сколько).

Они хотят посетить самые интересные места. О каждом месте они знают, насколько оно интересно (в баллах) и сколько стоит его посетить (в деньгах), — список есть в файле, имя которого вводится с клавиатуры.

К сожалению, они могут посещать только два места в день, а в списке — до 24 мест.

Разработайте программу, которая получит с клавиатуры максимальную сумму, имя файла-справочника и выведет на экран список мест, которые им стоит посетить, — максимальной интересности (в сумме баллов) и допустимо по деньгам. Места в списке надо перечислить через точку с запятой, в том числе в конце списка. Порядок вывода мест должен совпадать с порядком в исходном списке.

Тест работоспособности.

Ввод:

```
8000
test1.txt
```

Вывод:

```
Александр-Невская Лавра;Большой каскад
в Петергофе;Дворец «Марли»;Дворец
Монплеизир;Домик Петра I;Крейсер «Аврора»;
Летний дворец Петра I;Мариинский дворец;
Памятник Медный всадник;Смольный собор
Санкт-Петербурга; Собор Петра и Павла
в Петергофе;Спас на Крови;Чижик-пыжик;
```

Содержание файла test1.txt:

```
Александринский театр;8;1500
Александр-Невская Лавра;7;300
Большой каскад в Петергофе;8;100
Государственный Русский музей;10;2000
Дворец «Марли»;8;1000
Дворец Монплеизир;8;1000
Домик Петра I;8;1000
Зимний дворец;10;2000
```

```
Исаакиевский собор;9;1500
Казанский собор в Петербурге;9;1500
Крейсер «Аврора»;8;1000
Кунсткамера, или Кабинет редкостей;9;3000
Летний дворец Петра I;8;1000
Мариинский дворец;7;1000
Мариинский театр;9;4000
Михайловский дворец;7;1500
Музей Достоевского;8;2000
Памятник Медный всадник;8;100
Петропавловская крепость;9;2000
Смольный собор Санкт-Петербурга;8;200
Собор Петра и Павла в Петергофе;7;300
Спас на Крови;8;500
Чижик-пыжик;8;50
Эрмитаж;10;3500
```

Разбор решения задачи.

Задача представляет собой вариант классической переборной задачи, которую в олимпиадной литературе называют «задачей о дискретном рюкзаке».

Общий принцип решения задачи несложен: мы перебираем все возможные варианты, откидываем неподходящие варианты, а подходящие сравниваем по заданному критерию.

Самая простая часть задачи, практически не требующая пояснений, — это чтение файла с вариантами:

```
maxPrice = int(input(''))
fileName = input()

places = []
count = 0

with open(fileName, 'r', encoding='utf-8') as f:
    for line in f:
        stringDescribe = line.strip().split(';')
        places.append([stringDescribe[0],
                        int(stringDescribe[1]),
                        int(stringDescribe[2])])
        count = count + 1
```

Мы просто создаем список мест, в котором каждое место описывается списком из трех элементов: названием, количеством баллов, стоимостью.

Для организации перебора мы воспользуемся простым фактом: каждое место мы либо посещаем, либо нет. То есть факт посещения, очевидно, кодируется одним битом. Тогда мы можем просто договориться, что возьмем целое число, в котором каждый бит будет обозначать одно место — по номеру позиции.

Тогда полный перебор будет просто процедурой перебора всех чисел, которые уместятся в заданное количество бит (начиная с 1 — все-таки в одно место мы точно заедем). В нашем условии — максимум 24 места, т. е. 24 бита.

Остается подготовить функцию оценки каждого варианта.

В функции нам прежде всего надо будет оценить, не слишком ли много мест в предложенном варианте.

Для ускорения этого процесса подготовим небольшой массив, в котором каждому варианту значения 1 байта соответствует количество заполненных в нем битов:

```
import array

BitCount8 = array.array('I')

for i in range (256):
    BitCount8.extend( [(1 if i & 1 > 0 else 0)+
                      (1 if i & 2 > 0 else 0)+
                      (1 if i & 4 > 0 else 0)+
                      (1 if i & 8 > 0 else 0)+
                      (1 if i & 16 > 0 else 0)+
                      (1 if i & 32 > 0 else 0)+
                      (1 if i & 64 > 0 else 0)+
                      (1 if i & 128 > 0 else 0)])
```

Теперь, собственно, сама функция:

```
def calculateVariant ( varNum ) :
    global BitCount8 # Подготовленный заранее
                        массив для подсчета
    global places     # Подготовленный заранее
                        список мест
    global maxPrice

    current = 1
    price = 0
    ball = 0
    i=0
    placeCount = BitCount8[ varNum & 0xFF ]+ \
                BitCount8[ (varNum >> 8)& 0xFF ]+ \
                BitCount8[ (varNum >> 16)& 0xFF ]
    if placeCount > 16: # Не более 2 мест в день,
                        в течение 8 дней
        return (0,0)
    while current <= varNum and price <= maxPrice:
        if varNum & current != 0:
            price += places[i][2]
            ball += places[i][1]
        i += 1
        current = current << 1
    if price > maxPrice:
        return (0,0)
    return (price, ball)
```

Функция получит номер (фактически — описание) варианта посещения разных мест и вернет кортеж из двух чисел — стоимость варианта и «интересность» в баллах.

В этой функции мы начинаем с подсчета бит, равных 1, т. е. с подсчета мест, которые в этом варианте надо посетить. Сделаем это с помощью битовых операций, выбрав байты и сложив количество битов в них (берем количество из массива).

Если их больше 16 — возвращаем кортеж (0,0), т. е. вариант не рассматриваем и время на подсчеты не тратим.

Теперь в цикле перебираем все биты, начиная с первого (current = 1), пока не дойдем до конца, т. е. пока число с установленным «текущим» битом не станет больше номера варианта или пока не будет превышен максимум стоимости.

Если логическая операция «И» с этим битом нашего варианта дает не нулевое значение — место надо учесть, т. е. добавить его баллы и его стоимость к накопленной оценке.

В основном коде программы перебор довольно прост:

```
maxThings = 1 << count # Максимальный вариант
                        2 в степени количество

curVar = 1
maxBalls = 0
bestVariant = 0

while curVar < maxThings:
    curPrice, curBalls =
        calculateVariant ( curVar )
    if curPrice <= maxPrice and
        curBalls > maxBalls:
        bestVariant = curVar
        maxBalls = curBalls
    curVar += 1
```

В итоге мы выбираем вариант с максимальным «уровнем интереса», заранее оценив как «совсем не интересные» варианты, не соответствующие условию: слишком долгие или слишком дорогие.

Остается вывести найденный вариант, в соответствии с условием задачи:

```
pos = 0
while ( 1 << pos) <= bestVariant:
    if bestVariant & (1 << pos) > 0:
        print( places[pos][0], end=';')
    pos += 1
```

Задача 2.

В издательство прислали два произведения разных авторов в жанре фантастики с множеством уникальных слов. Требуется выявить совпадения по уникальным словам. Программисту издательства поставили эту задачу в следующей формулировке:

«Даны имена двух больших (до 10 Мбайт) текстовых файлов в кодировке UTF-8 с текстом на русском языке.

Написать программу, которая введет с клавиатуры имена файлов и выдаст на экран списки 50 уникальных слов первого файла (по убыванию частоты) и 50 уникальных слов второго файла (по убыванию частоты), используя не более чем 5 Мбайт памяти.

Слова вывести через один пробел».

Тест работоспособности.

Ввод:

```
Test0-1.txt
Test0-2.txt
```

Пример вывода:

было эти записки у руки условиях частично почерка
эксперт его хранились дело всякий таких папке
машинописный написанный от документ текст которых
разными многие в-листы кое-где оба двух запись
уверяет он левши то есть ли были переданы лицом
пожелавшим чтобы предавались гласности покуда
будет жив передавший указанное лицо изложило
мотивы пожелания
записок автора имя такого имени всех трех
названия наибольшую сложность представляло
приведение форме последовательного изложения

разработать применить приемов алгоритмизации приведения сведения приведенные документах настолько недостаточны или два почувствовал себя обязанным сделать вставки дабы пояснить канву хотя меру понимания приношу покорнейшие извинения читателям пронизательным теперь об названиях текстах упоминается разу почти

Разбор решения задачи.

Задача несложная, но требующая некоторых навыков организации анализа текстовых файлов.

Начнем с ввода исходных данных и подготовки к работе:

```
import re

firstWord = {}
secondWord = {}
firstFileName = input()
secondFileName = input()
```

Мы будем использовать для деления текста на слова механизм регулярных выражений, а сами слова хранить в двух словарях — для ускорения проверки.

Сразу введем имена файлов, чтобы не столкнуться с потерей второго имени из-за времени обработки первого файла.

Теперь создадим словарь слов первого файла:

```
with open(firstFileName, 'r', encoding='utf-8')
    as firstFile:
    for line in firstFile:
        line = line.strip().lower()
        for fword in re.findall("[A-яa-я]+")
            ([-]{1}[A-яa-я]+)?", line):
            word = ''.join(fword) # Объединяем
                                группы
            if not (word in firstWord):
                firstWord[word] = 1
            else:
                firstWord[word] =
firstWord[word] + 1
```

С точки зрения формальной обработки файла мы считаем «словом» просто последовательность символов между разделителями. Обратите внимание: в русском языке есть двойные слова, которые записываются через дефис (напомним, что дефис — это разделение именно частей слова). К сожалению, в текстовых файлах, особенно старых, тире и дефис записываются одним и тем же символом, так что будем считать, что если есть знак «-» без пробелов вокруг, то это дефис.

Наше регулярное выражение отбирает именно русские слова (т. е. слова из букв русского алфавита) и предполагает, что в слове может быть несколько групп — как минимум одна и еще несколько через тире. Поэтому мы находим вхождения этого шаблона — регулярного выражения, а потом объединяем все группы в одно слово.

Дальше мы просто проверяем: если слово в словаре есть, мы увеличиваем счетчик его вхождений, если нет — добавляем в словарь.

С анализом второго файла все немного сложнее. Встретив слово во втором файле, мы должны проверить, встречается ли оно в первом. Если встречается — оно не

уникально для обоих файлов. Но просто удалить слово из первого словаря и не добавлять во второй недостаточно, потому что оно может появиться и второй раз, и третий, и т. д. Поэтому мы заводим еще один словарь — словарь удаленных слов, куда и будем помещать удаляемые слова из первого словаря, и при анализе проверим не один, а два словаря.

```
with open(secondFileName, 'r', encoding='utf-8')
    as secondFile:
    for line in secondFile:
        line = line.strip().lower()
        for fword in re.findall("[A-яa-я]+")
            ([-]{1}[A-яa-я]+)?", line):
            word = ''.join(fword)
            if word in firstWord: # Было ли
                                слово в первом файле?
                deletedWord.append(word) # Если
                                да - оно не уникально
                del firstWord[word] # ... и из
                                словаря его убрать
            else:
                if not word in deletedWord:
                    if word in secondWord:
                        secondWord[word] =
                        secondWord[word] + 1
                    else:
                        secondWord[word] = 1
```

Словари сформированы, остается их отсортировать и вывести в соответствии с условием:

```
sortedFirst = [k for k, v in
                sorted(firstWord.items(), \
                        key=lambda item: item[1], reverse=True)]
print(' '.join(sortedFirst[0:50]))
sortedSecond = [k for k, v in
                 sorted(secondWord.items(), \
                        key=lambda item: item[1], reverse=True)]
print(' '.join(sortedSecond[0:50]))
```

Приводить в статье содержимое проверочных файлов было бы затруднительно, работу программ читатель вполне может проверить на своих примерах.

Далее рассмотрим решение двух задач второго этапа (задачи 3 и 4).

Задача 3.

Имеется база отзывов на фильмы в виде общего каталога с текстовыми файлами, разделенными внутри на три каталога — pos, neg и neu. В каждом из этих каталогов несколько (> 10) текстовых файлов на русском языке в кодировке UTF-8. Выведите через один пробел список из 50 самых информативных слов, т. е. тех, которые лучше всего позволяют различать, к какой категории относятся отзывы. Информативность слов оценивают с помощью критерия взаимной информации (https://ru.abcdef.wiki/wiki/Mutual_information#Conditional_mutual_information).

Тест работоспособности.

Ввод:

dataset

Вывод:

так это как части же теперь плохо они но бы он эпизод меликян исключительно общество никакого того между очень или веры мессия чернобыль аварии из серии его показать во меня вот ощущение где начинается достойно сделал лжи смогли место от чтобы вы для до интервью ради фильм есть может которая

Разбор решения задачи.

Несмотря на пугающий вид формул, приведенных по указанной в условии задачи ссылке, общая идея «взаимной информации» довольно проста: если у нас есть два набора данных со сходными наборами признаков, то мы можем определить, какие из этих признаков вносят больший вклад в отделение одного набора от другого.

В случае наборов текстов такими признаками являются слова. Очевидно, что, если слово встречается в одном наборе текстов 50 раз, а в другом — 51, оно малоинформативно, а вот если в одном наборе оно встречается 11 раз, а во втором — 205, очевидно, что слово гораздо более информативно.

