



Н. П. Панченко,

Дворец детского творчества, г. Дзержинск, Нижегородская область

МНОГОКРАТНО ПОВТОРЯЯ, УЧИМСЯ САМОСТОЯТЕЛЬНО МЫСЛИТЬ, ПОЗНАВАТЬ, ТВОРИТЬ*

Аннотация

Представлена методическая разработка лабораторной работы, содержащая комплект тренировочных тестов, заданий самоконтроля, заданий для обязательного и самостоятельного решения задач на циклы. Для всех задач описан алгоритм решения и дан код программ на языке программирования Паскаль. Задачи для самостоятельного решения имеют набор тестов. Разработка может быть адаптирована для любого языка программирования.

Ключевые слова: информатика, Паскаль, алгоритм, программа, цикл, олимпиада, ЕГЭ, оценка результата.

Под словом «цикл» — ряд повторений И нужных действий алгоритм. Познаешь только лишь в сравнении, В чем цикла смысл сокрыт.

Новые стандарты образования предъявляют высокие требования не только к учащимся, но, прежде всего, к педагогу. Основные условия образовательного процесса — сжатые сроки, огромный объем информации, жесткие требования к знаниям и умениям обучающихся. Такие запросы невозможно удовлетворить, используя только традиционные методы и средства обучения. Одним из способов развития творческой активности обучающихся являются задания, требующие выполнения исследовательской работы, включения в самостоятельный творческий поиск с постепенным усложнением заданий.

На примере циклов можно показать учащимся, что лучшие качества компьютеров проявляются не тогда, когда они используются для расчета значений сложных выражений, а когда многократно, с незначительными изменениями, повторяют сравнительно простые операции. Циклы позволяют записать такие действия в компактной форме. Само программирование имеет смысл, когда удастся найти циклический алгоритм. Циклы демонстрируют эффективность применения компьютеров. С помощью короткой циклической программы можно организовать выполнение большого количества действий.

Любой язык программирования опирается на три кита: следование (линейный алгоритм), ветвление (условный оператор), повторение (цикл). В любом языке программирования тема «Циклические алгоритмы» актуальна и является основой для изучения программирования.

Тема занятия: Операторы цикла в Паскале.

Цель занятия: формирование навыков практического применения приемов и способов решения задач циклических алгоритмов.

Задачи занятия:

обучающие:

- закрепление знаний, умений и навыков работы с операторами цикла в Паскале;
- формирование умений обучающихся работать с тестовыми материалами;

Контактная информация

Панченко Надежда Петровна, педагог дополнительного образования Дворца детского творчества, г. Дзержинск, Нижегородская область; *адрес:* 606023, Нижегородская область, г. Дзержинск, пл. Ленина, д. 1; *телефон:* (8313) 26-19-27; *e-mail:* pancenko@yandex.ru

N. P. Panchenko,
Kids Palace, Dzerzhinsk, Nizhny Novgorod Region

**REPEATING MANY TIMES
WE LEARN TO THINK
INDEPENDENTLY, TO EXPLORE,
TO CREATE**

Abstract

The article presents the laboratory work that contains a set of training tests, self-management tasks, tasks for the compulsory and independent solution of the tasks on the cycles. For all the tasks the algorithm for solving is given, code and programming in Pascal are described. Tasks for independent solution contain a set of tests. The work can be adapted for any programming language.

Keywords: informatics, Pascal, algorithm, program, cycle, Olympiad, Unified State Examination, assessment of result.

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_10-2016/

развивающие:

- развитие алгоритмического и логического мышления;
- развитие творческой активности, познавательного интереса;

воспитательные:

- мотивация к самостоятельной учебной деятельности — от постановки проблемы до ее практического решения;
- профессиональное самоопределение.

Тип занятия: комбинированное занятие.

Вид занятия: лабораторная работа.

Возраст учащихся: учащиеся соответствующего профиля учреждений дополнительного образования или учащиеся профильных групп десятых-одиннадцатых классов.

Оборудование занятия:

- маркерная доска, маркер;
- мультимедийный проектор;
- компьютеры, объединенные в локальную сеть;
- среда программирования — используется бесплатно распространяемая система программирования PascalABC, позволяющая посредством гиперссылок демонстрировать работу программ непосредственно из презентации.

Методическое обеспечение занятия*:

- учебно-методический комплекс:
 - презентация «Циклы в Паскале» (файл *Презентация.ppt*);
 - текст лабораторной работы (*Приложение 1* из файла *Приложения.doc*);
 - ответы (*Приложение 2* из файла *Приложения.doc*);
 - рейтинговая таблица для обучающихся (*Приложение 1, табл. 1* из файла *Приложения.doc*);
 - критерии оценивания программного продукта (*Приложение 1, табл. 2* из файла *Приложения.doc*);
 - сводная рейтинговая таблица с критериями оценивания (*Приложение 3* из файла *Приложения.doc*);
 - типичные ошибки (*Приложение 4* из файла *Приложения.doc*);
 - тексты программ (папка *Приложение 6*);
- комплекс упражнений для профилактики переутомления (*Приложение 5* из файла *Приложения.doc*);
- план занятия.

Методы и приемы обучения: словесные, наглядные, продуктивные (проблемные задачи), практические, поощрение.

Планируемая деятельность учащихся:

- проверка теоретических знаний — устный опрос, выполнение заданий самоконтроля;

- заполнение рейтинговой таблицы (*Приложение 1, табл. 1*);
- практическая работа на компьютере — применение теоретических знаний, умений и навыков в процессе выполнения заданий различного уровня сложности.

Формы организации учебной деятельности:

- индивидуальная работа;
- групповая работа.

Структура учебной деятельности (в условиях реализации ФГОС):

- учебная задача — цель, которую перед собой ставит учащийся;
- учебное действие — система существенных признаков понятия или алгоритм;
- самоконтроль — определение правильности выполненного действия;
- самооценка — определение степени соответствия образцу или качества выполненного действия.

Планируемые действия педагога:

- комментирует электронную презентацию;
- демонстрирует из презентации работу программ — предлагает один из возможных вариантов решения задач;
- просматривает у каждого учащегося выполнения заданий самоконтроля, обращает внимание учащихся на типичные ошибки (*Приложение 4*), заполняет сводную рейтинговую таблицу (*Приложение 3*);
- консультирует, ведет диалог;
- обсуждает с учащимися каждый этап самоконтроля, итоги выполненных заданий, отмечает и поощряет тех, кто успешно справился с ними;
- оценивает вместе с учащимися задания, выполненные на компьютере (*Приложение 1, табл. 2*).

Рекомендации по проведению занятия:

- презентация может демонстрироваться на экране для групповой работы или может быть передана по локальной сети на каждое рабочее место (компьютер) для работы в индивидуальном режиме;
- во время занятия необходимо соблюдать режим работы в компьютерном классе (*Приложение 5*);
- каждый учащийся получает для практической работы свой экземпляр лабораторной работы (*Приложение 1*), выполненная работа остается у обучающегося;
- в зависимости от уровня подготовки учащихся педагог может использовать проверочные задания не только в виде самоконтроля, но в форме педагогической проверки знаний;
- учащиеся отмечают выполненные задания в рейтинговой таблице (*Приложение 1, табл. 1*), сравнивают свои ответы с правильными ответами на слайдах;
- педагог может использовать взаимопроверку: обучающиеся обмениваются работами и проверяют работы друг друга;
- в затруднительных случаях задания по самоконтролю выполняются обучающимися на компьютере.

* Презентацию и приложение 6 можно скачать на сайте ИНФО, приложения 1–5 даны в конце статьи и на сайте ИНФО: http://infojournal.ru/journals/school/school_10-2016/

Формы контроля и оценки результатов:

- тестовые задания;
- устный опрос;
- самопроверка, взаимопроверка;
- выполнение заданий на компьютере.

Ожидаемые образовательные результаты:*предметные:*

- сформированность алгоритмического, логического и аналитического мышления; умений обучающихся писать и отлаживать программы циклической структуры;
- закрепление знаний, умений и навыков работы с операторами цикла в Паскале;
- сформированность умений работы обучающихся с тестовыми материалами;
- владение приемами и способами решения задач циклической структуры;

личностные:

- готовность и способность к самостоятельному поиску методов решения практических задач, принятию самостоятельных решений;
- мотивация обучающихся к практической деятельности в целях дальнейшего профессионального самоопределения;
- активная познавательная и творческая деятельность;

метапредметные:

- умение анализировать, формулировать цель, планировать, осуществлять самоконтроль и оценку своей деятельности;
- способность участвовать в обсуждениях, формулировать собственное мнение;
- умение сотрудничать с педагогом и другими обучающимися, определять способы взаимодействия, вести дискуссию;
- умение осуществлять совместную деятельность: согласование и координацию деятельности в подгруппах, объективное оценивание своего вклада в решение общих задач коллектива и определение своего уровня усвоения учебного материала;

интегрированные:

- умение обучающихся управлять собственной деятельностью — от цели до планируемого результата;
- умение выбрать способы действий в соответствии со своим индивидуальным стилем деятельности;
- умение ориентироваться в различных ситуациях, использовать способы и приемы по достижению поставленной цели.

Рекомендуемая литература для педагога.

1. *Зайдельман Я. Н., Ройтберг М. А.* Информатика. Подготовка к ЕГЭ в 2015 году. Диагностические работы. М.: МЦНМО, 2015.

2. *Златопольский Д. М.* Сборник задач по программированию. 3-е изд., перераб. и доп. СПб.: БХВ-Петербург, 2011.

3. *Кашаев С. М., Шерстнева Л. В.* Паскаль для школьников. Подготовка к ЕГЭ. СПб.: БХВ-Перербург, 2010.

4. *Панченко Н. П.* Компьютерный турнир «Проверь себя» // Информатика в школе. 2012. № 8.

