

Л.В. Батан

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ В СРЕДНЕЙ ШКОЛЕ

Могилев 2012

Электронный архив библиотеки МГУ имени А.А. Кулешова

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования
«МОГИЛЕВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени А.А. КУЛЕШОВА»

Л.В. Батан

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ В СРЕДНЕЙ ШКОЛЕ

Методические материалы



Могилев 2012



БІБЛІЯТЭКА
Магілёўскага
дзяржаўнага
універсітэта
імя А. А. Куляшова

УДК 372.8:004

ББК 74.263.2

Б28

*Печатается по решению редакционно-издательского совета
УО «МГУ им. А.А. Кулешова»*

Рецензент

кандидат физико-математических наук, доцент кафедры
информатики и вычислительной техники
УО «Могилевский государственный университет продовольствия»
В.Л. Титов

Батан, Л.В.

Б28 Основы алгоритмизации и программирования в средней школе : метод. мат-лы / Л.В. Батан. – Могилев: УО «МГУ им. А.А. Кулешова», 2012. – 84 с. : ил.

Одним из фундаментальных понятий в информатике является понятие алгоритма. При этом наибольшие затруднения школьники испытывают при освоении тем раздела «Основы алгоритмизации и программирования». Предлагаемые материалы к урокам по теме «Основы алгоритмизации и программирования» помогут студентам эффективно подготовиться к лабораторным занятиям по методике информатики, урокам информатики при прохождении активной педагогической практике в школе. Данные учебно-методические разработки предназначены для студентов физико-математического факультета, могут быть рекомендованы учителям информатики средней общеобразовательной школы.

УДК 372.8:004

ББК 74.263.2

© Батан Л.В., 2012

© УО «МГУ им. А.А. Кулешова», 2012

ОТ АВТОРА

Одним из фундаментальных понятий в информатике является понятие алгоритма. Под алгоритмом понимают конечную последовательность точных предписаний (команд), формальное выполнение которых позволяет получить решение поставленной задачи.

Вашему вниманию предлагаются учебно-методические разработки по обучению информатике при изучении темы «Основы алгоритмизации и программирования». Предлагаемые методические материалы учитывают преемственные связи по теме, охватывая несколько аспектов (преемственных линий).

- 1) Линия алгоритма – уточняется и закрепляется примерами понятие алгоритма, введенное в 6 классе. Понятие алгоритма неразрывно связано с понятием исполнителя. Акцентируется внимание учеников на том, что алгоритм составляется для конкретного исполнителя.
- 2) Линия исполнителя – закрепляется понятие исполнителя (Чертежник), введенное в 6 классе. Исполнителем программ является компьютер.
- 3) Виды алгоритмов. Закрепляются знания учеников о видах алгоритмов, полученные в 6 классе. Закрепление данного материала представляется нам целесообразным, поскольку эта линия будет интенсивно «отрабатываться» в 8 классе, когда будут изучены основные алгоритмические конструкции языка Паскаль.
- 4) Способы записи алгоритмов. Закрепляются знания учеников о способах записи алгоритмов, полученные в 6 классе. Наиболее наглядным для простых программ на начальном этапе обучения алгоритмизации является графический способ записи алгоритма (блок-схема). Блок-схемы используются в 8 классе при рассмотрении основных алгоритмических конструкций
- 5) Закрепление понятия константа на примерах из учебника. Строковые и числовые константы. Знание констант «в лицо» поможет ученикам лучше понимать материал учебника. Особенно это касается строковых констант, которые часто используются при написании программ. Понятие о строковых константах подготовит учеников к восприятию материала по обработке строковых величин в 10 классе.
- 6) Использование переменных при работе с графикой – преемственность и закрепление материала предыдущих параграфов учебника 7 класса.

Целью изучения темы «Основы алгоритмизации и программирования» является развитие логического и алгоритмического стиля мышления. «Основу

алгоритмического стиля мышления составляет умение планировать длинные последовательности действий, ясно и четко записывать их в виде программ, умение предвидеть их последствия и предусматривать все условия, которые могут возникнуть при выполнении программы» [1]. Составление только линейных алгоритмов лишь в незначительной степени может этому способствовать.