Выглядит нужная формула довольно громоздко, но реализовывать ее вычисление самостоятельно и не требовалось — вычисление необходимой характеристики есть в составе библиотеки ScikitLearn.

Основная цель задачи — позволить участникам (тем, кто не поленится поискать необходимую информацию и решить «скучную» задачу) подготовиться к решению одной из задач третьего тура. В частности, изучив набор данных.

Начинаем, как всегда, с подключения библиотек:

```
import re
import os
from sklearn.feature_selection
    import mutual_info_classif
from sklearn.feature_extraction.text
    import CountVectorizer
```

Определим регулярное выражение, выделяющее слова:

```
wordSplit = re.compile(r"[А-Яа-я]+")
```

Оно проще, чем то, что мы использовали ранее, но для выделения наиболее информативных слов вполне подойдет (можно взять более сложное, но на результат это не повлияет — двойные слова встречаются сравнительно редко).

Каждый файл нам надо будет представить в виде единой текстовой строки слов. Файлов много, так что напишем для этого несложную функцию:

```
def preprocess_file( filename, enc):
    words = ''

    for string in open(filename, 'r', encoding=enc):
        for word in re.
            findall(wordSplit, string):
                words = words + ' '+word
    return words.strip()
```

Как видно, функция получает имя файла, открывает этот файл, выделяет из него слова и собирает их в длин-

ную строку. Строка будет для нас частью нашего набора данных в последующей обработке.

```
basePath = input()
paths = ['pos', 'neg', 'neu']
corpus = []
target = []

for path in paths:
    for file in os.listdir('./'+basePath+'/'+path):
        corpus.append(
            preprocess_file('./'+basePath+'/'+path+'/'+
                file, "utf-8")
        )
        target.append(path)
```

В основной программе мы анализируем наш набор данных, т. е. готовим корпус текстов (тех самых строк) и отдельно список классов, к которым каждый текст относится. Поскольку файлов много и нам дают только название корневого каталога, мы организуем получение имен файлов из каждого подкаталога — по имени класса. Обрабатываем файл и добавляем его в список corpus, а его класс — в список target.

Чтобы преобразовать корпус к нужному нам виду, т. е. к матрице, где каждый документ — это отдельная строка, состоящая из количества вхождений каждого слова (те слова, которых нет в конкретном документе, имеют 0 вхождений), мы воспользуемся векторизатором (средство превращения документа в вектор, т. е. в набор чисел), который и обучим на нашем корпусе:

```
cv = CountVectorizer(max_df=0.95, min_df=2)
corpus_vec = cv.fit_transform(corpus)
```

Векторизатору мы указываем, что не будем обращать внимания на слова, которые встречаются в 95 % документов корпуса или встречаются всего один раз, — информативность у таких слов гарантированно низкая.

После этого мы можем обучить собственно классификатор взаимной информации:

```
res = dict(zip(cv.get_feature_names(),
    mutual_info_classif(corpus_vec, target,
        discrete_features=True)
    ))
```

Результат мы собираем в виде словаря, в котором каждому слову будет соответствовать информативность. Сами слова мы получим уже как признаки из подготовленного нами векторизатора: cv.get_feature_names() и соберем в единый словарь с показателями их информативности (функция zip создаст список парных кортежей, который мы и преобразуем в словарь).

Остается отсортировать словарь и выдать результат:

```
wordsGain = [k for k, v in sorted(res.items(),
    key=lambda item: item[1], reverse=True)]

print(' '.join(wordsGain[0:50]))
```

Задача 4.

Есть фото стенки до начала покраски и во время покраски, в формате JPEG.

Написать программу, которая получит с клавиатуры название файла с участком непокрашенной стены, название файла с покрашенной стеной и определит, сколько процентов стены уже покрашено. Учтите, что ровным и точным цвет не бывает.

Ответ вывести в целых процентах, возможно отклонение до 2 % включительно. Знак «%» выводить не надо.

Тест работоспособности.

Ввод:

wall1-1.jpg

wall1.jpg

Функция оценки ответа:

```
lambda answer: 1 if abs(answer-74)<2 else 0
```

Разбор решения задачи.

Очевидная (нехарактерная для обычного школьного курса информатики) особенность этой задачи — работа с графической информацией, причем не со сгенерированным простым примером, а с участком реальной фотографии. При решении данной задачи нам придется учесть, что в реальности при покраске стены стена не становится ровной, а значит, и о ровном точном цвете говорить нельзя.

Тем не менее изображение, как мы знаем, представляется в виде матрицы пикселей. Если сформулировать решение задачи, то можно сказать, что наша задача сводится к подсчету «покрашенных» и «непокрашенных» точек.

Если мы посмотрим на примеры такой частично покрашенной стены (достаточно воспользоваться поиском в сети), то можно отметить очевидный факт — недокрашенная стена делится на две части, которые заметно отличаются друг от друга. Таким образом, нам нужно найти способ разделить пиксели на два класса и выяснить, какой из них больше похож на ранее данный пример.

Способов это сделать достаточно много, но быстрее всего будет исходить из того, что возможности «учить» такой классификатор у нас не будет (фото очень разнообразны), так что выберем простой метод обучения без учителя.

Самый известный метод, который позволяет быстро разделить множество объектов на классы, — метод К-средних. Суть его в том, что мы делим имеющиеся у нас объекты на заданное количество групп и каждую описываем наиболее характерной «средней» точкой.

Для этого прежде всего подключим нужные библиотеки:

```
from PIL import Image
import numpy as np
import math
from sklearn.cluster import KMeans
```

Используем для работы с изображениями библиотеку Pillow (она предусматривалась правилами), а для разбиения на кластеры — класс KMeans из библиотеки ScikitLearn. Сначала загрузим изображения:

```
imgBaseFile = Image.open(input()) # Исходная
                                стенка
img = Image.open(input()) # Покрашенная стенка
```

Теперь найдем, исходя из условия, «средний» пиксель для неокрашенной стены:

```
colorPixelBase = [0,0,0,0]
for i in range(imgBaseFile.size[0]):
    for j in range(imgBaseFile.size[1]):
        pixel = imgBaseFile.getpixel((i, j))
        colorPixelBase[0] += pixel[0]
        colorPixelBase[1] += pixel[1]
        colorPixelBase[2] += pixel[2]
        colorPixelBase[3] += 1
```

Для этого создадим список из четырех элементов — суммы трех базовых цветов и общего количества пикселей. Используя Pillow, получим каждый пиксель и прибавим его к накопленным значениям. После чего останется усреднить компоненты:

```
colorPixelVector =
    ( colorPixelBase[0]/colorPixelBase[3],
      colorPixelBase[1]/colorPixelBase[3],
      colorPixelBase[2]/colorPixelBase[3]
    )
```

Далее сформируем массив пикселей уже второй картинки. Его не получится просто усреднить — его нужно будет делить на кластеры. Поэтому мы сформируем список пикселей (мы выделяем кластеры, исходя только из цвета, поэтому положение не важно) и преобразуем его в массив библиотеки Numpy (именно с такими массивами работает KMeans):

```
colorPixel = []
for i in range(img.size[0]):
    for j in range(img.size[1]):
        pixel = img.getpixel((i, j))
        colorPixel.append(pixel)
colors = np.array( colorPixel )
```

Теперь можем выделить нужное количество кластеров (выучить модель):

```
kmeans = KMeans(n_clusters = 2)
kmeans.fit(colors)
```

Получив два кластера, мы можем посчитать, сколько пикселей в каждом из них. Для этого создадим словарь и посчитаем количество упоминаний каждого полученного кластера:

```
clusterSize = {}
for label in kmeans.labels_:
    if label in clusterSize:
        clusterSize[label] += 1
    else:
        clusterSize[label] = 1
```

Но поскольку нам надо сравнивать с образцом, то необходимо выяснить, кто из них кто. Самый очевидный способ это сделать — посчитать евклидово расстояние: кто ближе в пространстве цветов к нашему образцу, тот и представляет неокрашенную стенку.

```
minDistanse = 255 * 3**0.5
label = -1
order = 0
```

```

for l in kmeans.cluster_centers_:
    distance = ((l[0]-colorPixelVector[0])**2+
                (l[1]-colorPixelVector[1])**2+
                (l[2]-colorPixelVector[2])**2 )**0.5
    if distance < minDistance:
        minDistance = distance
        label = order
    order += 1

```

После этого остается только вычислить и вывести соотношение:

```

print( math.floor((1 - clusterSize[label] /
                  (clusterSize[0]+clusterSize[1]))*100 ) )

```

Конечно, предложенный метод для «промышленного» решения малопригоден — он медленный, не может учесть наличие посторонних объектов и т. д. Но для быстрого решения поставленной узкой задачи вполне подходил.

Как вы видели в формуле оценки ответа, для оценки результат сравнивался с ручным подсчетом, выполненным с помощью программы обработки графической информации.

4. Заключение

Проведенная первая всероссийская олимпиада по искусственному интеллекту для VIII—XI классов показала повышенный интерес учащихся и педагогов к вопросам искусственного интеллекта, а также высокий

потенциал языка программирования Python для использования в качестве основного языка для олимпиады по искусственному интеллекту.

Список источников

1. Богданова А. Н. Элективный курс «Основы искусственного интеллекта» для учащихся старших классов // Информатика в школе. 2021. № 7. С. 27–33.
2. Брусард М. Искусственный интеллект: пределы возможного / пер. с англ. М.: Альпина нон-фикшн, 2020. 362 с.
3. Калинин И. А., Самылкина Н. Н. Информатика. Углубленный уровень. 11 класс. М.: БИНОМ. Лаборатория знаний, 2013. 216 с.
4. Пратик Дж. Искусственный интеллект с примерами на Python / пер. с англ. СПб.: Диалектика, 2019. 448 с.
5. Программа «Цифровая экономика Российской Федерации». Утв. распоряжением Правительства Российской Федерации от 28.07.2017 г. № 1632-р. <http://static.government.ru/media/files/9gFM4FHj4PsB79I5v7yLVuPgu4bvR7M0.pdf>
6. Самылкина Н. Н., Салахова А. А. Обучение основам искусственного интеллекта и анализа данных в курсе информатики на уровне среднего общего образования: монография. М.: МПГУ, 2022. 228 с.
7. Стоянов С., Глушкова Т., Тодоров Й. Искусствен интеллект. Решаване на проблеми посредством търсене. София: Изкуства, 2019. 120 с.
8. Фридл Дж. Регулярные выражения. 3-е изд. / пер. с англ. СПб.: Символ-Плюс, 2008. 608 с.
9. Ясницкий Л. Н. Введение в искусственный интеллект: учебное пособие для вузов. 2-е изд., испр. М.: Академия, 2008. 176 с.
10. Ясницкий Л. Н. Искусственный интеллект. Элективный курс: учебное пособие. М.: БИНОМ. Лаборатория знаний, 2012. 197 с.



DOI: 10.32517/2221-1993-2022-21-3-75-82

О. А. Завьялова, В. К. Маркелов

Шуйский филиал Ивановского государственного университета, г. Шуя, Ивановская область, Россия

ВОЗМОЖНОСТИ ОНЛАЙН-СРЕД ПРОГРАММИРОВАНИЯ ПРИ ОБУЧЕНИИ ЯЗЫКУ PYTHON В ШКОЛЕ

Аннотация

В рамках преподавания программирования в школьном курсе информатики важным вопросом является возможность использования различных онлайн-сред программирования, которые могут применяться при обучении языку программирования Python. Онлайн-среды программирования — это специализированные веб-сервисы, которые предоставляют возможности для написания и запуска программ на соответствующем языке программирования.

В статье приводится анализ содержания учебников и учебных пособий по предмету «Информатика» в контексте описания авторами возможности использования онлайн-сред при обучении языку Python. Рассматриваются возможности онлайн-сред программирования для работы с интерактивным и программным режимом Python. Описываются функциональные возможности четырех сервисов: Python Shell, Online Python, JDoodle Online Python 3 IDE, Trinket Python. Приводится демонстрация их работы на примерах программ на языке Python. Анализируются достоинства и недостатки данных онлайн-сред в рамках изучения раздела «Программирование» в школьном курсе информатики. Дается сравнительная характеристика рассмотренных онлайн-сред программирования. Описываются возможности их применения в качестве средства электронной поддержки при проведении уроков в рамках изучения языка программирования Python.

Ключевые слова: Python, программирование, онлайн-среда программирования, методика преподавания программирования, информатика.

1. Введение

Курс программирования является одним из основных элементов содержательной линии «Алгоритмы и программирование», изучаемой в рамках школьного предмета «Информатика». В соответствии с примерной рабочей программой основного общего образования по информатике (базовый уровень) к основным задачам

учебного предмета «Информатика» относится формирование у обучающихся знания основных алгоритмических структур, а также умений и навыков составления простых программ по построенному алгоритму на одном из языков программирования высокого уровня [6, с. 6].