5. *Панченко Н. П.* Лабораторно-практическая работа «Целочисленная арифметика в C/C++». <http://festival.1september.ru/articles/636786/>

6. *Панченко Н. П.* Урок информатики по теме «Оператор цикла в C/C++». <http://festival.1september.ru/articles/649122/>

7. *Панченко Н. П.* Чтобы научиться бегать, сначала нужно научиться ходить // Информатика в школе. 2014. № 8.

8. *Паронджанов В. Д.* Учись писать, читать и понимать алгоритмы. Алгоритмы для правильного мышления. Основы алгоритмизации. М.: ДМК Пресс, 2012.

9. *Шелест В. Д.* Программирование. СПб.: БХВ-Петербург, 2002.

Рекомендуемая литература для учащихся.

1. *Андреева Е. В.* Программирование — это так просто, программирование — это так сложно. Современный учебник программирования. М.: МЦНМО, 2009.

2. *Культин Н.* Turbo Pascal в задачах и примерах. СПб.: БХВ-Петербург, 2006.

3. *Поляков К. Ю., Еремин Е. Д.* Информатика. Углубленный уровень: учебник для 10 класса: в 2 ч. Ч. 2. М.: БИНОМ. Лаборатория знаний, 2013.

4. *Потопахин В. В.* Turbo Pascal. Освой на примерах. СПб.: БХВ-Петербург, 2005.

Рекомендуемые интернет-источники.

1. *Андреева Т. А.* Программирование на языке Паскаль. <http://old.intuit.ru/department/pl/plpascal/>

2. Клуб программистов. Материалы для изучения языков программирования. <http://www.programmersclub.ru/179/>

3. *Ларина Э. С.* Решение олимпиадных задач. <http://old.intuit.ru/department/school/olymp/>

4. Лучший курс обучения языку программирования Turbo Pascal. <http://www.pascal.ru/index.html>

5. Программирование. Паскаль. <http://www.pascal.helpov.net/>

6. Сайт подготовки к ЕГЭ и ОГЭ Д. Гушина. <http://inf.reshuege.ru/>

7. Сайт подготовки к ЕГЭ по информатике К. Полякова. <http://kpolyakov.spb.ru/school/ege.htm>

План занятия.

1. Организационный этап.
2. Подготовительный этап.
 - 2.1. Повторение.
 - 2.2. Самоконтроль.
3. Основной этап. Разбор задач.
 - 3.1. Вопросы для обсуждения.
4. Контрольный этап.
 - 4.1. Задачи для самостоятельного решения.
 - 4.2. Задачи повышенной сложности.
5. Итоговый этап.
 - 5.1. Подведение итогов занятия.
 - 5.2. Материалы для самоподготовки и самообразования.

Содержание и ход занятия

№ слайда	Задание	Задачи	Действия педагога	Рекомендации к выполнению заданий	Планируемая деятельность обучающихся	Практический результат
1. Организационный этап						
1		Организация внимания, настрой на предстоящую практическую работу	Приветствует учащихся, отмечает отсутствующих, знакомит с темой занятия		Приветствуют педагога, занимают рабочие места, готовят ручки, тетради	Активизация внимания, настрой на предстоящую работу
2		Знакомство с содержанием и целью занятия, правилами заполнения рейтинговой таблицы и оценкой выполняемых работ	Напоминает условные обозначения, раздает работы с заданиями	Данные условные обозначения используются во всех презентациях автора	Записывают тему занятия, подписывают свой экземпляр лабораторной работы	Готовность учащихся к активной учебно-познавательной деятельности
2. Подготовительный этап						
3	2.1. Повторение	Повторение тем, изученных на предыдущих занятиях	Обращает внимание учащихся на связь данной темы с предыдущими занятиями, психологически готовит учащихся к проведению самоконтроля	Оценивание заданий 1.1–1.3 производится на усмотрение педагога	Слушают объяснение педагога, вспоминают ранее изученный материал	Готовность выполнять задания по темам, изученным ранее
4	Задание 1.1	Проверка теоретических знаний	Производит устный опрос по ранее изученным темам	Теория, необходимая для выполнения лабораторной работы	Выполняют задания, сверяют свои ответы с ответами на слайде	Правильное написание определенных и служебных слов в тексте программы
5	Задание 1.2	Проверка умений: • понимать, что такое кратность, логические операции; • переводить алгоритм в программный код	К доске по желанию приглашает двух учеников, за проявление инициативы выполнить задание у доски педагог добавляет баллы	В случае затруднения педагог задает наводящие вопросы	Выполняют задание, сверяют ответ с ответом на слайде, отвечают на вопросы педагога, ведут с ним диалог	Понимание данных алгоритмов и правильность их применения
6	Задание 1.3	Повторение операций инкремента, декремента, отношений, логических операций, их обозначений в языке программирования Паскаль	Напоминает об особенностях использования операций инкремента, декремента, отличии обозначений логических операций	Обучающиеся используют обозначения операций из других языков программирования	Выполняют задание и сверяют ответы с ответами на слайде	Закрепление знаний
7	2.2. Самоконтроль	Определение степени усвоения программного материала учащимися	Рассказывает содержание данного этапа занятия, предлагает выполнить самоконтроль в форме тестирования. Напоминает о том, что необходимо заполнять личную рейтинговую таблицу, дающую возможность получить дополнительные баллы	Задания самоконтроля идут по нарастающему уровню сложности. Проверку можно осуществлять как по каждому заданию, так и после выполнения всех заданий самоконтроля	Выполняют задания по самоконтролю и сравнивают с ответами на слайдах	Развитие аналитического мышления. Умение применять полученные знания на практике

Продолжение таблицы

№ слайда	Задание	Задачи	Действия педагога	Рекомендации к выполнению заданий	Планируемая деятельность обучающихся	Практический результат
8	Задание 2.1	Обучение умения находить и исправлять ошибки	Предлагает выполнить задание, после его выполнения обращает внимание учащихся на типичные ошибки, предлагает исправить ошибки самостоятельно, в случае затруднения показывает правильные ответы на слайде		Выполняют задание. Сравнивают свои решения с ответом на слайде, исправляют ошибки, ведут диалог с педагогом, подсчитывают количество правильно выполненных заданий, заполняют рейтинговую таблицу	Отработка навыков и умений находить ошибки, их устранение, повышение мотивации к обучению, уверенности в собственных силах
9	Задание 2.2	Проверка умения записывать операции в сокращенном виде на языке программирования Паскаль	Предлагает выполнить задание. Отвечает на вопросы. Индивидуально консультирует. Показывает на слайде правильные ответы	Особенности записи операций в сокращенном виде. Подготовка к решению более сложных задач		Понимание особенностей записи и применения операций в сокращенном виде
10	Задание 2.3	Проверка умения по заданному алгоритму записать цикл		Через практическую работу подготовить к решению более сложных задач		Понимание структуры цикла
11	Задание 2.4	Применение знаний на практике, решение задач задания 8	Предлагает решить задачи на условный оператор, подготавливает учащихся к применению условных операторов в операторе цикла	Через практическую работу подготовить к решению задач ЕГЭ (задания 8, 20)		Умение решать практические задачи. Знакомство с задачами ЕГЭ. Приобретение опыта решения задач ЕГЭ. Повышение самооценки, уверенности в своих силах
12	Задание 2.5	Решение задач ЕГЭ (задание 8)	Предлагает решить задачи более сложного уровня. После выполнения заданий открывает правильные ответы, ведет диалог с учащимися			
13		Создание ситуации успеха	Поощряет учащихся и сообщает им, что предложенные задания взяты из заданий ЕГЭ (8, 20), выдает рекомендации по решению задач			
14		Подведение итогов подготовительного этапа. Акцентирование внимания на типичных ошибках, допущенных на этом этапе	Рассказывает о возможных ошибках в выполненных ранее заданиях (<i>Приложение 4</i>)	Основная ошибка в правильности записей логических операций, знака «равно» и знака «не равно»	Записывают информацию в тетради	Предупреждение возможных ошибок при выполнении практических заданий
3. Основной этап						
15	3.1. Разбор задач	Разбор задач для обязательного решения	Рассказывает о работе на этом этапе		Слушают педагога	Готовность к выполнению практических заданий на компьютере

Продолжение таблицы

№ слайда	Задание	Задачи	Действия педагога	Рекомендации к выполнению заданий	Планируемая деятельность обучающихся	Практический результат
16		Рекомендации по использованию циклов	Акцентирует внимание учащихся на возможных ошибках при написании программ (<i>Приложение 4</i>)		Записывают полученную информацию в тетради	Предупреждение возможных ошибок при написании программ циклической структуры
17	3.2. Вопросы для обсуждения	Подготовка к решению более сложных задач	Предлагает обсудить тему о циклах	При решении задач данного этапа обучающиеся должны знать и четко представлять структуру циклов	Отвечают устно, записывают ответы	Закрепление знаний о структуре циклов. Готовность учащихся к решению сложных задач
18	Задача 1	Разбор алгоритма и текста программы, подготовка к формам записи алгоритма: словесной и в виде программного кода, написание программы с использованием разных типов циклов	Напоминает о форме записи алгоритма — словесной. Указывает на соответствие алгоритма и текста программы. Подчеркивает, что текст программы — тоже одна из форм представления алгоритма, демонстрирует решение одной и той же задачи с разными циклами	Для решения одной и той же задачи можно применять разные циклы	Разбирают и анализируют алгоритм в словесном и программном видах, выполняют его на компьютере. Проверяют правильность работы программы на контрольных примерах	Знание правил записи алгоритма в словесном и программном кодах. Понимание того, что для решения одной и той же задачи можно применять разные типы циклов
19	Задача 2					
20	Задача 3	Обсуждение алгоритма решения задачи с применением цикла с предусловием	Обсуждает решение задачи, предлагает самим написать и отладить программу. Напоминает о том, что увеличить количество решенных задач и ускорить процесс написания программы можно за счет использования: <ul style="list-style-type: none"> • заготовки программы; • приемов работы с текстом: вставка, копирование и т. д.; • скорости набора текста программы; • знания и правильного написания операторов 	Педагог после обсуждения и решения задачи может вызвать желающих к доске или предложить учащимся выполнить задание самостоятельно	Обсуждают с педагогом решение задачи, пишут текст программы, показывают педагогу, выполняют работу на компьютере	Самостоятельное написание программ с использованием рациональных приемов работы, понимание возможности решения задач разными способами
4. Контрольный этап						
21–24	4.1. Задачи для самостоятельного решения	Самостоятельное решение задач	Предлагает задачи различного уровня сложности для самостоятельного решения, в том числе из ОГЭ и ЕГЭ. При затруднениях показывает алгоритм, демонстрирует работу программ из презентации	Программы учащиеся должны написать сами	Обучающиеся пишут тексты программ, показывают их педагогу, выполняют их на компьютере. Каждый обучающийся работает в индивидуальном режиме	Умение находить свои способы решения задач различного уровня сложности. Самостоятельное написание программы, проверка работы на заданных контрольных примерах и составление своих