С другой стороны, в дидактике имеется понятие «опережающее обучение», введенное С.Н. Лысенковой. Суть его состоит во включении дополнительных материалов, назначение которых: а) обеспечить преемственность между годами обучения (в программе 1-го класса есть материал 2-го и даже 3-го и т.д.); б) предоставить возможность детям, способным к раннему усвоению учебного материала, реализовать свою познавательную потребность; в) создать для менее способных или менее подготовленных детей реальные условия усваивать материал без спешки и напряжения, вводя его малыми дозами за гораздо более длительный период, чем отведено по программе.

Учитывая сложность материала 8 класса по теме «Основы алгоритмизации и программирования», считаем целесообразным включать элементы опережающего обучения в материал 7 класса. Можно рекомендовать использовать алгоритмы с циклами и ветвлением там, где это гармонично (без нарушения преемственных связей) вписывается в изучаемый материал. Примерами таких алгоритмов могут служить алгоритм рисования пунктирной линии и алгоритм определения четности/нечетности вводимого числа. Методические разработки к некоторым урокам содержат задания со звездочкой (*), предназначенные для учеников, способным к раннему усвоению учебного материала. Такие задания можно заготовить заранее на карточках. Упомянутые задания «со звездочкой» не являются обязательными для выполнения.

УРОКИ ПО ТЕМЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ» В 7 КЛАССЕ

Урок 11. СРЕДА ПРОГРАММИРОВАНИЯ

Цели урока:

- формирование представлений о языке программирования;
- закрепление умений загружать и сохранять программы;
- расширение представлений о структуре программы;
- развитие познавательного интереса, умений выделять главное.

Тип урока: усвоение новых знаний и умений.

Учащиеся должны знать структуру программы.

Учащиеся должны уметь запускать среду программирования Pascal ABC, загружать, выполнять и сохранять программы в среде Pascal ABC.

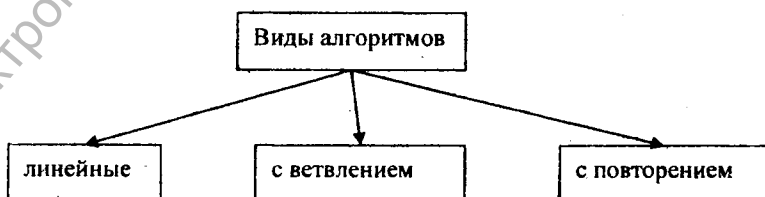
Программное и методическое обеспечение урока: система программирования Pascal ABC, учебное пособие «Информатика, 7» §9; раздаточный материал (*приложение к уроку 11*).

Методические рекомендации к уроку

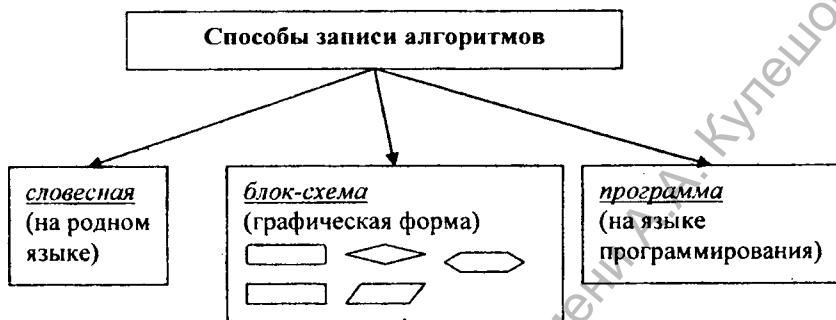
1. **Актуализация знаний и мотивация учащихся на изучение учебного материала.** В ходе фронтальной беседы повторите с учащимися понятие *алгоритм*. Обсудите наиболее важные вопросы темы алгоритмизация.

Алгоритм – это *описание* последовательности действий, направленных на решение поставленной задачи.

Вспомните с учениками виды алгоритмов:



Вспомните с учениками формы (способы) записи алгоритмов:



Подчеркните, что программа – это алгоритм, записанный на языке программирования. Исполнителем такого алгоритма является компьютер.