Одним из основных языков для обучения программированию в школе является Python. Python — это мощный язык программирования высокого уровня, который

Контактная информация

Завьялова Ольга Алексеевна, канд. пед. наук, доцент кафедры математики, информатики и методики обучения, Шуйский филиал Ивановского государственного университета, г. Шуя, Ивановская область, Россия; *адрес:* 155908, Россия, Ивановская область, г. Шуя, ул. Кооперативная, д. 24; *e-mail:* ooolga30@yandex.ru

Маркелов Валерий Константинович, студент 1-го курса магистратуры по направлению подготовки 44.04.01 «Педагогическое образование» (профиль «Информационные технологии в профессиональной деятельности педагога»), Шуйский филиал Ивановского государственного университета, г. Шуя, Ивановская область, Россия; *адрес:* 155908, Россия, Ивановская область, г. Шуя, ул. Кооперативная, д. 24; *e-mail:* v.a.l.e.m.a.r.k@yandex.ru

O. A. Zavyalova, V. K. Markelov

Shuya branch of Ivanovo State University, Shuya, Ivanovo Region, Russia

THE POSSIBILITIES OF ONLINE PROGRAMMING ENVIRONMENTS FOR TEACHING PYTHON AT SCHOOL

Abstract

As part of teaching programming in a school informatics course, an important issue is the possibility of using various online programming environments that can be used when teaching the Python programming language. Online programming environments are specialized web services that provide the ability to write and run programs in the appropriate programming language.

The article provides an analysis of the content of textbooks and manuals on the subject of "Informatics" in the context of the authors' description of the possibility of using online environments when teaching the Python language. The possibilities of online programming environments for working with the interactive and program mode of Python are considered. The functionality of four services is described: Python Shell, Online Python, JDoodle Online Python 3 IDE, Trinket Python. A demonstration of their work is given on examples of programs in the Python language. The advantages and disadvantages of these online environments are analyzed as part of the study of the "Programming" section in the school informatics course. A comparative description of the considered online programming environments is given. The possibilities of their use as a means of electronic support during lessons in the framework of studying the Python programming language are described.

Keywords: Python, programming, online programming environment, methods for teaching programming, informatics.

оптимизирован для обеспечения высокой продуктивности программистов, читабельности кода и качества программного обеспечения [2, с. 22]. Преимущества использования языка программирования Python при изучении темы «Программирование и алгоритмизация» очевидны, поскольку простой синтаксис и низкий порог входа в данный язык позволяют обучающимся быстро освоить базовые алгоритмические структуры и акцентировать основное внимание на оттачивании навыков программирования [3, с. 21].

2. Актуальность рассмотрения вопроса о возможностях онлайн-сред программирования для обучения языку Python

В рамках преподавания программирования в школьном курсе информатики важным вопросом является возможность использования различных онлайн-сред программирования, которые могут применяться при обучении языку Python. При этом анализ содержания учебников и учебных пособий по предмету «Информатика» в контексте методики преподавания программирования на базе языка Python показывает, что для разработки программ авторы рекомендуют устанавливать на свой компьютер интерпретатор Python. При этом они не акцентируют внимание на возможностях онлайн-сред программирования и лишь приводят примеры таких сервисов.

В частности, Л. Л. Босова с соавторами в учебном издании «Начала программирования на языке Python» для VIII—IX классов для разработки программ на языке Python предлагает установить на компьютер интерпретатор языка и использовать встроенную в установщик Python интегрированную среду разработки IDLE, которая предназначена для ввода, просмотра, редактирования, запуска или отладки программы на языке Python [1, с. 4]. При этом в конце параграфа 1.1.3 «Режимы работы интерпретатора Python» авторы приводят примеры ресурсов для запуска и отладки программ на языке программирования Python, в том числе таких сервисов, как PythonTutor (<https://pythontutor.com/visualize.html#mode=edit>) и RexTester (http://rextester.com/1/python3_online_compiler). Данные онлайн-среды программирования поддерживают работу исключительно с версией языка Python 3.6, поддержка которой завершилась 23 декабря 2021 года [9].

К. Ю. Поляков, Е. А. Еремин в своих учебниках «Информатика (базовый и углубленный уровни)» для X—XI классов не акцентируют внимание на возможностях использования онлайн-сред программирования. Однако в рамках параграфа «Введение в язык Python» одной из первых тем, посвященных непосредственно программированию на Python, в разделе «Интересные сайты» также приводятся примеры различных онлайн-сред программирования для работы с версиями языка Python 2 и Python 3, в частности, pythonfiddle.com — онлайн-среда для программирования на языке Python 3.6 и codeskulptor.org — онлайн-среда программирования на Python 2 с использованием графического интерфейса [5, с. 155]. Следует отметить, что данные онлайн-среды поддерживают работу только с устаревшими версиями

языка Python, поскольку, согласно официальному сайту языка программирования Python, python.org, поддержка Python 2 была прекращена 1 января 2020 года [10].

Таким образом, учителя, желающие использовать онлайн-среды программирования в рамках обучения языку Python, не могут полностью полагаться на информацию, изложенную в учебниках и методических пособиях.

3. Характеристика возможностей онлайн-сред программирования для обучения языку Python

Онлайн-среды программирования — это специализированные веб-сервисы, которые предоставляют возможности для написания и запуска программ на соответствующем языке программирования. Основным преимуществом применения подобных сервисов является отсутствие необходимости устанавливать среду программирования на свой компьютер, что делает онлайн-среды программирования оптимальным вариантом при работе на компьютерах с низкой производительностью.

Далее мы более подробно рассмотрим возможности различных онлайн-сред программирования для обучения языку Python.

3.1. Python Shell

На первых занятиях по программированию важно познакомить обучающихся с возможностями интерактивного режима языка программирования Python, что позволяет реализовать онлайн-среда Python Shell:

<https://www.python.org/shell/>

Онлайн-среда поддерживает работу с интерактивным режимом языка программирования Python 3.9.5, используя интерактивную оболочку IPython, а также позволяет работать со всеми основными библиотеками языка Python, такими как `math` (библиотека для работы с математическими функциями), `random` (библиотека для работы со случайными числами), `itertools` (библиотека для работы с итераторами — структурами данных, которые облегчают обработку последовательности объектов в Python) и т. д. Однако данная онлайн-среда не поддерживает работу с графическими библиотеками, такими как `tkinter`, `turtle` и т. д., поскольку вывод результата выполнения команд языка Python осуществляется исключительно в консоли.

Демонстрация работы онлайн-среды программирования Python Shell представлена на рисунке 1. Для начала работы с сервисом необходимо нажать на красный указатель курсора в окне консоли, ввести соответствующую команду языка Python и нажать Enter. Если программа записана в соответствии с синтаксисом языка программирования Python, то на экран консоли будет выведен результат работы команды.

Онлайн-среда Python Shell может применяться учителем в рамках изучения первых тем раздела «Программирование» при знакомстве с интерактивным режимом языка программирования Python и для написания первых программ на Python с использованием интерактивного режима.

В частности, на уроке учитель может предложить обучающимся найти значение выражений с использо-

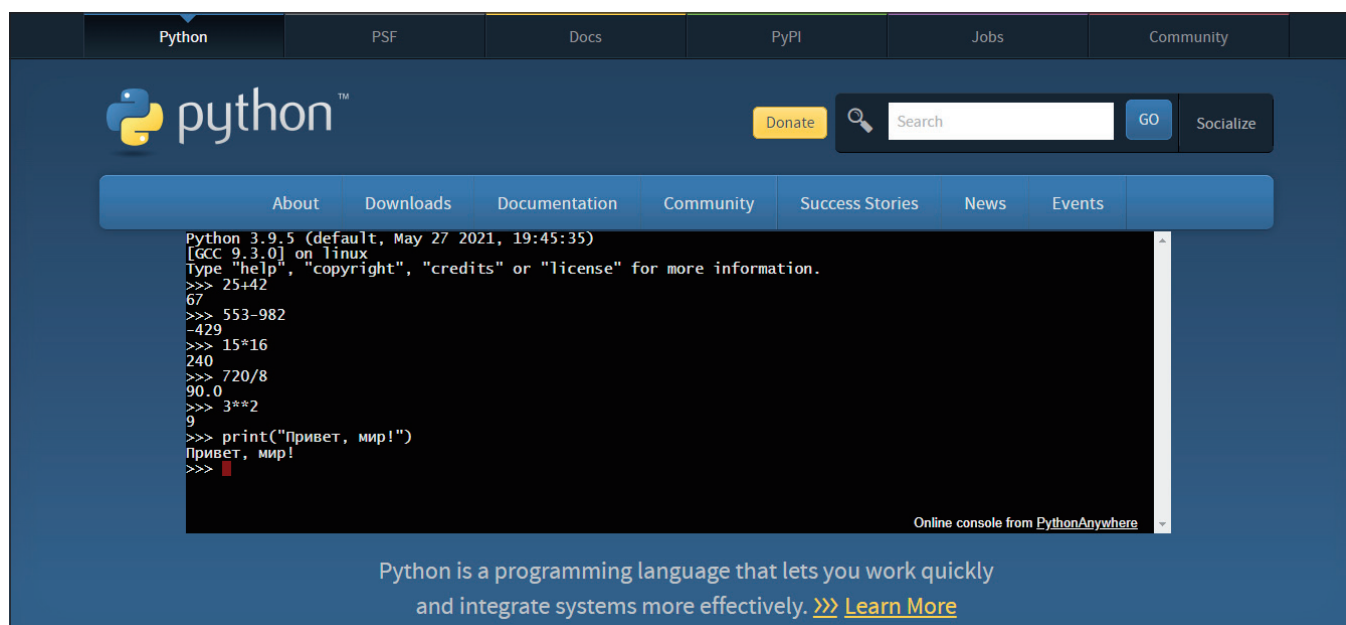


Рис. 1. Онлайн-среда Python Shell для работы с интерактивным режимом языка программирования Python

ванием математических операций (сложение «+», вычитание «-», умножение «*», деление «/» и возведение в степень «**»), а также написать небольшую программу, которая выводит на экран консоли текст «Привет, мир!»

3.2. Online Python

Важно отметить, что обучение языку Python в школьном курсе информатики в основном реализуется при работе с программным режимом среды программирования

Python. В данном режиме программа записывается в файл и выполняется при запуске. Одной из онлайн-сред, поддерживающих работу в программном режиме языка Python, является сервис Online Python:

<https://www.online-python.com/>

Online Python поддерживает работу в программном режиме среды программирования Python 3.8.5, а также дает возможность для работы со всеми стандартными библиотеками языка Python, которые не требуют созда-

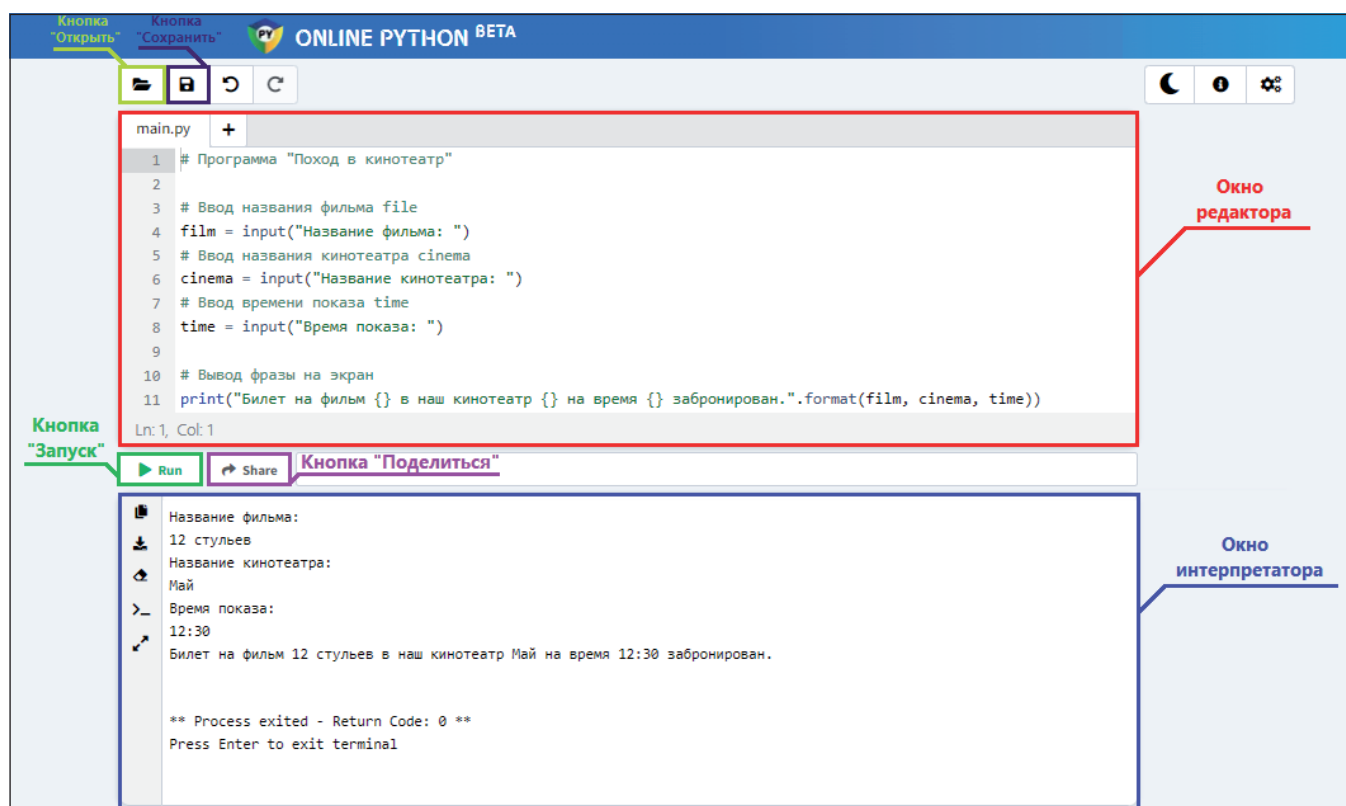


Рис. 2. Онлайн-среда Online Python для работы с программным режимом языка программирования Python