Окончание таблицы

№ слайда	Задание	Задачи	Действия педагога	Рекомендации к выполнению заданий	Планируемая деятельность обучающихся	Практический результат
25–27	4.2. Задачи повышенной сложности	Подготовка к решению повышенной сложности задач	Данный тип задач педагог предлагает решить тем, кто успешно справился с предыдущими заданиями	Предлагаются задачи разного уровня сложности.	Каждый обучающийся выбирает задачи соответственно уровню своей подготовленности, возможностей и способностей	Готовность к самостоятельному решению задач повышенной сложности
5. Итоговый этап						
28	5.1. Подведение итогов занятия	Оценка практической работы учащихся	Заполняет сводную рейтинговую таблицу по материалам выполнения практических работ учащихся, добавляет поощрительные баллы, подводит итог занятия. Демонстрирует результаты рейтинга на слайде (Приложение 3)	Необходимо оценить работу учащихся и сделать выводы: <ul style="list-style-type: none"> • освоили ли они тему; • что вызвало особое затруднение; • что можно использовать на следующем занятии для исправления ошибок 	В конце занятия обучающиеся представляют педагогу данные о выполненных работах	Осознание и адекватная оценка учащимися собственных возможностей, умение анализировать и применять знания на практике
29–31	5.2. Материалы для самоподготовки и самообразования	Снять эмоциональное напряжение после выполненной работы, мотивировать учащихся к повышению уровня своих знаний через самообразование	Педагог подводит итоги занятия, предлагает учащимся высказать свое мнение о пословице. Рекомендует материалы для самоподготовки и пополнения знаний, знакомит с темой следующего занятия «Вложенные циклы»		Слушают педагога, обсуждают предложенные вопросы, высказывают свое понимание смысла пословицы, приводят аналоги пословицы	Учащиеся принимают к сведению информацию. Мотивация учащихся к дальнейшему пополнению своих знаний

Приложения

Приложение 1

Лабораторная работа «Операторы цикла в Паскале»

Фамилия, имя _____ группа _____ дата _____

Таблица 1

Рейтинг для учащихся

№ п/п	Задания	Баллы	Задачи для самостоятельного решения					
			4а	Баллы	4б	Баллы	5	Баллы
1	1.1		4.1		4.8		5.1	
2	1.2		4.2		4.9		5.2	
3	1.3		4.3		4.10		5.3	
4	2.1		4.4				5.4	
5	2.2		4.5				5.5	
6	2.3		4.6				5.6	
7	2.4		4.7					
8	2.5							
9	Итого:							
Всего баллов:								

Примечание: 1 балл — за правильно выполненный пункт задания 2.1–2.5 (максимум 50 баллов);
1 балл — за решенную задачу (максимум 7 баллов);
1–2 балла — за решенную задачу (максимум 6 баллов);
1–3 балла — за решенную задачу (максимум 15 баллов).

Критерии оценки выполненной работы для заданий 4 и 5

№ п/п	Критерии	Да	Нет	Частично
1	Программа выполняется			
2	Получен достоверный результат (соответствие поставленной задаче)			
3	Правильно оформлен текст программы (наличие комментариев, отступов)			
4	Использованы материалы новой темы			
5	Грамотно оформлен результат (есть подсказки и рационально использован экран)			
6	Использованы ранее написанные программы (заготовки)			
7	Использованы рациональные приемы работы с текстом программы (копирование, вставка, удаление и т. д.)			
8	Оригинальность решения и творческие находки			
9	Применены операторы, не изученные ранее			
10	Компактность программы			
11	Отсутствуют грамматические ошибки			

1. Повторение**Задание 1.1.** Устный опрос.

- Какие числа называются натуральными? _____
- Какие числа называются целыми? _____
- Как описывается целый тип данных в языке программирования Паскаль? _____
- Операции целочисленного деления в Паскале: _____
- В чем заключается преимущество процедур `inc(i)` и `dec(i)`, какими операциями их можно заменить? _____
- Конструкции, реализующие циклы в Паскале _____

Задание 1.2. Перевести на язык программирования следующие команды:

№ п/п	Задание	Ответ
1	Число n кратно 4	
2	Число n кратно 7	
3	Число n оканчивается на 4	
4	Число n оканчивается на 7	
5	Число n кратно 4 и оканчивается на 7	
6	Число n кратно 7 и оканчивается на 4	
7	Число n кратно 7 или оканчивается на 4	
8	Число n кратно 7 или кратно 4	
9	Число n кратно 7 и кратно 4	
10	Число n оканчивается на 4 или на 7	

Вопросы для обсуждения:

- Какой тип данных должен быть у переменной n ? _____
- Может ли быть у n вещественный тип данных? _____
- Как привести вещественный тип данных к целому типу данных? _____

Задание 1.3. Продолжить:

№ п/п	Задание	Ответ
1	inc(i) операция	
2	inc(i, 3) увеличивает значение переменной i на	
3	dec(i) операция	
4	dec(i, 4) уменьшает значение переменной на	
5	Операции inc(i) и dec(i) могут применяться только к	
6	Операции inc(i) и dec(i) не могут применяться к	
7	Операция отношения «равно» записывается как	
8	Операция отношения «не равно» записывается как	
9	Логическая операция «дизъюнкция, логическое сложение, ИЛИ» записывается как	
10	Логическая операция «конъюнкция, логическое умножение, И» записывается как	

2. Самоконтроль**Задание 2.1.** Найти и объяснить ошибки.

№ п/п	Пример	Ответ	Пояснения
1	for i=1 to n do;		
2	for i:=n to 1 do		
3	for i=1 to n do step 1		
4	while a<5 do; a:=a+1; b:=b+a;		
5	repeat a<5 a:=a+1; b:=b+a; until;		
6	while c<k or k=0 do		
7	while c<k s:=s+k; inc(s);		
8	if a><b a:=a+b;b:=b+3		
9	if a=b or a>5 then a:=a+b;		
10	if (a=>b) c:=a+b;		

Задание 2.2. Записать операции в сокращенном виде.

№ п/п	Полный вид	Сокращенный вид	№ п/п	Полный вид	Сокращенный вид
1	n:=n+1		6	n:=n-a	
2	n:=n+4		7	n:=n+13 mod 5	
3	n:=n+a		8	n:=n+a mod 5	
4	n:=n-1		9	n:=n+a*k;	
5	n:=n-4		10	n:=n-a div 10	

Задание 2.3. Записать цикл, используя конструкцию for ... do.

№ п/п	Задание	Ответ
1	$p = 1\ 2\ 3 \dots n$	
2	$p = 2\ 4\ 6 \dots 2n$	
3	$p = 1\ 3\ 5 \dots (2n - 1)$	
4	$s = 1 + 2 + 3 + \dots + n$	
5	$s = 5 + 6 + 7 + \dots + (n + 5)$	
6	$s = 2 + 4 + 6 + \dots + 2n$	
7	$s = 1 + 3 + 5 + \dots + (2n - 1)$	
8	$s = 1 + 4 + 7 + \dots + (3n - 2)$	
9	$s = n + (n - 1) + (n - 2) + \dots + 1$	
10	$s = 1/2 + 1/4 + 1/6 + \dots + 1/(2n)$	

Вопросы для обсуждения:

- Обосновать, какой тип данных должен быть объявлен у переменных p и s . _____
- Какие значения должны быть у переменных p и s до начала цикла? _____
- Способы присваивания первоначальных значений переменным p и s . _____
- Особенности деления в Паскале. _____

Задание 2.4. Определить значение переменной c после выполнения следующего фрагмента программы:

№ п/п	Задание	Ответ
1	$a:=30; b:=6; a:=a \operatorname{div} 2+b; \text{if } (a>b) \text{ then } c:=a-3*b \text{ else } c:=a+3*b;$	
2	$a:=30; b:=6; a:=a \operatorname{div} 2+b; \text{if } (a+b) \text{ then } c:=a+4*b \text{ else } c:=a+4+b;$	
3	$a:=22; b:=3; a:=-2*b+a \operatorname{div} 2; \text{if } (a<b) \text{ then } c:=4*b-3*a \text{ else } c:=3+a+4+b;$	
4	$a:=15; b:=30; b:=a+2+b \operatorname{div} 2; \text{if } (a+b) \text{ then } c:=3+b - a \operatorname{div} 3 \text{ else } c:=3+a+4+b;$	
5	$a:=30; b:=10; a:=a \operatorname{div} b+2; \text{if } (a+b) \text{ then } c:=a+4+b \text{ else } c:=a+4+b;$	
6	$a:=30; b:=14; a:=a-2+b; \text{if } (a+b) \text{ then } c:=b+2+a \text{ else } c:=b+2+a;$	
7	$a:=60; b:=20; a:=a \operatorname{div} b+2; \text{if } (a+b) \text{ then } c:=a+4+b \text{ else } c:=a+4+b;$	
8	$a:=40; b:=6; a:=a+3 \operatorname{div} b; \text{if } (a+b) \text{ then } c:=a+5+b \text{ else } c:=a+5+b;$	
9	$a:=30; b:=6; a:=a \operatorname{div} 5+(b+3); \text{if } (2*a+b) \text{ then } c:= a+5*b+2 \text{ else } c:=a+5*b+2;$	
10	$a:=120; b:=100; a:=a+b \operatorname{div} 2; \text{if } (b<a) \text{ then } c:=b+a \text{ else } c:=b+a \operatorname{div} 2;$	

Задание 2.5. Определить, что будет напечатано в результате работы фрагмента программы.