Далее расскажите об истории создания языка Паскаль (Pascal). Язык создан швейцарским ученым, преподавателем Станфордского университета Николаусом Виртом в начале 1970-х годов специально для обучения программированию (студентов). Язык назван в честь французского математика и физика Блеза Паскаля, создавшего первое механическое вычислительное устройство. Паскаль создавался как язык, который, с одной стороны, был бы хорошо приспособлен для обучения программированию, а с другой – давал бы возможность эффективно решать самые разнообразные задачи на компьютере. Паскаль стал прародителем нескольких языков программирования, например: Modula-2, Oberon.

Расскажите учащимся, что в языке программирования имеются специальные слова, смысл и способ употребления которых задан раз и навсегда. Такие слова называют *зарезервированными* или *ключевыми* (иногда *служебными*). Примеры ключевых слов: **program**, **begin**, **end**. Приступая к разработке программ, необходимо ознакомиться с ключевыми словами (см. приложение в учебнике, с. 150, таблица 1). Для правильной разработки программ необходимо также знать правила построения команд – *синтаксические правила*.

2. Объяснение нового материала. Напомните учащимся, что предложения, из которых строится программа, набираются на клавиатуре. Для того чтобы исполнитель-компьютер смог выполнить записанную нами программу, ее нужно перевести на понятный компьютеру язык – машинный язык. Такой перевод осуществляет специальный переводчик – *компилятор*, встроенный в систему программирования. Например, в систему программирования Pascal ABC встроен компилятор, который позволяет программировать на языке Pascal.

Вспомните с учащимися структуру программы, с которой они знакомились в 6-м классе:

Program <имя программы>;
<раздел описаний, в котором описываются данные>;
Begin
<раздел команд (тело программы)>;
End

Обратите внимание на правила представления текста программы и на алгоритм работы с программой в среде Pascal ABC, с которым они познакомились в 6-м классе (см. приложение 1 к уроку 11).

Обсудите с учениками способ запуска системы Pascal ABC, предложите ее запустить. В окне редактора набрать текст программы примера 1 (§9, и. 9.2), сохранить набранную программу, а затем ее выполнить.

Задайте следующие вопросы ученикам:

1. Для чего нужно писать комментарии в программе? (для разъяснения отдельных предложений, (фрагментов) программы)
2. Вопрос на внимательность: Какой комментарий отсутствует в программе text1 (примере 1, с. 55)? ({раздел команд})
3. Сколько команд записано в теле программы text1? (одна)
4. Кто является исполнителем программы text1? (компьютер)
5. Что является результатом выполнения программы text1? (текст: Я всемогущий компьютер! Жду ваших команд)
6. Что происходит при выполнении программы, если в ее тексте есть ошибка? (курсор в окне редактора устанавливается в области ошибки, а в окне вывода выдается сообщение об ошибке)

Ошибку необходимо найти в тексте программы и исправить. После чего снова попытаться выполнить программу. Такие ошибки называют *синтаксическими* – они связаны с правильностью записи предложений (команд) программы.

Сообщите о возможности возникновения логических ошибок в программе (например, использована не та команда). Сообщите о том, что в языках программирования не допускаются даже незначительные (с точки зрения человека) нарушения в правилах написания программ; что даже маленькая неточность может вызвать невыполнимость программы или трудноуловимую ошибку.

3. Закрепление нового материала.

Вопрос ученикам: какие изменения нужно внести в программу text1, чтобы она выводила на экран дату Дня Республики Беларусь? (заменить текст в апострофах на 3 июля)

Предложите учащимся самостоятельно выполнить упражнения 1, 2 из § 9.

4. Подведение итогов урока.

5. Домашнее задание. Изучить материал § 9, записи в тетради, ответить на вопросы 1 – 4.

Приложения к уроку 11

Приложение 1. Правила представления текста программы

1. Имя программы может содержать произвольное количество латинских букв, цифр, знаков подчеркивания и должно начинаться с буквы или знака подчеркивания «_».
2. Нельзя присваивать программам либо данным (величинам) имена зарезервированных в языке программирования слов (например, Begin, End) и стандартных команд (например, Write).
3. Предложения в программе отделяются друг от друга точкой с запятой «;».
4. При вводе текста программы с клавиатуры *не имеет значения, какие буквы вы используете: строчные либо прописные.*
5. В фигурных скобках можно писать комментарии (пояснительный текст, который облегчает понимание программы и не влияет на ее выполнение). Использование в программе комментариев желательно, но необязательно.