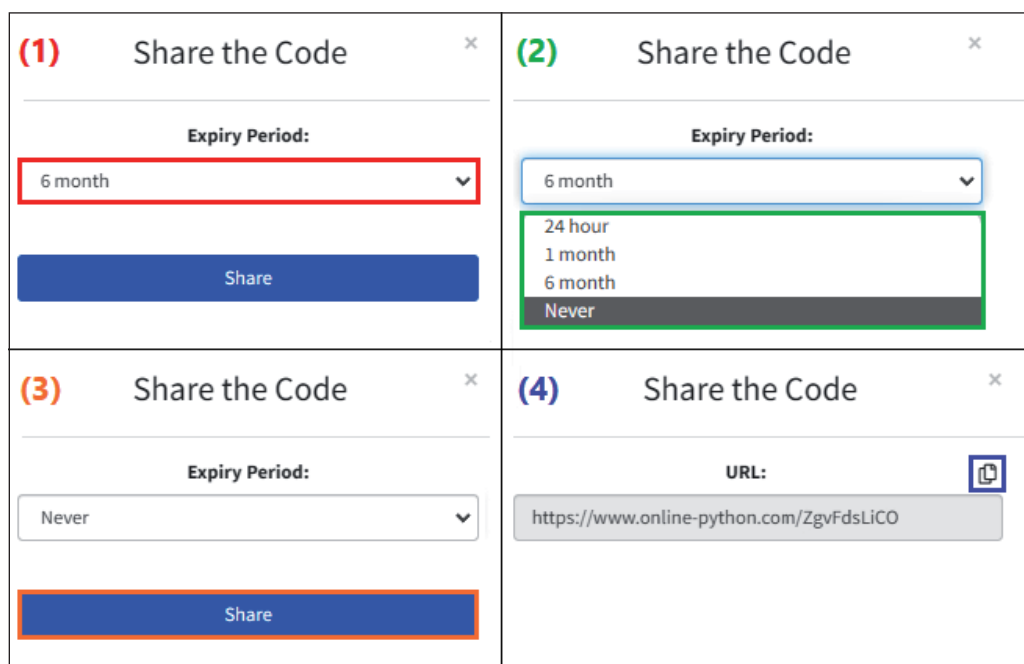


Рис. 3. Демонстрация работы функции создания ссылки на программу в онлайн-среде программирования Online Python

ния графического интерфейса (math, random, itertools, datetime и т. д.).

В качестве примера для демонстрации работы онлайн-среды программирования рассмотрим практическое задание «Поход в кинотеатр» к параграфу 54 «Введение в язык Python» в учебнике К. Ю. Полякова, Е. А. Еремина «Информатика (базовый и углубленный уровни)» для 10 класса [5, с. 144]. Условие задания «Поход в кинотеатр» звучит следующим образом:

«Напишите программу, которая получает на вход: название фильма, название кинотеатра, время показа, а затем выводит на экран фразу: “Билет на фильм (название фильма) в наш кинотеатр (название кинотеатра) на (время показа) забронирован“».

Демонстрация работы среды программирования Online Python представлена на рисунке 2. Программа на языке программирования Python записывается в окне редактора. Для того чтобы запустить программу, необходимо нажать на кнопку *Run (Запуск)*, в окне интерпретатора ввести с клавиатуры входные данные программы (название фильма, название кинотеатра, время показа), а затем нажать Enter и получить результат работы программы. Кроме того, обучающийся может сохранить свою программу и получить к ней доступ по ссылке при помощи кнопки *Share (Поделиться)*.

Важной особенностью данной среды является возможность загрузки и редактирования готовой программы, а также сохранения написанной программы на свой компьютер при помощи соответствующих кнопок — *Open (Открыть)* и *Save (Сохранить)*.

Среда Online Python может использоваться в качестве замены среды программирования IDLE Python в рамках изучения программирования как обучающимися при выполнении домашних заданий, так и учителем при проведении уроков по соответствующим темам раздела «Программирование». При этом отдельно необходимо отметить возможность онлайн-среды поделиться

написанной программой при помощи кнопки *Share*, что позволяет обучающемуся отправить учителю свою программу на языке Python посредством ссылки (рис. 3). При нажатии на кнопку *Share (Поделиться)* появляется всплывающее окно (1), в котором ученик может выбрать срок действия ссылки на программу (2): 24 часа, 1 месяц, 6 месяцев, постоянная ссылка. Затем необходимо нажать на кнопку *Share (Поделиться)*, чтобы подтвердить свой выбор (3) и получить ссылку, которую можно скопировать для отправки программы учителю (4). Написанная программа доступна по ссылке:

<https://www.online-python.com/ZgvFdsLiCO>

3.3. JDoodle Online Python 3 IDE

В качестве онлайн-среды для работы в программном режиме языка Python в рамках изучения программирования на уроках информатики также может использоваться сервис JDoodle Online Python 3 IDE:

<https://www.jdoodle.com/python3-programming-online/>

Следует отметить, что данная онлайн-среда приводится в качестве ресурса для запуска и отладки программ на Python в режиме online [1, с. 8].

Online Python 3 IDE позволяет работать в программном режиме языка Python 3.9.9, при этом так же, как и Online Python, поддерживает работу со стандартными библиотеками Python без поддержки графического интерфейса (math, random и т. д.).

Отличительной особенностью онлайн-среды является возможность написания программ с получением входных данных программы из загруженного файла и записи выходных данных программы в отдельный файл, который можно скачать на свой компьютер.

В качестве примера для демонстрации работы онлайн-среды JDoodle Online Python 3 IDE рассмотрим практическое задание «Поиск ученика» к параграфу 68 «Работа с файлами» в учебнике К. Ю. Полякова, Е. А. Еремина «Информатика (базовый и углубленный

уровни)» для 10 класса [5, с. 255]. Условие задания «Поиск ученика» звучит следующим образом:

«В текстовый файл school.txt записано 20 фамилий обучающихся 7А класса через пробел (Иванов Даниловский Вадимов Петров Сидоров Данков Семенов Чайкин Фадеев Жаворонков Евсеев Ожегов Дедов Царапкин Демидов Морозов Макаренко Закаатов Климов Калинин). Выведите в файл list.txt фамилии обучающихся через пробел, которые начинаются на букву Д длиной не более шести символов. Если по заданным условиям не было найдено фамилий обучающихся, запишите в файл list.txt слово «NO»».

Демонстрация работы онлайн-среды JDoodle Online Python 3 IDE представлена на рисунке 4. В окне редактора записана программа, которая получает из файла school.txt фамилии обучающихся 7А класса и записывает в файл list.txt фамилии обучающихся, которые начина-

ются на букву Д и имеют длину не более шести символов. Если фамилий по данным критериям не было найдено, программа записывает в файл list.txt слово «NO». Чтобы загрузить входной файл программы для работы с онлайн-средой, используется кнопка *Загрузить файл* со значком документа. Загруженный файл отображается в онлайн-среде в разделе загруженных файлов входных данных программы (*Files Uploaded*). После того как написана программа и файл входных данных загружен в систему, необходимо нажать кнопку *Execute* (*Выполнить*), чтобы запустить программу. В качестве результата работы программы в разделе сгенерированных файлов выходных данных (*Files Generated*) появляется файл list.txt со списком искомых фамилий обучающихся (которые начинаются на букву Д и имеют длину не более шести символов). Данный файл можно скачать на свой компьютер при нажатии на соответствующую ссылку. В свою

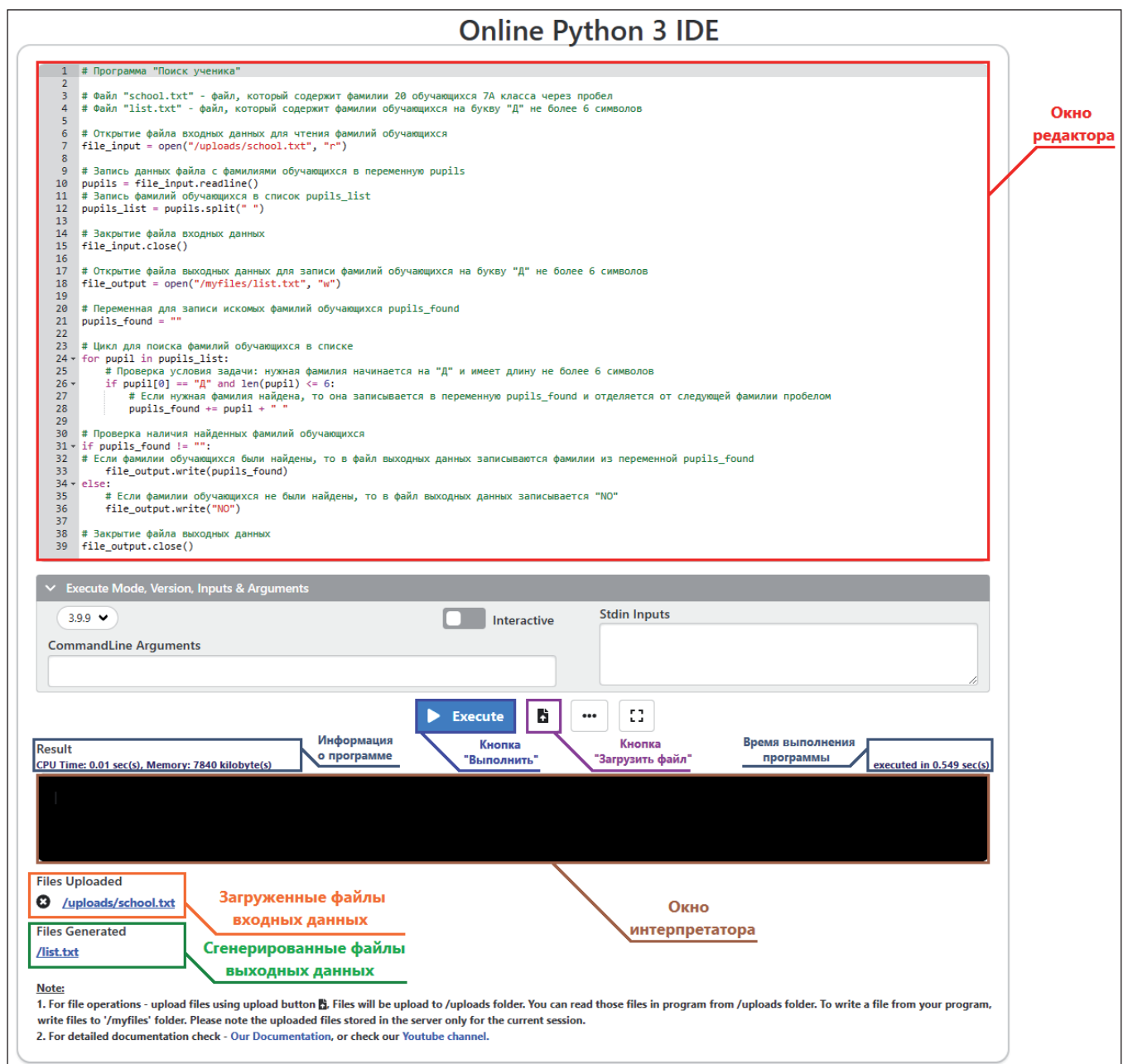


Рис. 4. Онлайн-среда JDoodle Online Python 3 IDE для работы с программным режимом языка программирования Python

очередь, над окном интерпретатора Python отображается информация о времени работы процессора, количестве памяти, затраченных на выполнение программы, а также о времени работы программы. При этом само окно интерпретатора является пустым, поскольку результат работы программы был записан в файл. Исходный код программы доступен по ссылке (на стороннем сервисе):

<https://pastebin.com/riaef3Bs>

Важно отметить, что при написании программы с использованием входных данных из файла необходимо записывать путь к файлу в следующем виде: /uploaded/school.txt, где school.txt — это файл входных данных программы. Если выходные данные также записываются в файл, в соответствующей команде необходимо использовать путь: /myfiles/list.txt, где list.txt — это файл выходных данных программы.