№ п/п	Задание	Ответ	№ п/п	Задание	Ответ
1	$n:=3; s:=0; \text{while } n<=7 \text{ do} \text{ begin } s:=s+n; n:=n+1; \text{end}; \text{writeln}('s=', s);$		4	$s:=0; n:=3; \text{while } n<=365 \text{ do} \text{ begin } n:=n+33; s:= s+5; \text{end}; \text{writeln}('s=', s);$	
2	$s:=0; n:=0; \text{while } n<30 \text{ do} \text{ begin } n:=n+3; s:=s+n; \text{end}; \text{writeln}('s=', s);$		5	$n:=4; s:=0; \text{while } (n<=8) \text{ do} \text{ begin } s:=s+15; n:=n+1; \text{end}; \text{writeln}('s=', s);$	
3	$s:=3; n:=1; \text{while } n<25 \text{ do} \text{ begin } s:=s+n; n:=n+2; \text{end}; \text{writeln}('s=', s);$		6	$n:=0; s:=512; \text{while } s>=0 \text{ do} \text{ begin } s:=s-20; n:=n+1; \text{end}; \text{writeln}('s=', s);$	

3. Разбор задач

Вопросы для обсуждения:

- Для чего используются операторы цикла? _____
- Какие циклы относятся к итерационным? _____
- Какие циклы относятся к арифметическим? _____
- В каких случаях предпочтительнее использовать цикл с параметром, а в каких — условные циклы? _____

Задача 1. Используя цикл `for`, написать программу, которая определяет среднее арифметическое всех чисел на отрезке $[a; b]$.

Алгоритм	Программа	Контрольные примеры
1. Описать переменные с указанием типов данных	<code>var a, b, i, k: integer; s: real;</code>	$a = 5;$ $b = 10.$ Ответ: $s = 7.5.$ $a = 200;$ $b = 250.$ Ответ: $s = 225.$
2. Начало программы	<code>begin</code>	
3. Очистить переменные для суммирования	<code>s:=0; k:=0;</code>	
4. Задать интервал $[a; b]$	<code>write('Задай a, b'); readln(a,b);</code>	
5. Записать цикл с параметром	<code>for i:=a to b do</code>	
6. Подсчитать сумму	<code>begin inc(s, i); {s:=s+i};</code>	
7. Подсчитать количество чисел в сумме	<code>inc(k); {k:=k+1} end;</code>	
8. Найти среднее арифметическое	<code>s:= s/k;</code>	
9. Вывести ответ s	<code>writeln('s=', s:3:2);</code>	
10. Конец программы	<code>end.</code>	

Задача 2. Используя цикл `while`, написать программу, которая определяет среднее арифметическое всех чисел на отрезке $[a; b]$.

Алгоритм	Программа	Контрольные примеры
1. Описать переменные с указанием типов данных	<code>var i, a, b, k: integer; s: real;</code>	$a = 5;$ $b = 10.$ Ответ: $s = 7.5.$ $a = 200;$ $b = 250.$ Ответ: $s = 225.$
2. Начало программы	<code>begin</code>	
3. Обнулить переменные для суммирования s, k	<code>s:=0; k:=0;</code>	
4. Задать интервал значений	<code>write('Задай a, b'); readln(a,b);</code>	
5. Пока $a \leq b$ выполнять цикл, в противном случае перейти к п. 10	<code>while (a<=b) do</code>	
6. Подсчитать сумму	<code>begin s:=s+a; {inc(s,a);</code>	
7. Подсчитать количество чисел в сумме	<code>k:=k+1; {inc(k);</code>	
8. Увеличить счетчик цикла на 1	<code>a:=a+1; {inc(a);</code>	
9. Перейти к п. 5	<code>end;</code>	
10. Подсчитать среднее арифметическое	<code>s:=s/k;</code>	
11. Вывести результат	<code>writeln('s=', s:3:2);</code>	
12. Конец программы	<code>end.</code>	

Задача 3. Используя цикл `repeat`, решить предыдущую задачу с проверкой ввода интервала.

Вопросы для обсуждения:

- Какой оператор цикла используют для проверки ввода данных? _____
- Как запишется этот цикл? _____
- Можно ли условный цикл заменить на цикл for? _____
- Как можно обойтись без счетчика элементов ($k = k + 1$)? _____
- Сколько раз выполнится цикл, если $a = 3$, а $b = 7$? _____
- Нужны ли операторные скобки begin ... end в операторе repeat? _____

Задание. Написать программу и выполнить на компьютере, используя контрольные примеры предыдущей задачи.

4. Задачи для самостоятельного решения

4а. Задачи для обязательного решения.

Задача 4.1. Напечатать 10 раз столбиком и строкой слово «komp».

Задача 4.2. По введенному числу n вычислить $n!$.

Входные данные	Выходные данные
3	6
5	120
6	720

Задача 4.3. По трем заданным числам определить и вывести на экран число, имеющее в своем составе больше всего единиц.

Входные данные	Выходные данные
31122 1234 411115	411115
112 1113 23456	1113
1031 10101 12345	10101

Задача 4.4. Определить, является ли введенное число палиндромом.

Входные данные	Выходные данные
123321	yes
3215123	yes
123456	no

Задача 4.5. Дано десятичное число. Определить, сколько единиц и нулей в его двоичном представлении.

Входные данные	Выходные данные
45	4 2
100	3 4
7	3 0

Задача 4.6. Вычислить: $s = x/1! + x/2! + x/3! + \dots + x/n!$. (Первое введенное число — x , второе — n .)

Входные данные	Выходные данные
1 10	1.71828
2 10	3.43656
3 20	5.15485

Задача 4.7. Написать программу, которая находит наибольшее произведение двух чисел из последовательности натуральных чисел. Программа получает на вход натуральные числа, количество введенных чисел неизвестно, последовательность чисел заканчивается числом 0. Количество чисел не превышает 1000. Введенные числа не превышают 30 000.

Входные данные	Выходные данные
7 8 9 10 0	90
4 9 35 7 16 5 18 3 0	630

46. Задачи для подготовки к ОГЭ и ЕГЭ.

Задача 4.8. Написать программу, которая в последовательности целых чисел определяет наибольшее произведение двух чисел. Программа получает на вход целые числа, количество введенных чисел неизвестно, но не больше 1000. Последовательность чисел заканчивается числом 0. Введенные числа по модулю не превышают 30 000.

Задача 4.9. Написать программу, которая из введенного с клавиатуры натурального числа удаляет все цифры 5. Программа получает на вход целое число, не превышающее 30 000. Программа должна вывести одно число — число, полученное из исходного, после удаления всех цифр 5 из его записи.

Задача 4.10. Написать программу, которая в последовательности целых чисел находит наименьшие порядковые номера двух соседних чисел, произведение которых меньше 50. Программа получает на вход целые числа, количество введенных чисел неизвестно. Ввод последовательности прекращается, как только произведение введенного и предыдущего числа меньше 30. количество чисел не превышает 1000. Введенные числа по модулю не превышают 100. Гарантируется, что в последовательности существуют числа, удовлетворяющие условию задачи. Программа должна вывести два числа (разделенных пробелом) — порядковые номера двух соседних чисел, произведение которых меньше 50.

5. Задачи повышенной сложности

Задача 5.1. Для заданных чисел a и b ($a > 1$) найти такое наименьшее целое натуральное число k , что $a^k > b$.

Задача 5.2. Для заданного натурального числа n найти такое наименьшее число k , что $k! \geq n$.

Задача 5.3. Написать программу, которая в последовательности целых чисел, не превышающих по модулю 30 000, находит третье положительное число и его порядковый номер. Программа должна вывести два числа (разделенных пробелом) — это число и его порядковый номер. Если такого числа нет — сообщение «no».

Задача 5.4. Известно, что число делится на 3, когда сумма его цифр делится на 3. Проверить этот признак на примере заданного натурального числа n .

Входные данные	Выходные данные
34 -9 -7 16 -5 18 -3 35 -4 0	1190
34 9 2 7 -16 5 -18 3 35 4 0	1190
9 -2 7 -16 -18 -35 -4 0	630

Входные данные	Выходные данные
25455	24
15256	126
25551	21

Входные данные	Выходные данные
10 8 9 6 11 4	5 6
11 5 3 15	2 3
9 7 8 6 1	3 4

Входные данные	Выходные данные
3 20	2
7 120	2
2 240	7

Входные данные	Выходные данные
7	4
6	3
23	4

Входные данные	Выходные данные
2 3 -4 -5 -6 0	no
-1 2 -4 3 6 0	6 5
-1 2 -4 3 -6 7 6 0	7 6

Входные данные	Выходные данные
333	Yes
27	Yes
121	No

Задача 5.5. Посчитать сумму таких натуральных чисел, у которых удалены все четные цифры.