Приложение 2. Алгоритм работы с программой в среде программирования Pascal ABC

1. Запустить систему программирования Pascal ABC.
2. Записать подготовленный текст программы в окне редактора.
3. При редактировании текста использовать такие же приемы, как при работе в текстовом редакторе.
4. Выполнить программу с помощью команды **Программа** → **Выполнить (F9)**.
5. При обнаружении ошибок курсор устанавливается на ошибку в окне редактирования и выдается сообщение об ошибке в окне вывода.
6. Если компилятор не обнаружил ошибки, вы увидите результат работы вашей программы в окне вывода.

Урок 12. ОСНОВЫ ЯЗЫКА ПРОГРАММИРОВАНИЯ

Цели урока:

- закрепление материала предыдущего урока;
- формирование представлений о переменной и правилах ее записи, о числовых типах данных;
- развитие умений выделять главное.

Тип урока: комбинированный.

Учащиеся должны знать понятие переменной и ее типа.

Учащиеся должны уметь описывать переменные.

Программное и методическое обеспечение урока: система программирования Pascal ABC; учебное пособие «Информатика. 7», § 10; раздаточный материал (*приложения 1–3 к уроку 12*).

Методические рекомендации к уроку

1. Проверка знаний и закрепление умений предыдущего урока. Предложите учащимся проверить степень усвоения материала предыдущего урока с помощью теста (приложение 1 к уроку 12) с последующей взаимопроверкой при фронтальном обсуждении правильных ответов с целью коррекции знаний.

2. Объяснение нового материала (объяснительно-иллюстративный метод обучения). Напомните учащимся, что человек в своей деятельности имеет дело с различного рода информацией. Информацию, обрабатываемую компьютером или полученную в качестве промежуточных либо итоговых результатов, называют данными.

Введите понятия *величина, переменная, константа* (см. приложение 2 к уроку 12). Обратите внимание учеников на то, что они уже имели дело со строковыми константами. Так, например, текст “Я всемогущий компьютер! Жду ваших команд” (программа `text1` из примера 1 §9) – это строковая константа.

Обратите внимание учащихся, что в программировании переменные играют важную роль, так как они предназначены для хранения данных. Далее расскажите о том, что конкретное значение переменной хранится в ячейках оперативной памяти. Каждая ячейка имеет свой адрес – порядковый номер.

Примеры адресов:

`0F000H, 0C000H, 0EFFFH, 3FFFH, 7000H`

Очевидно, что с адресами ячеек в таком виде работать неудобно, поэтому каждой переменной в программе дается имя.

Далее познакомьте учащихся с правилами записи имени переменной (*приложение 2 к уроку 12*). Обратите внимание учащихся на то, что имя переменной (как и имя программы) всегда должно начинаться с латинской буквы (либо знака подчеркивания), после чего могут следовать несколько латинских букв, цифры либо символ подчеркивания. В имени не может быть русских букв и пробелов.

В одной программе не может быть двух переменных с одинаковыми именами, также имя переменной не должно совпадать с именем программы.

Для того чтобы можно было использовать переменную в программе, ее нужно описать. Описать все переменные нужно до их использования, в разделе описания. Описания всех переменных, которые будут задействованы в программе, производят после служебного слова `var` (после строки заголовка программы – см. *структуру программы на с. 55 учебника*). При описании переменной указывают тип этой переменной. Тип определяет диапазон значений и набор

допустимых операций. Тип также определяет объем памяти, необходимый для хранения значения. Познакомьте учащихся с типами данных **Integer** и **Real**. Приведите примеры описания переменных (*приложение 2 к уроку 12*).

Обратите внимание учащихся на то, что если переменная в программе не определена, то компьютер выдаст соответствующее сообщение об ошибке (*приложение к учебному пособию, с. 152, таблица 4*), при этом программа не выполнится.

3. Закрепление нового материала (репродуктивный метод обучения, индивидуальная и фронтальная формы работы). Предложите учащимся в форме фронтальной беседы ответить на вопросы упражнений 1–4 из § 10.

Далее проведите индивидуальную коррекцию усвоения полученных знаний с помощью теста (*приложение 3 к уроку 12*), который соответствует целям урока. Предложите учащимся сверить результаты теста друг у друга и сравнить с эталоном. Отметку за тест выставляйте только по желанию учащегося.