Таким образом, онлайн-среда JDoodle Online Python 3 IDE, с одной стороны, может использоваться в качестве замены среды программирования IDLE Python при проведении уроков по различным темам раздела «Программирование». Однако, с другой стороны, среда JDoodle Online Python 3 IDE также ориентирована на написание программ с использованием входных данных из файла и может применяться учителем при проведении уроков по теме «Работа с файлами» в рамках изучения языка Python.

3.4. Trinket Python

Рассмотренные ранее онлайн-среды для обучения программированию на языке Python предполагают ра-

боту с использованием консоли интерпретатора Python и не рассчитаны на поддержку графических библиотек языка программирования Python.

В свою очередь, онлайн-среда программирования Trinket Python на базе среды программирования Python 3.9.6 поддерживает как стандартные библиотеки Python, такие как math (работа с математическими функциями), random (работа со случайными числами), time (работа со временем), так и библиотеку turtle для создания графических объектов и рисунков. Данная онлайн-среда доступна по ссылке:

<https://trinket.io/python>

Turtle — это графическая библиотека языка программирования Python, в которой обучающийся пишет для виртуальной черепашки инструкции, управляющие ее перемещением по экрану, при этом черепашка носит с собой ручку, а обучающийся может проинструктировать черепашку воспользоваться этой ручкой для рисования линий по мере перемещения по экрану [7, с. 25]. При помощи библиотеки turtle можно создавать простые и короткие программы для создания красивых и сложных орнаментов. Визуализация данных при использовании графической библиотеки turtle для раннего обучения школьников программированию способствует обеспечению плавного перехода от блочного программирования к освоению языка Python [8, с. 10].

Для демонстрации работы онлайн-среды Trinket Python с использованием библиотеки turtle рассмотрим задачу «Цветная спираль» из учебника Брайсона Пэйна «Python для детей и родителей» [7, с. 38]. В данном за-

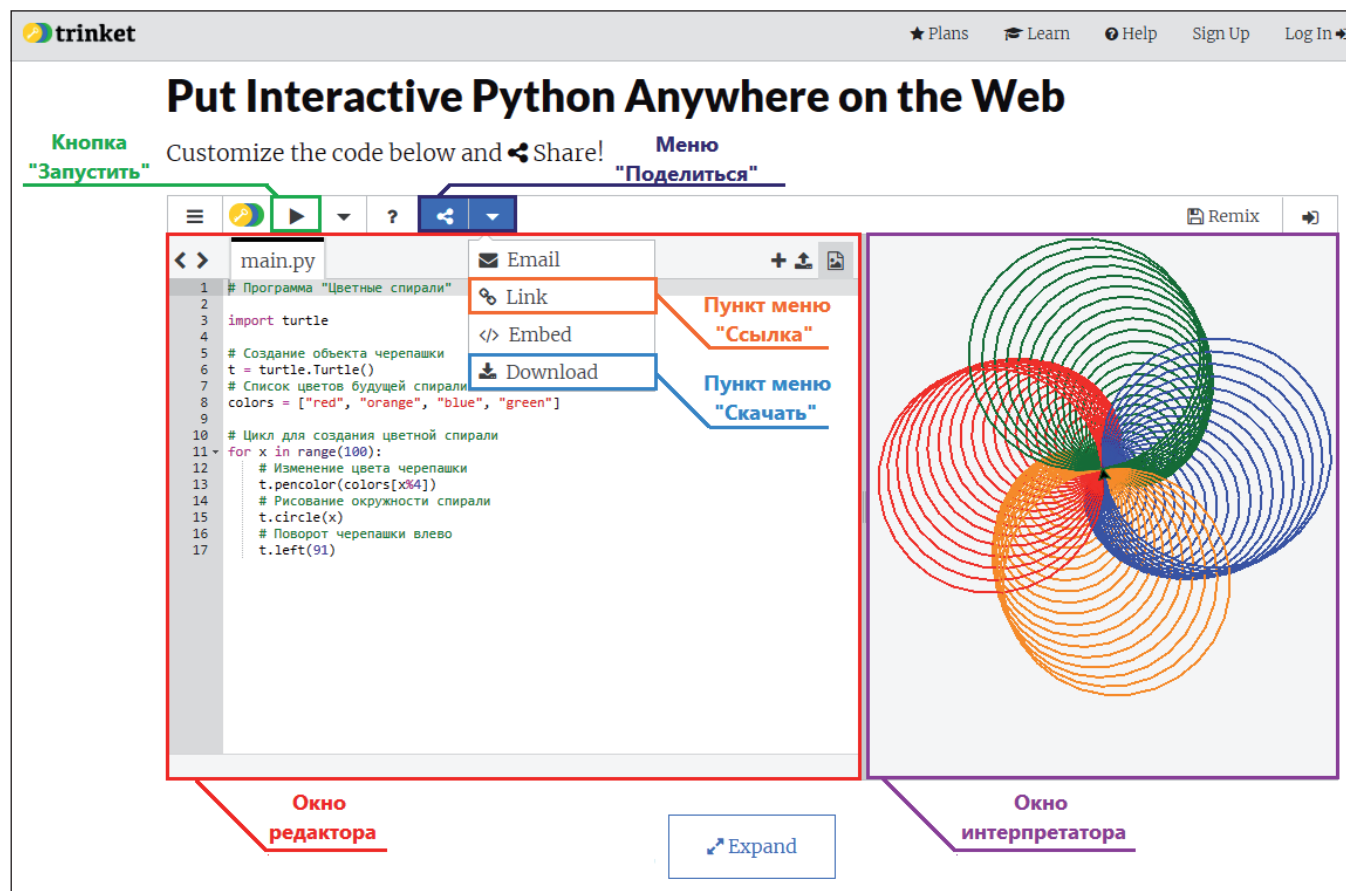


Рис. 5. Онлайн-среда Trinket Python с использованием графической библиотеки turtle

Таблица

Сравнительная характеристика рассмотренных онлайн-сред программирования

№ п/п	Название среды	Версия Python	Режим работы языка Python	Поддерживаемые библиотеки	Ссылка на сервис
1	Python Shell	3.9.5	Интерактивный	Все стандартные библиотеки Python без графического интерфейса	https://www.python.org/shell/
2	Online Python	3.8.5	Программный	Все стандартные библиотеки Python без графического интерфейса	https://www.online-python.com/
3	JDoodle Online Python 3 IDE	3.9.9	Программный	Все стандартные библиотеки Python без графического интерфейса	https://www.jdoodle.com/python3-programming-online/
4	Trinket Python	3.9.6	Программный	builtins, math, matplotlib.pyplot, numpy, operator, processing, pygal, random, re, string, time, turtle, urllib.request	https://trinket.io/python

дании необходимо написать программу для рисования четырехцветной спирали.

Демонстрация работы онлайн-среды программирования Trinket Python с использованием библиотеки turtle представлена на рисунке 5. Программа на Python записывается в окне редактора. Для того чтобы запустить написанную программу, необходимо нажать на кнопку *Запустить* с соответствующим значком. После нажатия на кнопку запуска программы в окне интерпретатора Python начнет появляться изображение спирали, создаваемое при помощи черепашки. Используя кнопку *Поделиться*, обучающийся может поделиться с учителем ссылкой на написанную программу, выбрав пункт меню *Link (Ссылка)*, или скачать программу на свой компьютер, выбрав пункт *Download (Скачать)*.

Программа доступна по ссылке:

<https://trinket.io/python/c99ce50202>

Онлайн-среда Trinket Python также может использоваться как учителем, так и обучающимися в рамках изучения программирования в школьном курсе информатики. В частности, применение данной среды будет наиболее эффективным при рассмотрении возможностей библиотеки Turtle в рамках изучения темы «Циклы» раздела «Алгоритмизация и программирование».

3.5. Сравнение онлайн-сред программирования

Представленные онлайн-среды программирования для работы в интерактивном и программном режимах языка программирования Python (Python Shell, Online Python, JDoodle Online Python 3 IDE, Trinket Python) дают возможность как учителю, так и обучающимся применять основной потенциал языка без установки на компьютер среды программирования IDLE Python.

В частности, онлайн-среды программирования предоставляют возможности:

- работать с новыми версиями языка Python — Python 3.8 и Python 3.9;
- использовать стандартные библиотеки Python, применяемые в рамках обучения языку программирования (math, random), в том числе библиотеку turtle для создания графических изображений;
- создавать программы с возможностью обработки входных данных из файла и записи выходных данных программы в файл;

- редактировать готовые программы;
- сохранять написанные программы на свой компьютер.

Сравнительная характеристика рассмотренных онлайн-сред программирования для языка Python представлена в таблице.

Рассказав о достоинствах онлайн-сред программирования, следует отметить и их недостатки. В частности, эти среды требуют стабильного подключения к сети Интернет во время запуска написанной программы. Кроме того, онлайн-среды программирования не предоставляют возможности для работы с самой последней версией языка Python — Python 3.10. Одним из наиболее важных нововведений языка программирования Python является появление новой алгоритмической конструкции — структурного сопоставления шаблонов, а также значительное улучшение отчетов об ошибках, в которых интерпретатор языка Python указывает не только место, в котором присутствует ошибка, но и возможные варианты ее исправления (на английском языке) [4, с. 96].

4. Заключение

Онлайн-среды программирования могут применяться в качестве электронной поддержки при проведении уроков информатики в рамках изучения языка программирования Python, в том числе в условиях смешанного и дистанционного обучения, при работе с интерактивным и программным режимом Python для написания программ с использованием стандартных библиотек языка (math, random, turtle и т. д.), а также программ, которые занимаются обработкой файлов (чтение входных данных из одного файла и запись результата работы программы в другой файл). При этом онлайн-среды не требуют установки программного обеспечения на компьютер и могут использоваться при наличии подключения к сети Интернет.

Список источников

1. Босова Л. Л., Аквилянов Н. А., Кочергин И. О., Штепа Ю. Л., Бурцева Т. А. Информатика. 8–9 классы. Начала программирования на языке Python. Дополнительные главы к учебникам. М.: БИНОМ. Лаборатория знаний, 2020. 96 с.
2. Лутц М. Изучаем Python / пер. с англ. Ю. Н. Артеменко. 5-е изд. СПб.: Диалектика, 2019. 832 с.

3. *Маркелов В. К.* Методика преподавания темы «Программирование и алгоритмизация» в средней школе на базе комплекта интерактивных заданий: Выпускная квалификационная работа. 2021. 104 с.

4. *Маркелов В. К., Завьялова О. А.* Возможности языка Python 3.10 при обучении программированию в школе // Шуйская сессия студентов, аспирантов, педагогов, молодых ученых. Материалы XIV Международной научной конференции (Москва—Иваново—Шуя, 6–7 октября 2021 г.). Иваново: Изд-во ШФ Ивановский государственный университет, 2021. С. 96–97.

5. *Поляков К. Ю., Еремин Е. А.* Информатика (базовый и углубленный уровни) (в 2 частях). 10 класс. Ч. 2: учебник. М.: БИНОМ. Лаборатория знаний, 2019. 352 с.

6. Примерная рабочая программа основного общего образования. Информатика. Базовый уровень (для

7–9 классов образовательных организаций) (Одобрена решением федерального учебно-методического объединения по общему образованию, протокол 3/21 от 27.09.2021 г.). <https://fgosreestr.ru/uploads/files/dcca994c21165f0d49d4baf4a7e008c0.pdf>

7. *Пэйн Б.* Python для детей и родителей / пер. с англ. М. А. Райтман. 2-е изд. М.: Эксмо, 2021. 352 с.

8. *Сорокина Т. Е.* Использование графической библиотеки Turtle Graphics языка Python для плавного перехода от блочного программирования к текстовому // Информатика в школе. 2018. № 3. С. 6–10.

9. PEP 494 -- Python 3.6 Release Schedule | Python.org // Python.org. <https://www.python.org/dev/peps/pep-0494>

10. Sunsetting Python 2 | Python.org // Python.org. <https://www.python.org/doc/sunset-python-2/>



DOI: 10.32517/2221-1993-2022-21-3-83-90

Н. К. Князева

Средняя школа № 137, г. Красноярск, Россия

ФОРМИРОВАНИЕ НАВЫКОВ КОММУНИКАЦИИ И КООПЕРАЦИИ У МЛАДШИХ ШКОЛЬНИКОВ ПРИ СОЗДАНИИ АНИМАЦИИ НА УРОКАХ МАТЕМАТИКИ

Аннотация

В статье рассматривается формирование у младшего школьника навыков коммуникации и кооперации как составляющих его функциональной грамотности. Раскрываются способы работы с анимацией (мультипликацией) на уроках математики при объединении учеников в малые группы. Рассматриваются формы организации учебной деятельности младших школьников; предлагается характеристика задач, наиболее подходящих для анимирования; приводится методика организации урока с примерами заданий, содержание которых позволяет организовать коммуникацию учащихся; даются примеры задач, решенных при помощи анимации, созданной детьми; приводятся методики для диагностики уровня коммуникации учеников. Представлен анализ результатов проводимой в течение нескольких лет работы по формированию навыков коммуникации и кооперации через создание анимации. Материал может быть полезен школьным учителям, студентам педагогических вузов и методистам.