Входные данные	Выходные данные
123 1221 2344 0	27
122 426 1681 0	12
724 1255 111 0	118

Задача 5.6. Написать программу, которая в последовательности целых чисел, не превышающих по модулю 500, находит и выводит среднее арифметическое всех положительных чисел, которые кратны первому введенному числу. Если таких чисел нет, нужно вывести сообщение «по».

Входные данные	Выходные данные
3 -6 7 -8 14 0	по
3 6 -6 9 8 -3 0	7.5

Вопросы для обсуждения:

- В чем отличие цикла while от цикла repeat? _____
- Какой цикл удобнее использовать для проверки корректности вводимых данных? _____
- Когда тело цикла заключается в операторные скобки? _____
- В чем особенность применения цикла for? _____
- В чем особенности применения оператора цикла while? _____
- В чем особенности применения оператора цикла repeat? _____
- Как вы понимаете пословицу «Аз да буки, а потом науки»? _____

Материалы для самоподготовки и самообразования

Рекомендуемая литература для учащихся.

1. *Андреева Е. В.* Программирование — это так просто, программирование — это так сложно. Современный учебник программирования. М.: МЦНМО, 2009.
2. *Культин Н.* Turbo Pascal в задачах и примерах. СПб.: БХВ-Петербург, 2006.
3. *Поляков К. Ю., Еремин Е. Д.* Информатика. Углубленный уровень: учебник для 10 класса: в 2 ч. Ч. 2. М.: БИНОМ. Лаборатория знаний, 2013.
4. *Потопахин В. В.* Turbo Pascal. Освой на примерах. СПб.: БХВ-Петербург, 2005.

Рекомендуемые интернет-источники.

1. *Андреева Т. А.* Программирование на языке Паскаль. <http://old.intuit.ru/department/pl/plpascal/>
2. Клуб программистов. Материалы для изучения языков программирования. <http://www.programmersclub.ru/179/>
3. *Ларина Э. С.* Решение олимпиадных задач. <http://old.intuit.ru/department/school/olymp/>
4. Лучший курс обучения языку программирования Turbo Pascal. <http://www.pascal.ru/index.html>
5. Программирование. Паскаль. <http://www.pascal.helpov.net/>
6. Сайт подготовки к ЕГЭ и ОГЭ Д. Гушина. <http://inf.reshuege.ru/>
7. Сайт подготовки к ЕГЭ по информатике К. Полякова. <http://kpolyakov.spb.ru/school/ege.htm>

Приложение 2

Ответы

1. Повторение

Задание 1.1. Устный опрос.

- Какие числа называются натуральными? (*Натуральными числами называются числа, полученные при счете.*)
- Какие числа называются целыми? (*Целыми числами называются натуральные числа, их противоположные и ноль.*)
- Как описывается целый тип данных в языке программирования Паскаль? (*word, byte, integer, longint, shortint.*)
- Операции целочисленного деления в Паскале. (*div, mod.*)
- В чем заключается преимущество процедур inc(i) и dec(i), какими операциями их можно заменить? (*Процедуры выполняются быстрее, чем аналогичные им операции: i:=i+1; i:=i-1.*)
- Конструкции, реализующие циклы в Паскале. (*for, while ... do, repeat ... until.*)

Задание 1.2. Перевести на язык программирования следующие команды:

№ п/п	Задание	Ответ
1	Число n кратно 4	$N \bmod 4 = 0$
2	Число n кратно 7	$N \bmod 7 = 0$
3	Число n оканчивается на 4	$N \bmod 10 = 4$
4	Число n оканчивается на 7	$N \bmod 10 = 7$
5	Число n кратно 4 и оканчивается на 7	$(n \bmod 4 = 0) \text{ and } (n \bmod 10 = 7)$
6	Число n кратно 7 и оканчивается на 4	$(n \bmod 7 = 0) \text{ and } (n \bmod 10 = 4)$
7	Число n кратно 7 или оканчивается на 4	$(n \bmod 7 = 0) \text{ or } (n \bmod 10 = 4)$
8	Число n кратно 7 или кратно 4	$(n \bmod 7 = 0) \text{ or } (n \bmod 4 = 0)$
9	Число n кратно 7 и кратно 4	$(n \bmod 7 = 0) \text{ and } (n \bmod 4 = 0)$
10	Число n оканчивается на 4 или на 7	$(n \bmod 10 = 4) \text{ or } (n \bmod 10 = 7)$

Вопросы для обсуждения:

- Какой тип данных должен быть у переменной n ? (Целый, так как операции div и mod работают только с целочисленными типами данных.)
- Может ли быть у n вещественный тип данных? (Нет.)
- Как привести вещественный тип данных к целому типу данных? (Использовать стандартные функции округления $trunc$, $round$).

Задание 1.3. Продолжить:

№ п/п	Задание	Ответ
1	$inc(i)$ операция	инкремента, увеличивает значение i на 1.
2	$inc(i, 3)$ увеличивает значение переменной i на	3
3	$dec(i)$ операция	декремента, уменьшает значение i на 1.
4	$dec(i, 4)$ уменьшает значение переменной на	4
5	Операции $inc(i)$ и $dec(i)$ могут применяться только к	целочисленным типам данных.
6	Операции $inc(i)$ и $dec(i)$ не могут применяться к	вещественным типам данных
7	Операция отношения «равно» записывается как	=
8	Операция отношения «не равно» записывается как	<>
9	Логическая операция «дизъюнкция, логическое сложение, ИЛИ» записывается как	or
10	Логическая операция «конъюнкция, логическое умножение, И» записывается как	and

2. Самоконтроль

Задание 2.1. Найти и объяснить ошибки.

№ п/п	Пример	Ответ	Пояснения
1	<code>for i=1 to n do;</code>	<code>for i:=1 to n do</code>	Пропущено «:», после «do» «;» не ставится
2	<code>for i:=n to 1 do</code>	<code>for i:=n downto 1 do</code>	Нарушена конструкция цикла по убыванию
3	<code>for i=1 to n do step 1</code>	<code>for i:=1 to n do</code>	Пропущено «:», в Паскале шаг задается автоматически и равен 1 (-1)

Окончание таблицы

№ п/п	Пример	Ответ	Пояснения
4	<code>while a<5 do; a:=a+1; b:=b+a;</code>	<code>while a<5 do begin a:=a+1; b:=b+a; end;</code>	После «do» «;» не ставится. Пропущены операторные скобки <code>begin ... end</code>
5	<code>repeat a<5 a:=a+1; b:=b+a; until;</code>	<code>repeat a:=a+1; b:=b+a; until a<5;</code>	В конструкции цикла <code>repeat</code> условие ставится в конце цикла
6	<code>while c<k or k=0 do</code>	<code>while (c<k) or (k=0) do</code>	Операции отношения должны быть заключены в скобки
7	<code>while c<k s:=s+k; inc(s);</code>	<code>while c<k do begin s:=s+k; inc(k); end;</code>	Пропущено служебное слово «do», операторные скобки. В теле цикла нет переменной, влияющей на изменение условий
8	<code>if a>b a:=a+b;b:=b+3</code>	<code>if a>b then begin a:=a+b; b:=b+3 end;</code>	Неправильно записана операция неравенства, пропущено служебное слово «then» и операторные скобки
9	<code>if a=b or a>5 then a:=a+b;</code>	<code>if (a=b) or (a>5) then a:=a+b;</code>	Операции отношения в сложных условиях должны быть заключены в скобки
10	<code>if (a>b) c:=a+b;</code>	<code>if (a>b) then c:=a+b;</code>	Неправильно записана операция отношения, пропущено служебное слово «then»

Задание 2.2. Записать операции в сокращенном виде.

№ п/п	Полный вид	Сокращенный вид	№ п/п	Полный вид	Сокращенный вид
1	<code>n:=n+1</code>	<code>inc(n)</code>	6	<code>n:=n-a</code>	<code>dec(n,a)</code>
2	<code>n:=n+4</code>	<code>inc(n,4)</code>	7	<code>n:=n+13 mod 5</code>	<code>inc(n, 13 mod 5)</code>
3	<code>n:=n+a</code>	<code>inc(n,a)</code>	8	<code>n:=n+a mod 5</code>	<code>inc(n, a mod 5)</code>
4	<code>n:=n-1</code>	<code>dec(n)</code>	9	<code>n:=n+a*k;</code>	<code>inc(n, a*k)</code>
5	<code>n:=n-4</code>	<code>dec(n,4)</code>	10	<code>n:=n-a div 10</code>	<code>dec(n, a div 10)</code>

Задание 2.3. Записать цикл, используя конструкцию `for ... do`.

№ п/п	Задание	Ответ
1	$p = 1\ 2\ 3 \dots n$	<code>p:=1; for i:=1 to n do p:=p*i;</code>
2	$p = 2\ 4\ 6 \dots 2n$	<code>p:=1; for i:=1 to n do p:=p*i*2;</code>
3	$p = 1\ 3\ 5 \dots (2n - 1)$	<code>p:=1; for i:=1 to n do p:=p*(i*2-1);</code>
4	$s = 1 + 2 + 3 + \dots + n$	<code>s:=0; for i:=1 to n do inc(s,i);</code>
5	$s = 5 + 6 + 7 + \dots + (n + 5)$	<code>s:=0; for i:=5 to n+5 do inc(s,i);</code>
6	$s = 2 + 4 + 6 + \dots + 2n$	<code>s:=0; for i:=1 to n do inc(s,i*2);</code>
7	$s = 1 + 3 + 5 + \dots + (2n - 1)$	<code>s:=0; for i:=1 to n do inc(s,i*2-1);</code>
8	$s = 1 + 4 + 7 + \dots + (3n - 2)$	<code>s:=0; for i:=1 to n do inc(s,i*3-2);</code>
9	$s = n + (n - 1) + (n - 2) + \dots + 1$	<code>s:=0; for i:=0 to n - 1 to inc(s, n-i);</code>
10	$s = 1/2 + 1/4 + 1/6 + \dots + 1/(2n)$	<code>s:=0; for i:=1 to n s:=s + 1/(2*i);</code>

Вопросы для обсуждения:

- Обосновать, какой тип данных должен быть объявлен у переменных p и s . (*Везде, кроме задания 10, тип данных должен быть целым.*)
- Какие значения должны быть у переменных p и s до начала цикла? ($p:=1, s:=0$.)
- Особенности деления в Паскале. (*Для получения целочисленного результата должны быть использованы операции целочисленного деления div и mod .*)

Задание 2.4. Определить значение переменной c после выполнения следующего фрагмента программы:

№ п/п	Задание	Ответ
1	$a:=30; b:=6; a:=a \div 2+b; \text{if } (a>b) \text{ then } c:=a-3*b \text{ else } c:=a+3*b;$	$c = 72$
2	$a:=30; b:=6; a:=a \div 2+b; \text{if } (a+b) \text{ then } c:=a+4*b \text{ else } c:=a+4+b;$	$c = 66$
3	$a:=22; b:=3; a:=-2*b+a \div 2; \text{if } (a<b) \text{ then } c:=4*b-3*a \text{ else } c:=3+a+4+b;$	$c = 27$
4	$a:=15; b:=30; b:=a+2+b \div 2; \text{if } (a+b) \text{ then } c:=3+b - a \div 3 \text{ else } c:=3+a+4+b;$	$c = -15$
5	$a:=30; b:=10; a:=a \div b+2; \text{if } (a+b) \text{ then } c:=a+4+b \text{ else } c:=a+4+b;$	$c = 46$
6	$a:=30; b:=14; a:=a-2+b; \text{if } (a+b) \text{ then } c:=b+2+a \text{ else } c:=b+2+a;$	$c = 10$
7	$a:=60; b:=20; a:=a \div b+2; \text{if } (a+b) \text{ then } c:=a+4+b \text{ else } c:=a+4+b;$	$c = 86$
8	$a:=40; b:=6; a:=a+3 \div b; \text{if } (a+b) \text{ then } c:=a+5+b \text{ else } c:=a+5+b;$	$c = 50$
9	$a:=30; b:=6; a:=a \div 5+(b+3); \text{if } (2*a+b) \text{ then } c:= a+5*b+2 \text{ else } c:=a+5*b+2;$	$c = 35$
10	$a:=120; b:=100; a:=a+b \div 2; \text{if } (b<a) \text{ then } c:=b+a \text{ else } c:=b+a \div 2;$	$c = 270$

Задание 2.5. Определить, что будет напечатано в результате работы фрагмента программы.

№ п/п	Задание	Ответ	№ п/п	Задание	Ответ
1	$n:=3; s:=0; \text{while } n<=7 \text{ do } \text{begin } s:=s+n; n:=n+1; \text{end}; \text{writeln('s=', s);}$	25	4	$s:=0; n:=3; \text{while } n<=365 \text{ do } \text{begin } n:=n+33; s:= s+5; \text{end}; \text{writeln('s=', s);}$	60
2	$s:=0; n:=0; \text{while } n<30 \text{ do } \text{begin } n:=n+3; s:=s+n; \text{end}; \text{writeln('s=', s);}$	165	5	$n:=4; s:=0; \text{while } (n<=8) \text{ do } \text{begin } s:=s+15; n:=n+1; \text{end}; \text{writeln('s=', s);}$	75
3	$s:=3; n:=1; \text{while } n<25 \text{ do } \text{begin } s:=s+n; n:=n+2; \text{end}; \text{writeln('s=', s);}$	147	6	$n:=0; s:=512; \text{while } s>=0 \text{ do } \text{begin } s:=s-20; n:=n+1; \text{end}; \text{writeln('s=', s);}$	-8

3. Разбор задач

Вопросы для обсуждения:

- Для чего используются операторы цикла? (*Для организации многократно повторяющихся вычислений.*)
- Какие циклы относятся к итерационным? (*Итерационными циклами являются условные циклы: с условием и предусловием.*)
- Какие циклы относятся к арифметическим? (*Арифметические, или счетные, циклы — циклы со счетчиком или параметром цикла.*)
- В каких случаях предпочтительнее использовать цикл с параметром, а в каких условные циклы? (*Цикл с параметром применяется при известном числе повторений, в остальных случаях предпочтительнее использовать условные циклы.*)

Задача 3. Решить предыдущую задачу с проверкой ввода интервала.

Вопросы для обсуждения:

- Какой оператор цикла используют для проверки ввода данных? (*Repeat.*)
- Как запишется этот цикл? (*Repeat write(a,b); readln(a,b) until a<b;*)
- Можно ли условный цикл заменить на цикл for? (*Нет, не всегда.*)
- Как можно обойтись без счетчика элементов ($k = k + 1$)? (*Заменить процедурой inc(i).*)
- Сколько раз выполниться цикл, если $a = 3$, а $b = 7$? (*Пять раз.*)
- Нужны ли операторные скобки begin ... end в операторе repeat? (*Нет, данная конструкция позволяет обходиться без них.*)

Листинг программы.

```

{z3r}
var
  i, a, b, k: integer;
  s: real;
begin
  s:=0; k:=0;

repeat
  write('Задай a, b');
  readln(a,b);
until a<b;
repeat
  s:=s+a; {inc(s,a)

k:=k+1; {inc(k)}
a:=a+1; {inc(a)}
until a>b;
s:=s/k;
writeln('s=', s:3:2);
end.

```

4. Задачи для самостоятельного решения

4а. Задачи для обязательного решения.

Задача 4.1. Алгоритм. Для вывода в столбик использовать writeln (перевод курсора на следующую строку), для вывода в строку — write. writeln заменить на write.

Начало программы	while	repeat	for
{p4_1} var i: integer; begin	i:=1; while (i<=10) do begin writeln('komp'); end; end.	i:=1; repeat writeln('komp'); inc(i); { i:=i+1;} until i>10; end.	for i:=1 to 10 do begin writeln('komp'); inc(i); { i:= i+1;} end; end.

Задача 4.2. Алгоритм. Применить алгоритм нахождения произведения: $f:=f*i$. Переменной f, в которой будет храниться значение произведения, не забывать в начале присваивать значение 1 ($f:=1$).

Начало программы	while	repeat	for
{p4_2} var i, n: integer; f: longint; begin write('n:='); readln(n);	writeln('цикл while'); i:=1; f:=1; while (i<=n) do begin f:=f*i; inc(i); end; writeln('n', '!', f:10); end.	writeln('цикл repeat'); i:=1; f:=1; repeat f:=f*i; inc(i); until i>n; writeln('n', '!', f:10); end.	writeln('цикл for'); f:=1; for i:=1 to n do f:=f*i; writeln('n', '!', f:10); end.

Задача 4.3. Алгоритм. Для хранения вводимых данных ввести дополнительные переменные. Используя операции целочисленного деления и цикл, найти количество единиц в каждом числе. Далее решение задачи сведется к алгоритму нахождения наибольшего из трех чисел.

Начало программы	while	repeat
{p4_3} var a, b, c, z, k_a, k_b, k_c: integer; a1, b1, c1: integer; begin a:=0; k_b:=0; k_c:=0; write('a, b, c = '); readln(a, b, c); a1:=a; b1:=b; c1:=c;	writeln('Цикл по a'); while a1<>0 do begin z:=a1 mod 10; a1:=a1 div 10; if (z=1) then inc(k_a); end; writeln(' кол. ед. = ', k_a); writeln('Цикл по b'); while b1<>0 do begin z:=b1 mod 10; b1:=b1 div 10; if (z=1) then inc(k_b); end; writeln(' кол. ед.= ', k_b); writeln('Цикл по c'); while c1<>0 do begin z:=c1 mod 10; c1:= c1 div 10; if (z=1) then inc(k_c); end; writeln(' кол. ед. = ', k_c); {Нахождение max из 3-х} writeln ('max кол-во ед. в числе '); if (k_a>k_b) and (k_a>k_c) then writeln(a, ' - ', k_a) else if (k_b>k_c) then writeln(b, ' - ', k_b)	writeln('Цикл по a'); repeat z:=a1 mod 10; a1:= a1 div 10; if (z=1) then inc(k_a); until a1=0; writeln('кол. ед. = ', k_a); writeln('Цикл по b'); repeat z:=b1 mod 10; b1:=b1 div 10; if (z=1) then inc(k_b); until b1=0; writeln(' кол. ед.= ', k_b); writeln(' Цикл по c'); repeat z:=c1 mod 10; c1:= c1 div 10; if (z=1) then inc(k_c); until c1=0; writeln(' кол. ед.= ', k_c); {Нахождение max из 3-х} writeln (' max кол-во ед. в числе '); if (k_a>k_b) and (k_a>k_c) then writeln(a, ' - ', k_a) else if (k_b>k_c) then writeln(b, ' - ', k_b)

Окончание таблицы

Начало программы	while	repeat
	<pre>else if (k_b>k_c) then writeln(b, ' - ', k_b) else writeln(c, ' - ', k_c) end.</pre>	<pre>else writeln(c, ' - ', k_c) end.</pre>

Задача 4.4. Алгоритм. Необходимо перевернуть введенное число и сравнить его с оригиналом. Алгоритм перевертывания: $123 \rightarrow 3 \cdot 100 + 2 \cdot 10 + 1$.

Начало программы	while	repeat
<pre>{p4_4} var a, b, a1: longint; begin write('a:='); readln(a); a1:=a;</pre>	<pre>while a1<>0 do begin b:=b*10 + a1 mod 10; a1:=a1 div 10; end; {Сравнение} if (a=b) then writeln('yes') else writeln(' no '); end.</pre>	<pre>repeat b:=b*10 + a1 mod 10; a1:=a1 div 10; until a1=0; {Сравнение} if (a=b) then writeln('yes') else writeln(' no '); end.</pre>

Задача 4.5. Алгоритм. Применить алгоритм перевода числа из десятичной системы счисления в двоичную. В цикле делим число на два до тех пор, пока оно не станет равным нулю, при этом подсчитываем количество остатков, равных единице и нулю.