Ключевые слова: начальная школа, анимация, мультипликация, коммуникация, кооперация, математика, логические задачи.

1. Введение

Требования современного общества к образованию — воспитание функционально грамотного человека, быстро и успешно адаптирующегося к изменяющимся условиям жизни. Важными составляющими функциональной грамотности являются навыки коммуникации и кооперации. Такой вывод можно сделать на основании изучения как мнения работодателей о наиболее востребованных навыках сотрудников компаний [9], так и обновленных федеральных государственных образовательных стандартов [14]. Актуальность формирования навыков коммуникации и кооперации подтверждают также международные исследования качества образования [6].

Коммуникация — вид деятельности, содержанием которой является обмен информацией между членами одного языкового сообщества для достижения взаимо-

понимания и взаимодействия [1]. Чтобы коммуникация была эффективной, ее участникам необходимо выражать и интерпретировать свои мысли, чувства и факты в устной и письменной формах. Под **кооперацией** понимается эффективное взаимодействие с другими людьми и эффективная работа в команде [12].

У младших школьников навыки коммуникации формируются в учебном сотрудничестве со взрослым, со сверстниками, с самим собой [18]. Самыми равноправными являются отношения со сверстниками, которые воспитывают у учащегося критичность в отношении мыслей, слов и поступков другого, а также учат видеть позицию другого, оценивать ее, а главное — иметь собственную точку зрения, отличную от точки зрения другого [15].

Цель данной статьи — раскрыть способы формирования навыков коммуникации и кооперации как составляющих функциональной грамотности.

Контактная информация

Князева Надежда Константиновна, учитель начальных классов, средняя школа № 137, г. Красноярск, Россия; адрес: 660012, Россия, г. Красноярск, ул. Судостроительная, д. 50; e-mail: nadusha8@yandex.ru

N. K. Knyazeva

School 137, Krasnoyarsk, Russia

FORMATION OF COMMUNICATION AND COOPERATION SKILLS IN JUNIOR SCHOOLCHILDREN WHILE CREATING ANIMATION AT MATH LESSONS

Abstract

The article discusses formation of communication and cooperation skills in a junior schoolchild as components of his functional literacy. The ways of working with animation at math lessons when students are combined into small groups are revealed. Forms of organization of educational activity of junior schoolchildren are considered; a description of the tasks most suitable for animation is proposed; the methodology for organizing a lesson is given with examples of tasks, the content of which allows organizing communication between students; examples of tasks solved with the help of animation created by children are given; methods for diagnosing the level of communication of students are described. An analysis of the results of several years of work on the formation of communication and cooperation skills through the creation of animation is presented. The material can be useful to school teachers, students of pedagogical universities and methodologists.

Keywords: primary school, animation, communication, cooperation, mathematics, logic problems.

Для решения задачи формирования навыков коммуникации и кооперации важно правильно выбрать форму организации сотрудничества детей, подобрать задачи, выбрать методику формирования соответствующих компетенций, подтвержденную экспериментом. В настоящей статье рассматриваются форма организации сотрудничества и задачи, которые обусловлены методикой формирования указанных навыков, основанной на использовании анимации.

2. Форма организации сотрудничества учащихся

Одним из способов организации сотрудничества ребенка с ребенком является объединение учеников в **малые группы** (от двух до пяти человек). Именно в малых группах у детей будет возможность участвовать в дискуссии внутри группы и создавать совместный продукт, в нашем случае — анимационный ролик [8].

Организацию групп целесообразно проводить в таком порядке: учитель ставит проблему перед всем классом, фиксирует с учащимися основные вопросы, а далее помогает ученикам разделиться на группы.

Деление на группы может происходить разными способами:

- Самый эффективный способ — самостоятельный выбор учениками партнеров в соответствии с предстоящей учебной задачей. Однако в таком случае дети, не имеющие теплых дружеских отношений в коллективе, могут остаться вне групп. Поэтому учителю надо продумать, как разделить учеников, чтобы каждый имел возможность высказаться и быть услышанным.
- Учащиеся могут собраться в группы, выбрав одно и то же задание. Однако при этом способе деления существует риск, что какая-то группа будет слишком многочисленной.
- Учитель может сам разделить учеников на группы, создавая равноценные по силам коллективы, включив в каждую группу и лидеров, и генераторов идей, и исполнителей.
- Можно использовать для создания групп расположение учеников в классе за партами, объединив учеников по два или по четыре человека.
- Парная работа — также эффективная форма для активного взаимодействия учащихся (фото 1).

Дети, испытывающие трудности с контролем своей эмоционально-волевой сферы, **могут отказаться работать с партнером**. Так как формирование групп происходит для работы над созданием мультфильма, учителю не стоит принуждать таких детей присоединиться к группе. Партнером для них станет воображаемый зритель, для которого и будет создаваться мультфильм. Мультипликация (анимация) позволяет поставить такого ребенка в позицию собеседника. Собеседника, ведущего диалог с невидимым пока зрителем, будущим потребителем анимационного продукта. Поэтому такой ребенок вынужден предполагать запросы зрителя, бороться за его интерес, даже работая в одиночку. Это важно понимать учителю, который *должен позволить любому ученику самому выбрать формат работы: групповая работа или индивидуальная*. Позже ученик-одиночка сможет



Фото 1. Парная работа

разделить работу с учителем, с другим мультипликатором. Постепенно он сам придет к необходимости включиться в группу, где существует распределение ролей и процесс анимации происходит быстрее, где мнение одноклассников поможет в решении задачи.

Нам представляется, что **групповая форма является наиболее эффективной для организации дискуссии при создании мультипликации**.

При традиционной фронтальной форме учитель может столкнуться с разными точками зрения, например, при решении задачи про улитку:

Улитка ползет по вертикальной стене высотой 5 метров. Улитка днем проползла вверх 3 метра, а за ночь съезжала вниз на 2 метра. Сколько дней надо улитке, чтобы добраться до верха стены (фото 2)? [5]

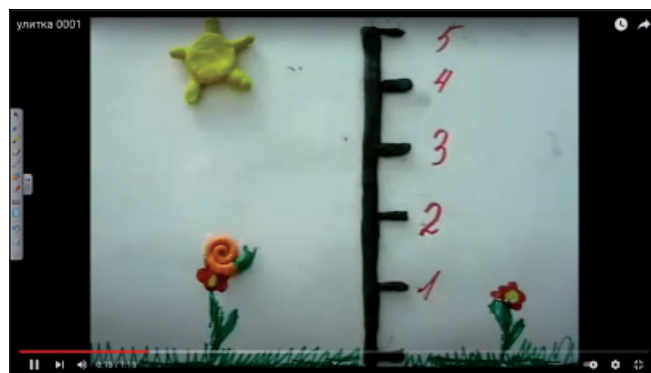


Фото 2. Кадр из мультипликационного ролика к задаче про улитку

Опыт показывает, что при решении подобных задач возникает несколько вариантов. Однако как раз обнаружение разных точек зрения является слабым звеном в общеклассной дискуссии младших школьников. Ребенку в этом случае яснее других точек зрения видится точка зрения учителя и неотчетливо — точки зрения других детей, которых он может не услышать.

Поэтому, на наш взгляд, **формировать у учащихся умения высказывать свою точку зрения, слышать мнение другого, противопоставлять свое мнение**

мнению другого эффективнее в малых группах с организацией микродискуссии внутри группы.

3. Характеристика математических задач для анимирования

Рассмотрим, какие задания полезно использовать для организации взаимодействия внутри малых групп. Это должны быть задачи, не имеющие однозначного решения, лежащего на поверхности. Задачи, которые дадут пищу для внутригрупповой дискуссии. В то же время такие задачи должны наталкивать на стройный ход рассуждения при решении, позволяющий убедить собеседника в выборе ответа. Математическое содержание позволяет организовать коммуникацию учеников, основанную на совместном создании мультипликации при решении задач.

Для работы по созданию мультипликации мы использовали нестандартные логические задачи, имеющие определенное, единственно верное решение и условие, которое можно иллюстрировать. Изучив возможности разных групп логических задач, мы пришли к мнению, что *удобнее и нагляднее анимировать задачи, речь в которых идет о процессе (движения, работы, составления целого из частей) или механическом действии (задачи на движение, переливание, взвешивание, переправы, работу и др.).*

Нестандартные логические задачи целесообразно вводить в соответствии с планированием основного содержания курса математики начальной школы.

Например, при изучении темы «Числовая прямая» уместны задачи про счет с двух сторон:

Маша стоит в строю на уроке физкультуры пятая, если считать слева, и пятая, если считать справа. Сколько детей в строю?

а также про распилы и разрезы:

Бревно распилили на пять частей. Сколько распилов сделали (фото 3)?



Фото 3. Кадр из мультипликационного ролика к задаче про распилы

При изучении темы «Задачи на движение» целесообразно организовать решение задач про движение объекта, обладающего длиной, например:

Поезд длиной 1 км едет со скоростью 60 км в час. Впереди лес протяженностью в 1 км. Сколько времени понадобится поезду, чтобы проехать лес (рис. 1)?

В данной статье мы не ставим целью рассмотреть методику работы с нестандартными логическими задачами и типологию логических задач. Цель статьи — рас-

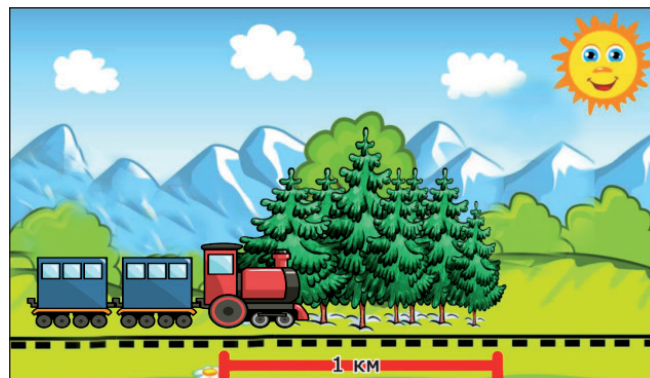


Рис. 1. Задача про движение поезда (объекта, обладающего длиной)

крыть возможности мультипликации для формирования у младших школьников навыков коммуникации при решении нестандартных логических задач. Сам же прием создания детьми мультипликации является одним из способов решения нестандартных задач и демонстрации ответа для других учеников.

Задачи для создания анимационной модели могут быть как предложены учителем, так и подобраны самими учениками. Исследования в данном направлении говорят о том, что если решение задачи вызывает затруднение учеников при обсуждении внутри группы, то в процессе совместного анимирования условия задачи учащиеся найдут правильное решение.

4. Методика организации урока

Для того чтобы групповая работа стала эффективной формой для совместного поиска, сотрудничества и плодотворной деятельности учеников, должны быть соблюдены некоторые условия:

- Задача должна быть подобрана так, чтобы индивидуальное ее решение было трудоемким и занимало много времени, но совместное обсуждение значительно ускорило поиск способа решения и требовало распределения операций между членами группы.
- Продолжительность групповой работы должна быть небольшой (от 7 до 20 мин) [15].
- Результат совместной работы должен быть зафиксирован. В нашем случае это фиксация в виде анимационной модели способа решения математической задачи.

С чего следует начинать групповую работу? Организация учебного сотрудничества происходит с первых дней ребенка в школе. Понятные для взрослых правила работы в команде и нормы поведения в обществе надо прожить, оценить их эффективность (или неэффективность) и только потом принять. При этом сценарии различных отношений в группе (как и отношений с учителем) проигрываются на практике.

Приведем пример. Правило поднятой руки вводится именно с нарушения этого правила на первом уроке: «Давайте познакомимся. Скажите, ребята, как вас зовут? Назовите свое имя на раз, два, три...» В общем шуме детям становится очевидно, что, если говорить всем сразу, имена друг друга не услышишь и не разберешь. Прихо-

дится вместе с детьми придумывать правило, регулирующее эту ситуацию. Этим правилом и становится правило поднятой руки. Но об этом детям сообщили не учитель, не предписание школьного устава, не администрация школы. Этот вывод дети сделали сами как единственно возможный способ конструктивного общения на уроке. Чтобы его зафиксировать, полезно сделать графическую модель *своего* правила (рис. 2).

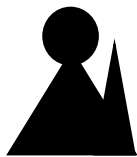


Рис. 2. Графическая модель правила поднятой руки

Кажущиеся мелочами детские действия при работе в команде (обращение к соседу, взгляд, удобная посадка, уровень громкость речи, жесты и др.) все проходят «апробацию» в детском коллективе, все обсуждается для того, чтобы превратить их со временем в нормы учебного сотрудничества.