Начало программы	while	repeat
<pre>{p4_5} var a, k1, k0: integer; begin write('a:='); readln(a); k1:=0; k0:=0;</pre>	<pre>while a<>0 do begin if a mod 2 =0 then inc(k0) else inc(k1); a:=a div 2; end; writeln('k1= ', k1, 'k0=', k0); end.</pre>	<pre>repeat if a mod 2 = 0 then inc(k0) else inc(k1); a:=a div 2; until a=0; writeln('k1= ', k1, 'k0=', k0); end.</pre>

Задача 4.6. Алгоритм. Применить алгоритм суммирования, рекурсию и алгоритм задачи 4.2.

Начало программы	while	repeat	for
<pre>{p4_6} var n, x, i: integer; t, s: real; begin write('x, n='); readln(x, n); t:=x; s:=0;</pre>	<pre>while i<=n do begin inc(i); t:=t/i; s:=s+t; end; writeln('s=', s:3:5); end.</pre>	<pre>repeat t:=t/i; s:=s+t; inc(i); until i>=n; writeln('s=', s:3:5); end.</pre>	<pre>for i:=1 to n do begin t:=t/i; s:=s+t; end; writeln('s= ', s:5:2); end.</pre>

Задача 4.7. Алгоритм. Вводим первое число в *max1*. Продолжаем ввод данных в цикле до тех пор, пока вводимое число станет равным 0. Если вводимое число $a > max1$, то заменяем *max2* на *max1*, а *max1* присваиваем *a*. **Особенность:** вводимое число может быть меньше *max1*, но больше *max2*, поэтому нужно заменять *max2* на это число.

Начало программы	while	repeat
<pre>{p4_7} var a, max1, max2: integer; begin write('max1, a='); readln(max1, a);</pre>	<pre>while a<>0 do begin if (a>max1) then begin max2:=max1; max1:=a; end else if a>max2 then max2:=a; write('a='); readln(a); end; writeln (max1*max2); end.</pre>	<pre>repeat if (a>max1) then begin max2:= max1; max1:=a; end else if a>max2 then max2:=a; write('a='); readln(a); until a=0; writeln (max1*max2); end.</pre>

46. Задачи для подготовки к ОГЭ и ЕГЭ.

Задача 4.8. Алгоритм. Использовать алгоритм задачи 4.7. Наибольшее произведение могут дать как два положительных числа, так и два отрицательных. В последовательности вводимых чисел находим два максимальных и два отрицательных минимальных числа и определяем наибольшее произведение.

Начало программы	while	repeat
<pre>{p4_8} var a, max1, max2, min1, min2 : integer; begin write('max1, a='); readln(max1, a); min1:=max1;</pre>	<pre>while a<>0 do begin if (a>max1) then begin max2:= max1; max1:=a; end else if a>max2 then max2:=a; if (a<min1) and (a<0) then begin min2:=min1; min1:=a; end else if a<min2 then min2:=a; write('a='); readln(a); end; writeln('max произв. '); if (max1*max2)> (min1*min2) then writeln (max1*max2) else writeln (min1*min2) end.</pre>	<pre>repeat if (a>max1) then begin max2:=max1; max1:=a; end else if a>max2 then max2:=a; if (a<min1) and (a<0) then begin min2:= min1; min1:=a; end else if a<min2 then min2:=a; write('a='); readln(a); until a=0; writeln('max ???'); if (max1*max2)>(min1*min2) then writeln(max1*max2) else writeln(min1*min2) end.</pre>

Задача 4.9. Алгоритм. Делим число до тех пор, пока оно не станет равным нулю. Если цифра числа не равна 5, оставляем ее. Используем позиционность десятичной системы счисления. *Особенность:* оставшиеся цифры должны сохранять последовательность. $k := k * 10$ отвечает за сдвиг числа влево каждый раз на разряд, $a - z * k$ — цифра z в своем разряде.

Пример. Дано число 1253, получим новое $n = 1 + 2 \cdot 10 + 1 \cdot 100 = 123$.

Начало программы	while	repeat
<pre>{p4_9} var a, k, n: longint; z: integer; begin n:=0; k:=1; write('a='); readln(a);</pre>	<pre>while a<>0 do begin z:=a mod 10; if (z<>5) then begin {n:=n+z*k;} inc(n, z*k); k:=k*10; end; a:=a div 10; end; writeln('n=', n); end.</pre>	<pre>repeat z:=a mod 10; if (z<>5) then begin {n:=n+z*k;} inc(n, z*k); k:=k*10; end; a:=a div 10; until a=0; writeln('n=', n); end.</pre>

Задача 4.10. Алгоритм. Находим произведение двух чисел — предыдущего и текущего. Если произведение чисел больше 50, опять переходим к вводу нового числа до тех пор, пока оно будет меньше 50. Также считаем количество введенных чисел, чтобы определить порядковые номера двух соседних чисел, дающих такой результат.

Начало программы	while	repeat
<pre>{p4_10} var a, k, s: integer; begin</pre>	<pre>write('a='); readln(s); write('a='); readln(a); k:=2; s:=s*a; while s>50 do begin s:=a; write('a='); readln(a); s:=s*a; inc(k); end; writeln(k-1, ' ', k); end.</pre>	<pre>write('a='); readln(a); k:=1; repeat s:=a; write('a='); readln(a); s:=s*a; inc(k); until s<50; writeln(k-1, ' ', k); end.</pre>

5. Задачи повышенной сложности

Задача 5.1. Алгоритм. Возведение в степень — это сокращенная форма умножения. Умножаем число само на себя до тех пор, пока результат не станет больше заданного. Количество умножений даст число k .

Начало программы	while	repeat
<pre>{p5_1} var a, b, k: integer; p: longint; begin write('a, b'); readln(a, b);</pre>	<pre>p:=a; k:=0; while (p<b) do begin inc(k); p:=p*a; end; writeln('k:= ',k:5); end.</pre>	<pre>repeat inc(k); p:=p*a; until p>b; writeln('k:= ',k:5); end.</pre>

Задача 5.2. Алгоритм. Использовать алгоритм задачи 4.2.

Начало программы	while	do...while
<pre>{p5_2} var n, k, k_rez: integer; begin k:=0; k_rez:=1; write('n'); readln(n);</pre>	<pre>while (k_rez<n) do begin inc(k); k_rez:=k_rez*k; end; writeln('k= ', k:3, ' k_rez = ', k_rez:5); end.</pre>	<pre>repeat inc(k); k_rez:=k_rez*k; until k_rez>=n; writeln('k= ', k:3, ' k!=', k_ rez:5); end.</pre>

Задача 5.3. Алгоритм. Организовать счетчик по количеству введенных положительных чисел и количеству введенных чисел. Как только встретится третье положительное число или число, равное нулю, цикл прекратит работу.

Начало программы	while	repeat
<pre>{p5_3} var a, x, na, k, i: integer; begin x:=1; k:=0; i:=0;</pre>	<pre>while (k<>3) and (x<>0) do begin write('x ='); readln(x); inc(i); if (x>0) then begin na:=i; a:=x; inc(k); end; end; if (k = 3) then writeln (a:5, ' ', na:5) else writeln('no'); end.</pre>	<pre>repeat write('x='); readln(x); inc(i); if (x>0) then begin na:=i; a:=x; inc(k); end; until (k=3) or (x=0); if (k=3) then writeln(a:5, ' ', na:5) else writeln('no'); end.</pre>

Задача 5.4. Алгоритм. Используя операции целочисленного деления, найти сумму цифр числа s , проверить на кратность трем ($s \bmod 3 = 0$).

Начало программы	while	repeat
<pre>{p5_4} var n, s; integer; begin s:=0; write('n ='); readln(n);</pre>	<pre>while n<>0 do begin inc(s,n mod 10); n:=n div 10; end; if(s mod 3 =0) then writeln('Yes') else writeln('No'); end.</pre>	<pre>repeat inc(s,n mod 10); n:=n div 10; until n = 0; if(s mod 3 =0) then writeln('Yes') else writeln('No'); end.</pre>

Задача 5.5. Алгоритм. Организовать вложенные циклы. Во внутреннем цикле из числа удалить все четные цифры (алгоритм задачи 4.9), во внешнем — подсчитать сумму чисел.

Начало программы	while	repeat
<pre>{p5_5} var z, s: integer; a, n, k, x: longint; begin s:=0; x:=1;</pre>	<pre>while x<>0 do begin write('x='); readln(x); a:=x; k:=1; n:=0; while a<>0 do begin z:=a mod 10; if (z mod 2 =1) then begin inc(n, z*k); k:=k*10; end; a:=a div 10; end; if(n<>0) then s:=s+n; end; writeln('s:= ', s:5); end.</pre>	<pre>repeat write('x = '); readln(x); a:=x; k:=1; n:=0; repeat z:=a mod 10; if (z mod 2 =1) then begin inc(n, z*k); k:=k*10; end; a:=a div 10; until a = 0; if (n<>0) then s:=s+n; until x=0; writeln('s:= ', s:5); end.</pre>

Задача 5.6. Алгоритм. Ввести до начала цикла первое число. В цикле проверять вводимое число на кратность первому введенному числу, на положительность; подсчитать количество таких чисел.

Начало программы	while	repeat
<pre>{p5_6} var n, a, b, s: integer; begin n:=0; s:=0; write('a='); readln(a); b:=a;</pre>	<pre>while a<>0 do begin write('a='); readln(a); if ((a>0) and (a mod b = 0)) then begin inc(s, a); inc(n); end; end; if (s>0) then writeln(s/n:3:2) else writeln('No'); end.</pre>	<pre>repeat write('a='); readln(a); if ((a>0) and (a mod b = 0)) then begin inc(s, a); inc(n); end; until a=0; if (s>0) then writeln(s/n:3:2) else writeln('No'); end.</pre>

Вопросы для обсуждения:

- В чем отличие цикла while от цикла repeat? (Цикл repeat выполнится хотя бы один раз, и проверка условий осуществляется в конце цикла. В цикле while проверка выполняется в начале цикла.)
- Какой цикл удобнее использовать для проверки корректности вводимых данных? (repeat.)
- Когда тело цикла заключается в операторные скобки? (При наличии в теле цикла более одного оператора тело цикла должно быть заключено в операторные скобки begin ... end.)
- В чем особенность применения цикла for? (Параметр цикла for может изменяться (увеличиваться или уменьшаться) каждый раз при выполнении тела цикла только на единицу. Если нужен другой шаг изменения параметра, предпочтительнее циклы repeat или while.)
- В чем особенности применения оператора цикла while? (Значения всех переменных, входящих в условие, должны быть определены до выполнения оператора while, в теле цикла обязательно должна быть переменная, влияющая на изменения условия.)
- В чем особенности применения оператора цикла repeat? (Он выполнится хотя бы один раз, в теле цикла обязательно должна быть переменная, влияющая на изменения условия.)
- Как вы понимаете пословицу «Аз да буки, а потом науки»?
- Приведите аналоги пословиц. (Без труда не вынешь рыбку из пруда. Чтобы научиться бегать, нужно сначала научиться ходить. Ремеслу учиться — всегда пригодится.)

Рейтинговая таблица подведения итогов занятия

№ п/п	Фамилия, имя	Задания								Всего
		1.1	1.2	1.3	2	4а	4б	5	Доп. баллы	
1	А. Кирилл		1		50	7	2	3		63
2	Б. Владимир				47	6	3	3		59
3	В. Сергей				47	7	3	7	1	65
4	Г. Геннадий	1		1	50	7	6	15	3	83
5	К. Александр				46	5	6	2		59
6	К. Павел				47	6	3	2		58
7	К. Максим				45	7	6	8	2	68
8	Л. Вадим				46	7	5	10		68
9	О. Егор		1		50	7	6	6	3	73
10	С. Анна				50	7	6	12	2	77
11	С. Сергей	1	1	1	50	7	6	18	6	90
12	Т. Евгений				47	7	6	9		69
13	Ю. Илья	1	1	1	50	7	6	17	8	91
	Средний балл:				48,1	6,7	4,9			

Задания 1.1–1.3 оцениваются педагогом.

Задания 2 оцениваются обучающимся (самоконтроль), общий результат передается педагогу. За каждый правильный ответ — 1 балл. Максимальное количество баллов — 50.

Задания 4а (для обязательного решения) оцениваются в 1 балл. Максимальное количество баллов — 7.

Задания 4б (для подготовки к ОГЭ и ЕГЭ) оцениваются в 1–2 балла. Максимальное количество баллов — 6.

Задания 5 оцениваются в 1–3 балла. Максимальное количество баллов — 18.

Педагог может выставлять дополнительные баллы.

*Приложение 4***Перечень возможных ошибок**

- Синтаксические ошибки.
- Ошибки в вычислениях.
- Ошибки в обозначении операций отношения и логических операций.
- Операции отношения и логические операции в условных операторах и циклах должны быть заключены в скобки ().
- Скобки begin ... end должны быть парными.
- В процессе выполнения программы происходит обновление данных.

Рекомендации по использованию циклов

- Если в теле цикла больше одного оператора, тело цикла заключается в скобки begin ... end.
- В условных циклах обязательно должен быть оператор, влияющий на изменение условий.
- В условных циклах переменные, входящие в условия, должны быть определены до цикла.
- Цикл с постусловием repeat рекомендуется применять для проверки правильности ввода данных

или когда цикл должен быть выполнен хотя бы один раз.

- Оператор цикла с предусловием while удобнее использовать в случаях, когда число итераций заранее неизвестно.
- Оператор for предпочтительнее в большинстве остальных случаев (однозначно — для организации циклов со счетчиками).

*Приложение 5***Профилактика переутомления**

1. Общеукрепляющие, активизирующие упражнения необходимо проводить в начале занятия.

«Стульчик». И. п. — сидя. Глаза закрыты, ладони обхватывают край стула. По команде педагога «Начали упражнение» ребенок начинает тянуть на себя стул, своим телом осуществляя противоположное движение. Такое положение удерживается около 15 секунд (ведущий мысленно считает до 15). По команде «Закончили упражнение» ребята расслабляются. Повторить 3–5 раз.

«Дерево». И. п. — сидя или стоя. Ноги вместе, стопы прижаты к полу, руки опущены, спина прямая. Сделали спокойные вдох и выдох, плавно подняли руки вверх ладонями друг к другу. Потянулись всем телом. Внимание на позвоночник. Дыхание свободное. Представьте себя деревом. Оно корнями глубоко вросло в землю. Высокий стройный ствол тянется к солнцу. Ваш организм, как дерево, наливается силой, бодростью, здоровьем. Удержите позу 15–20 секунд. Затем руки плавно опустите и расслабьтесь.

2. Упражнения, направленные на профилактику глазных болезней, выполняются через 15–20 минут после начала занятия.

Комплекс упражнений для глаз (желательно проделывать этот комплекс перед серьезной зрительной работой и после нее).

- «Жмурки». Крепко зажмурить глаза на 3–5 секунд, затем открыть их. Повторить 6–8 раз. Упражнение укрепляет мышцы век, расслабляет мышцы глаз и улучшает их кровообращение.

- Быстро моргать в течение 15 секунд. Повторить 3–4 раза. Упражнение улучшает кровоснабжение глаз.
- Закрывать глаза и массировать веки круговым движением пальца в течение 1 минуты. Упражнение расслабляет мышцы глаз, улучшает их кровоснабжение.

Пальминг. Закрывать оба глаза и прикрыть их ладонями. Дождаться появления «черного поля». Глаза держать расслабленными 1–2 минуты.

3. Упражнения, направленные на профилактику нарушения осанки и снятия напряжения, рекомендуется проводить через 30 минут после начала занятия, когда наступает утомление мышц спины.

«Буратино». Лепим красивый нос для Буратино. Буратино рисует носом «солнышко», «морковку», «цифры» и т. д. При этом у детей появляется радостное ощущение от медленных и плавных движений шеи. Упражнение дает терапевтический эффект для шейного отдела позвоночника.

НОВОСТИ

Кибербезопасность предлагают ввести в число основных школьных предметов

В Совете Федерации предлагают включить в число основных школьных предметов информационную безопасность, сообщает Агентство городских новостей «Москва» со ссылкой на заместителя председателя комитета верхней палаты парламента России по конституционному законодательству и государственному строительству Людмилу Бокову.

«Необходимость введения такого урока, в принципе, допустима. Мы рассматривали такое предложение, но не как отдельный курс, а как комплексное интегрирование контента в школьную программу. Мы посмотрели и выяснили, что на уроках информатики, ОБЖ, географии, биологии помимо основных тем предмета можно говорить и об информационной безопасности», — сказала Бокова.

Сенатор отметила, что, например, на уроках биологии можно рассказывать о типах вирусов, которые бывают инфекционные и компьютерные.

«На других уроках можно рассказать, как подготовить презентацию, которая требует представления своих персональных данных. Нужно сказать ребенку о том, что фотографии, использованные в презентации, являются биометрическими данными», — сообщила Бокова.

Она уточнила, что сенаторы уже подготовили соответствующее пособие, в котором представлены 270 уроков практически по всем предметам школьной программы.

«Сейчас это пособие проходит некую общественную экспертизу. Я думаю, что любая школа сочтет за необходимость для себя ввести ее в школьную программу», — отметила Бокова.

По ее словам, в настоящее время материалами Единого урока безопасности поведения в Интернете заинтересовались около десяти стран СНГ и Вьетнам.

Единый урок по безопасности в Интернете — это цикл мероприятий для школьников, который направлен на повышение уровня кибербезопасности, привлечение внимания родителей и педагогов к проблеме детской безопасности в сети Интернет и развитие цифровой грамотности у школьников.

Урок безопасности проходит при поддержке Министерства образования и науки РФ, Министерства связи и массовых коммуникаций РФ, Института Развития Интернета, федеральных и региональных органов власти, а также представителей ИТ-отрасли.

Единый урок впервые прошел в 2014 году по предложению спикера Совета Федерации Валентины Матвиенко.

Разработана первая электронная карта археологических памятников России

Археологи разработали первую электронную карту археологических памятников России, сообщает Агентство городских новостей «Москва» со ссылкой на пресс-службу Института археологии Российской академии наук.

Геоинформационная система «Археологические памятники России» содержит данные более чем о 15 тысячах объектов.

Пресс-служба уточняет, что основу для карты в 2014 году начала создавать группа под руководством директора института Николая Макарова.

Был разработан стандарт описания археологического памятника (в него входят название объекта, тип, хронологи-

ческая атрибуция, принадлежность к речному/морскому бассейну, особенности микрорельефа, географические координаты и сведения об исследовании).

Также для карты при поддержке Российского научного фонда создали программную оболочку — автоматизированную систему обработки информации.

«На сегодня в базу данных, составляющую основу археологической карты, загружена информация о памятниках, исследованных с 2009 по 2012 годы — это сведения о 15 тысячах 367 объектах археологического наследия, а также о 11 тысячах 230 “пустых” шурфах, где не было найдено культурных напластований», — говорится в сообщении.

(По материалам федерального портала «Российское образование»)