Г. А. Цукерман [15] рекомендует учителю различать такие на первый взгляд схожие действия, как спор и ссора, и непосредственно проигрывать ситуацию отношений ссоры и спора. Ведь для того чтобы спор стал источником истины, необходимо, чтобы дети отстаивали свое мнение аргументированно, пытаясь убедить своего оппонента. Для этого предлагается сначала «поссориться» от лица сказочных героев: «Так правильно! Нет, неправильно! Нет, правильно! Нет, неправильно!», чтобы увидеть тупиковый итог диалога, а потом «поспорить», чтобы определить эффективный способ решения проблемы: «Я думаю, что так правильно, потому что...», «А я думал, что это из-за того, что...», «Ты прав, спасибо тебе».

Особенно удобно проводить такое сравнение на уроках математики при решении нестандартных логических задач, ведь именно тогда понадобятся доказательства и логические рассуждения. Первоклассникам пока трудно отделить свои личные отношения от учебных содержательных отношений, на основе которых и строится учебное сотрудничество. Поэтому все споры должны быть организованы на предметном содержании, иногда противоречивом — для возникновения разных точек зрения.

Рассмотрим пример. При изучении темы «Площадь» на уроке математики каждой паре учеников дается несколько геометрических фигур разной формы, но равных по площади, за исключением одной фигуры (рис. 3). Чтобы показать, что у всех фигур, кроме одной, площади одинаковые, и убедить своего соседа в правоте, надо привести доказательства: например, приложить мерку и узнать, сколько раз она содержится в площади всех фигур, или сосчитать количество клеток, или разрезать и перекроить так, чтобы фигуры стали одинаковыми

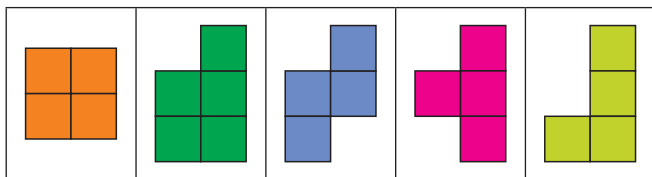


Рис. 3. Задание: сравнить площади фигур

по форме, и проверить наложением фигур друг на друга соотношение их площадей.

Если, работая в паре, дети обращаются непосредственно к своему соседу, то при работе в четверке ученику приходится прислушиваться сразу к трем однокурсникам.

Упражнение, которое учит договориться без слов о совместном действии, — игра «Однорукие». «Одноруким» надо сложить рисунок из геометрических фигур. В группе четыре участника, на группу выдается банк — четыре фигуры (рис. 4). Каждый ребенок использует только одну руку (вторую руку прячет за спину). На счет «раз» каждый берет из общего банка одну фигуру, на счет «два» — кладет фигуру в общий рисунок. Может возникнуть ситуация, что на счет «раз» сразу два ребенка попытаются взять одну фигуру из банка. Чтобы так не получилось, детям приходится следить друг за другом внимательно и договариваться при помощи жестов (показывать предварительно пальцем, какую фигуру планируют взять). Если на одну фигуру нацелились сразу два пальца, надо поменять решение и взять другую фигуру. Такое упражнение учит быть внимательным к другим членам группы и действовать сообща для достижения общей цели.

Фигуры для рисунка	Рисунок «Дом с деревом»
Фигуры для рисунка	Рисунок «Елочка»

Рис. 4. Наборы геометрических фигур для игры «Однорукие»

Учителю полезно показывать позитивные образцы взаимодействия (так же, как и негативные) на примере сказочных героев, игрушек, разыгранных по ролям ситуаций. При этом негативные образцы взаимодействия ни в коем случае не должны быть связаны с конкретными детьми из класса [4].

Рассмотрим, как происходит организация коммуникации при решении нестандартной логической задачи. Ученики знакомятся с текстом задачи, прочитывая его вслух:

Во дворе школы ребята устроили каток и поставили по углам больших снеговиков. Однако каток оказался слишком мал. Как увеличить его площадь в два раза, но при этом снеговиков оставить за бортиком и каток оставить квадратным?

Это задача по типу известной задачи про пруд:

Имеется квадратный пруд. По углам его у воды растут четыре старых дуба. Пруд понадобилось расширить, сделав вдвое больше по площади, сохраняя квадратную форму. Но старых дубов трогать не желают. Можно ли расширить пруд до требуемых размеров так, чтобы все четыре дуба остались на своих местах у берегов пруда (рис. 5)? [11]

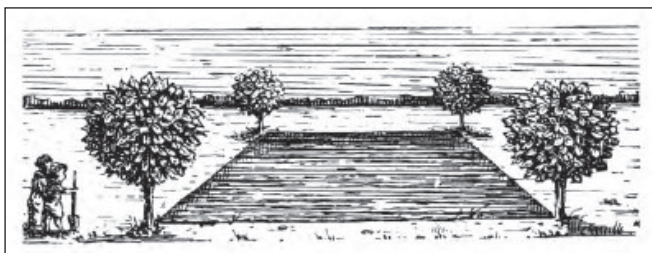


Рис. 5. Задача про пруд квадратной формы

Учитель организует дискуссию между всеми учениками класса, вызывает нескольких учеников нарисовать решение на доске. Каждый ученик объясняет свое решение, остальные слушатели показывают знаками согласия и несогласия (знаками «+» и «-», принятыми для обозначения оценки происходящего на уроке) свое отношение к предложенному решению (рис. 6).

Знаки согласия и несогласия также были приняты детьми при первых пробах совместной деятельности. Использование знаков согласия и несогласия «+» и «-» помогает, во-первых, формировать оценку и взаимную оценку учащихся, во-вторых, организовать дискуссию даже в первом классе. Главное — задать нормы вежливого обращения друг к другу: «Прошу вашего отношения, ребята (обращение к классу). Почему ты не согласен, Кирилл?» И далее, в зависимости от того, какие аргументы приведет Кирилл, ученик согласится: «Спасибо за замечание, ты прав» или возразит: «Спасибо за отношение, но посмотри внимательно, здесь вот какой знак, а это значит...»

Условие задачи	Решение задачи

Рис. 6. Решение задачи про квадратный каток

Учитель предлагает ученикам показать свое решение в виде анимации. Для этого создаются группы по четыре человека самым простым способом — ученики, сидящие за первой партой, разворачиваются к ученикам, сидящим за второй партой, — образуется четверка. Ребята договариваются, как распределить обязанности при съемке мультфильма. Выясняют, каких героев надо слепить из пластилина (четыре снеговика), какой нужен фон (квадратный каток), готовят бортики катка квадратной формы. При создании мультфильмов необходимы некоторые технические навыки педагога и учеников [2].

При показе своего ролика члены группы обязательно комментируют остальным детям решение задачи. Слушатели показывают свое отношение к работе группы, задают вопросы.

На этом этапе урока необходимо организовать рефлексию групповой работы. Важно выяснить, что особенно удалось группе, почему и на каких этапах воз-

никли трудности, как группа с ними справилась. Благодаря рефлексии ученики отметят и присвоят наиболее удачные способы работы в сотрудничестве с другими детьми [17].

5. Оценка результатов

Чтобы оценить эффективность описываемой методики создания анимации в группе, в течение четырех лет с 2016-го по 2020 год были проведены исследования в первых-вторых классах с помощью диагностики коммуникативных умений (методики «Дорога к дому», «Рукавички», «Мотивы выбора друга» и др.) [16, с. 100] и в третьих-четвертых классах с помощью итоговых диагностических работ.

Стартовая диагностика умений первоклассников показывает, что у большинства из них есть проблемы с коммуникацией. По результатам исследования краевого государственного Красноярского центра оценки качества образования (ЦОКО) [3] такие умения, как усвоение норм общения и взаимодействие со сверстниками, сформированы к началу обучения в первом классе на низком уровне.

В таблице 1 приведены результаты диагностики коммуникативных умений ЦОКО, проведенной в начале учебного года у первоклассников трех параллельных классов, участвующих в эксперименте. Из таблицы видно, что уровень взаимодействия со сверстниками во всех трех экспериментальных классах на начало первого класса в 2016 году ниже, чем этот же уровень по региону.

Таблица 1

Результаты диагностики ЦОКО умений первоклассников, 2016 год

Сравнение уровня взаимодействия со сверстниками по классу и по региону		
Экспериментальный класс 1	Экспериментальный класс 2	Экспериментальный класс 3
—○— Класс —●— Регион	—○— Класс —●— Регион	—○— Класс —●— Регион

Чтобы проследить динамику в формировании навыков коммуникации учащихся, на третьем году обучения были проведены оценочные процедуры независимым центром педагогических измерений [13]. Представленные в таблице данные показывают сравнение результатов

Таблица 2

Сформированность универсальных учебных действий по участникам эксперимента (параллель третьих классов) в 2019 году, в %

УУД	Низкий (не сформированы)	Базовый (достаточный)	Высокий (творческий)
Личностные	0,0	63,9	36,1
Регулятивные	0,0	49,1	50,9
Познавательные	0,0	71,7	28,3
Коммуникативные	0,0	50,1	49,9

Таблица 3

Сформированность универсальных учебных действий в среднем по Российской Федерации (параллель третьих классов) в 2019 году, в %*

УУД	Низкий (не сформированы)	Базовый (достаточный)	Высокий (творческий)
Личностные	1,5	62,7	35,9
Регулятивные	2,0	54,2	43,9
Познавательные	1,0	69,6	29,3
Коммуникативные	2,0	54,2	43,9

* Даны округленные значения, поэтому суммы не равны 100 %.

по участникам эксперимента (табл. 2) и в среднем по Российской Федерации (табл. 3). В отношении личностных результатов выше процент участников эксперимента с базовым и высоким уровнями при отсутствии учеников с низким уровнем личностных умений. Регулятивные умения, т. е. умения по учебной самоорганизации сформированы у этих третьеклассников на более высоком уровне по сравнению со средним по стране при отсутствии низких показателей. Познавательные умения отмечены на базовом уровне выше, чем по стране, при отсутствии низких показателей. Коммуникативные умения участники эксперимента продемонстрировали на высоком (творческом) уровне, это выше относительно уровня по Российской Федерации.

Диагностическая процедура «Групповой проект», проводимая ЦОКО в конце обучения в начальной школе, показала высокий уровень коммуникативных УУД учащихся, обучающихся с использованием создания анимации [10]. Умения, которые оценивались экспертом при работе группы, представлены в таблице 4. Ученики четвертого класса, обучающиеся с организацией создания анимации, продемонстрировали более высокие показатели сформированности коммуникативных умений по сравнению со средними показателями по региону. Важно отметить, что ни один из испытуемых не выбрал для оценки собственного вклада в работу группы пограничные значения показателя «Я сделал(а) очень много, без меня работа бы не получилась» и «Почти все сделали без меня», что говорит об участии в решении проекта всех членов малых групп.

Для получения качественного результата при формировании навыков коммуникации важно также самим

детям оценить эффективность своей групповой работы. Самооценка как система оценок и представлений о взаимодействии с другими людьми формирует поведение и деятельность ребенка в соответствии с социальными нормами [7].

Существуют способы самооценки навыков коммуникации при создании детьми мультипликации. Ниже в качестве примера перечислены критерии, выбранные самими детьми в качестве показателей для оценивания эффективности групповой работы:

- «Обсуждали вежливо»;
- «Распределяли работу»;
- «Говорили по очереди»;
- «Старались понять мнение других»;
- «Убедились, что все согласны с общим мнением»;
- «Мультфильм делали сообща»;
- «Все участвовали в презентации мультфильма».

Такие критерии помогают формировать действие контроля и самоконтроля учащихся при работе в команде, осознанного отношения к своей деятельности при групповом взаимодействии, а также напрямую влияют на повышение уровня коммуникации и кооперации младших школьников.

6. Заключение

Проделанная работа по изучению способов формирования навыков коммуникации и кооперации младших школьников при помощи детской самодельной анимации говорит о важности и результативности данной методики для воспитания функционально грамотного, успешного выпускника начальной школы.

Таблица 4

Результаты диагностической процедуры «Групповой проект», 2020 год*

Результаты выполнения группового проекта по участникам эксперимента, по отдельным действиям (IV класс)			
Элементы деятельности обучающихся	Успешность сформированности действий (в %)		
	Участники эксперимента	Конструкторский проект, в среднем по региону	Исследовательский проект, в среднем по региону
Коммуникативные действия			
Участие в презентации:			
• в презентации не участвовал	0,00	4,73	4,98
• участие в презентации незначительное	29,40	25,31	26,21
• участие в презентации значительное	70,60	69,91	68,80
Активность/инициативность ученика:			
• не проявляет инициативу	0,00	3,58	4,53
• активен, но инициативу не проявляет	37,48	32,34	31,55
• активен, проявляет инициативу	62,52	63,94	63,92
Ориентация на партнера и согласованность позиций:			
• не слушает, перебивает, не учитывает мнение партнера	0,00	0,91	0,87
• прислушивается к партнеру, старается учесть его позицию, если считает верной	21,20	20,18	24,50
• внимательно выслушивает партнера, с уважением относится к его позиции, старается ее учесть, если считает верной	78,80	78,87	74,63
Лидерство:			
• стремления к лидерству не проявляет, довольствуется ролью ведомого	11,00	28,75	28,57
• проявляет стремление к лидерству, в команде работать не умеет	5,30	10,11	10,19
• проявляет стремление к лидерству, умеет работать в команде на вторых ролях	83,70	61,06	61,17
Оценка личного вклада			
Самооценка — оценка личного вклада			
0 баллов — «Почти все сделали без меня»	0,00	0,14	0,36
1 балл	0,00	0,66	0,74
2 балл	0,00	0,83	1,49
3 балл	0,00	1,82	2,88
4 балл	3,70	5,21	7,67
5 балл	26,09	39,01	39,97
6 балл	34,78	15,12	15,28
7 балл	9,35	10,88	10,49
8 балл	4,35	10,27	9,45
9 балл	21,74	8,16	6,50
10 баллов — «Я сделал(а) очень много, без меня работа не получилась бы»	0,00	7,49	5,18

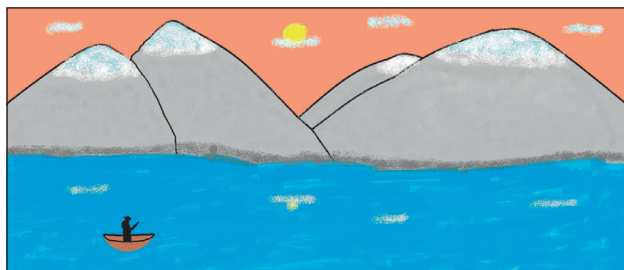
* Даны округленные значения, поэтому суммы не везде равны 100 %.

Список источников

1. Азимов Э. Г., Шукин А. Н. Новый словарь методических терминов и понятий (теория и практика обучения языкам). М.: Икар, 2009. 448 с.
2. Анофриков П. И. Принципы работы детской студии мультипликации. Учебное пособие для тех, кто хочет создать детскую студию мультипликации. 2-е изд., испр. и доп. Новосибирск, 2011. 43 с.
3. Даниленко О. В., Ермакова И. В., Ковалева Г. С., Нурминская Н. В., Серков М. В. Рекомендации по интерпретации результатов изучения готовности первоклассников к обучению в школе. М.: ИСМО РАО, 2010. 52 с.
4. Дусаевичкий А. К., Шилкунова З. И., Толмачева И. Н., Кондратьев Е. М. Урок в начальной школе: реализация системно-деятельностного подхода к обучению: книга для учителя. 5-е изд. М.: Вита-Пресс, 2012. 288 с.
5. Задача про улитку // Территория Надежды. <http://knyazeva.mmc.rightside.ru/home/the-news/243-2014-05-16-16-25-15>
6. Исследования // Центр оценки качества образования ФГБНУ «Институт стратегии развития образования Российской академии образования». <http://www.centeroko.ru/projects.html>
7. Как проектировать универсальные учебные действия в начальной школе: от действия к мысли: пособие для учителя / под ред. А. Г. Асмолова. М.: Просвещение, 2008. 151 с.
8. Князева Н. К., Носков М. В. Формирование математической грамотности в начальной школе с использованием математических моделей и их анимационной иллюстрации // Информатика в школе. 2021. № 8. С. 53–62. DOI: 10.32517/2221-1993-2021-20-8-53-62
9. Креативность, критическое мышление, кооперация и коммуникация — что такое 4К навыки и зачем их развивать, рассказали в Центре «Моя работа» // Моя работа. Центр занятости населения города Москвы. <https://czn.mos.ru/News/Detail/0e5edd27-5268-4dbd-a52a-bbd5cc291e25>
10. Описание краевой диагностической работы «Групповой проект» для 4 класса (КДР4 ГП) // Красноярский ЦОКО. <https://coko24.ru/wp-content/uploads/2021/06/Описание-КДР4-Групповой-проект.pdf>
11. Перельман Я. Головоломки. Вып. 2. М.: АСТ; Астрель, 2008. 200 с.
12. Пинская М. А., Михайлова А. М. Компетенции «4К»: формирование и оценка на уроке: практические рекомендации. М.: Российский учебник, 2019. 76 с.
13. Сводный аналитический отчет участия общеобразовательной организации МАОУ СШ № 137 (Код — 24401137), г. Красноярск, Красноярский край в полиатлон-мониторинге «Политоринг-2020». https://knyazeva.mmc.rightside.ru/documents/doc_download/238----
14. Федеральный государственный образовательный стандарт начального общего образования. <https://www.garant.ru/products/ipo/prime/doc/400807193/>
15. Цукерман Г. А. Виды общения в обучении. Томск: Пеленг, 1993. 270 с.
16. Шаталова О. А. Диагностика уровня сформированности универсальных учебных действий учащихся начальной школы: методическое пособие / под ред. С. А. Болотовой. М.: Русское слово — учебник, 2014. 168 с.
17. Щедровицкий Г. П. Рефлексия в деятельности. Доклад на семинаре ММК. 1972 // Вопросы методологии. 1994. № 3-4. С. 76–121. <https://www.elibrary.ru/item.asp?id=39140476>
18. Эльконин Д. Б. Психология обучения младшего школьника. М.: Знание, 1974. 63 с.

КОНКУРСЫ

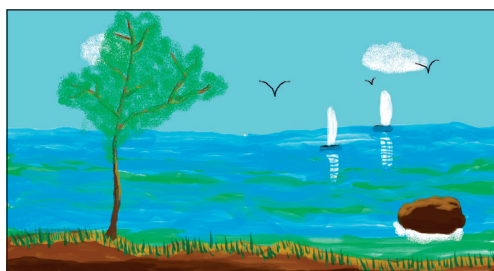
В оформлении обложки данного номера журнала «Информатика в школе» использованы работы следующих авторов:



Вадим Максимов,
студент Мамадышского политехнического колледжа,
г. Мамадыш, Республика Татарстан, Россия



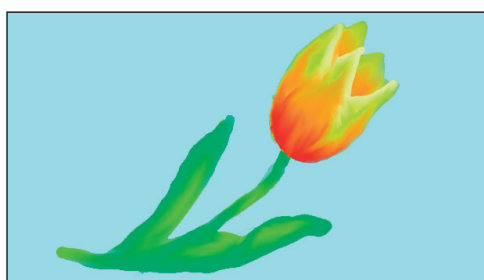
Анастасия Швецова,
учащаяся Центра технического творчества,
ученица Вятской гуманитарной гимназии с углубленным
изучением английского языка, г. Киров, Россия



Анастасия Булатова,
ученица гимназии № 4 им. А. С. Пушкина,
г. Йошкар-Ола, Республика Марий Эл, Россия



Степан Булатов,
ученик гимназии № 4 им. А. С. Пушкина,
г. Йошкар-Ола, Республика Марий Эл, Россия



Анна Замесина, Татьяна Немиро,
ученицы средней общеобразовательной школы № 26 г. Сочи
имени Героя Советского Союза
Диброва Кирилла Селиверстовича, Россия



Алина Легоцкая, Арман Авагян,
ученики средней общеобразовательной школы № 26 г. Сочи
имени Героя Советского Союза
Диброва Кирилла Селиверстовича, Россия



М. Е. Крекин,
учитель информатики школы № 1228 «Лефортово», г. Москва, Россия



В. А. Гаевская,
учитель информатики средней школы № 206, г. Минск, Беларусь

ПОДПИСКА

Журнал «Информатика в школе»

Индекс подписки
на 2-е полугодие 2022 года
(«Урал-Пресс», «АРЗИ» и другие агентства подписки)
81407

Периодичность выхода: 3 номера в полугодие (август, октябрь, декабрь)
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

КУРСЫ ПРОГРАММИРОВАНИЯ ДЛЯ ШКОЛЬНИКОВ

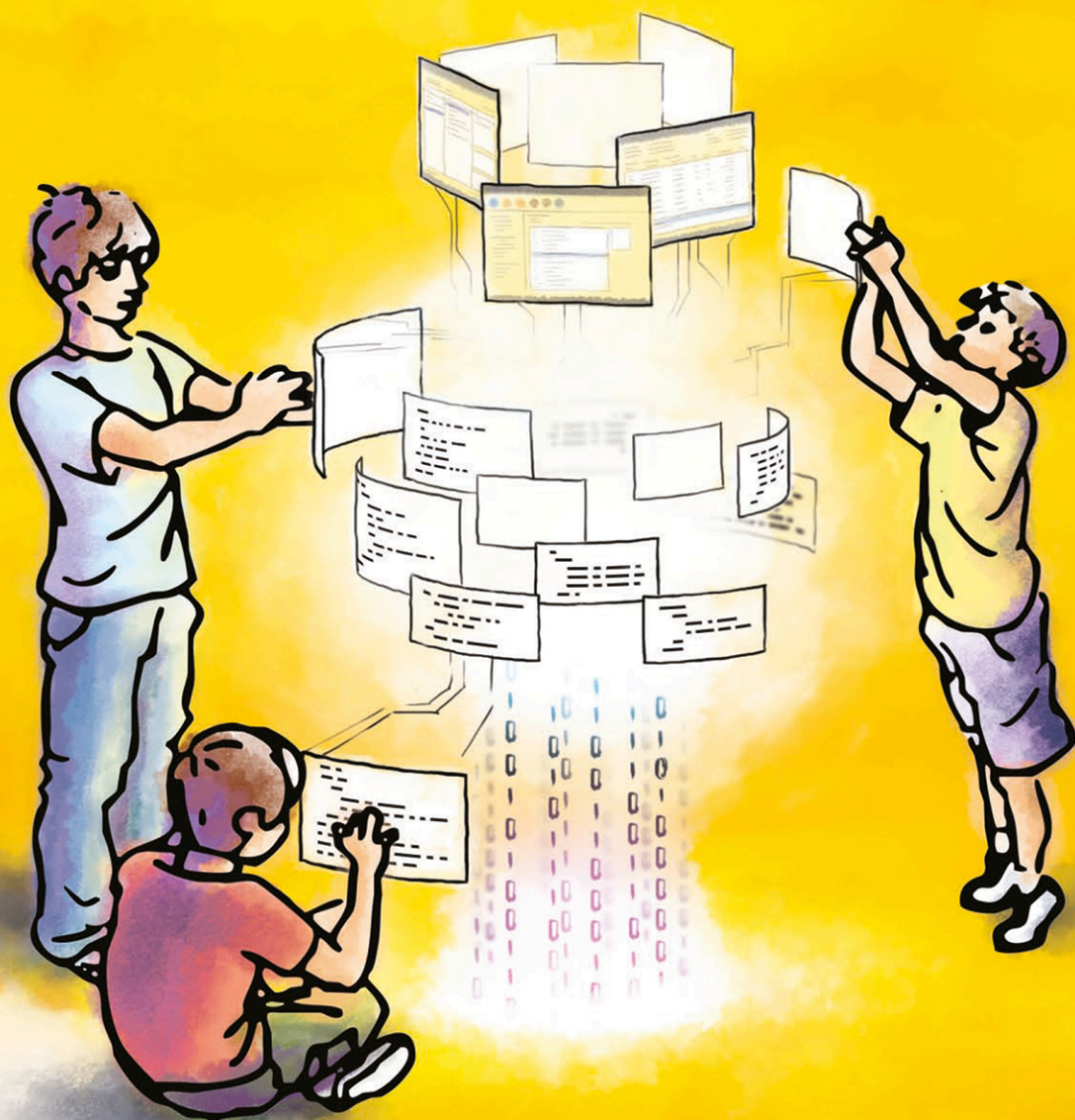
От ведущего ИТ-разработчика – Фирмы «1С»

Алгоритмы / Олимпиадное программирование

club.1c.ru

+7 (495) 688-90-02

teen@1c.ru





1С:Оценка качества образования. Школа

Трехуровневая
система
оценки качества
образования

Единые подходы
к внутренней
и внешней
оценке качества
образования

Прогнозирование
результатов
итоговой
государственной
аттестации



Программно-методическая система предназначена для оценки качества освоения образовательной программы на следующих уровнях: оценка индивидуальные достижений обучающихся, внутриклассное и внутришкольное оценивание.

Программа разработана на основе методики ведущего научного сотрудника Института управления образованием РАО, кандидата педагогических наук, доцента Н.Б. Фоминой.

Функциональные возможности

- Оценка индивидуального уровня освоения ФГОС.
- Аналитические расчеты успеваемости учащихся и качества образования.
- Анализ объективности оценивания индивидуальных образовательных достижений обучающихся.
- Персональный контроль профессиональной деятельности педагога с выявлением проблемных компонентов.
- Прогноз повышения качества образования, включая результаты государственных экзаменов (ОГЭ и ЕГЭ).

Преимущества использования

- Обеспечение индивидуализации образования, выявление способностей и предрасположенности каждого учащегося к определенному спектру дисциплин.
- Предоставление педагогам необходимой информации для практической деятельности (корректировка программ, выбор технологий обучения, выявление проблем в обучении).
- Предоставление руководителю данных, необходимых для анализа работы педагогического коллектива.