4. Подведение итогов урока. Подведите итоги индивидуальной работы, дайте оценку успешности достижения цели на уроке и рекомендации по повышению уровня знаний по рассматриваемой теме. Скорректируйте неверные представления.

Если останется время, предложите учащимся обобщить изученное на уроке, выделяя главное.

- данные в программировании принято называть **величинами**.
- величины, которые могут изменять в процессе работы программы, называют **переменными**, а те, которые не изменяются, – **константами**.
- каждая переменная, кроме имени и значения, имеет **тип**. например, **integer** – целый тип; **real** – вещественный тип.
- все переменные, используемые в программе, должны быть описаны в разделе описаний. **описать переменную** – это значит указать ее имя и тип после зарезервированного слова **Var**.

5. Домашнее задание. Изучить материал § 10, ответить на вопросы 1–4.

Приложения к уроку 12

Приложение 1. Тест по теме «Среда программирования. Структура программы» (входной контроль)

Выберите правильный ответ.

1. Алгоритм – это:

- а) набор команд;
- б) последовательность действий;
- в) описание последовательности действий;
- г) процесс решения поставленной задачи.

2. Программа — это:

- а) один из способов самовыражений;
- б) один из способов записи алгоритма;
- в) инструкция по разработке алгоритма.

3. Исполнителем программы является:

- а) компьютер
- б) автор учебника
- в) ученик

4. Для загрузки программы из файла на диске нужно выполнить команду:

- а) **Файл** → **Открыть** → выбрать нужный файл;
- б) **Файл** → **Сохранить** → выбрать нужный файл;
- в) **Программа** → **Выполнить** → выбрать нужный файл.

5. Тело любой программы начинается и заканчивается зарезервированными

словами:

- а) начало... конец;
- б) Program...End;
- в) Begin ...End.

6. В программе после каждого предложения нужно ставить:

- а) точку;
- б) запятую;
- в) точку с запятой.

7. В конце программы после слова End ставят:

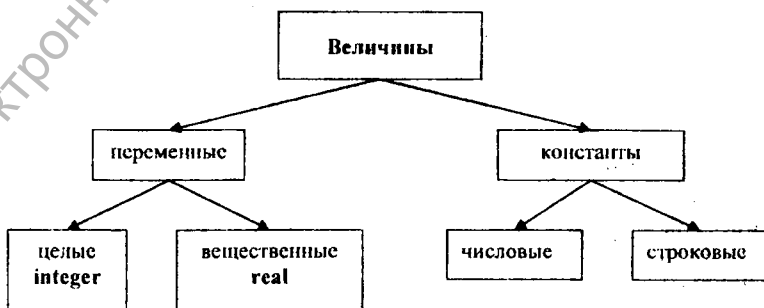
- а) точку;
- б) восклицательный знак;
- в) точку с запятой.

Правильные ответы:

1 – в; 2 – б; 3 – а; 4 – а; 5 – в; 6 – в; 7 – а.

Приложение 2. Величины

Записать на доске, либо на слайде, либо на карточке-инструкции:



Переменные и их описание

Имя переменной

- начинается с буквы или знака подчеркивания
- может состоять из:
 - латинских букв
 - цифр
 - подчеркивания
- не должно содержать точки и пробелы
- количество символов не более 255

Возможные имена переменных:

A ab a1 a12 c10 summa sum2 x1x2 yuyu

Неверные имена переменных:

2b 5sum 2 pr 453 pr 5 (имя с пробелом)

Общий вид описания переменной:

Var <имя_переменной>: <тип>;

Примеры:

Var a : integer;

b : real;

Если *переменные одного и того же типа*, то они записываются *через запятую*:

Var a, a12, b : integer;

x, y : real;

Приложение 3. Тест по итогам урока:

1. Укажите допустимое имя программы:

а) Primer_1

б) 1_primer

в) Primer 1

г) Пример1

2. Укажите допустимые имена переменных:

- а) aB
- б) 2b
- в) r 2
- г) _clO
- д) xlx2

3. Заголовок программы начинается служебным словом:

- а) Program
- б) Var
- в) Begin
- г) Write
- д) End

4. Раздел команд (тело программы) начинается после служебного слова:

- а) Program
- б) Var
- в) Begin
- г) Write
- д) End

5. Раздел описания переменных начинается после служебного слова:

- а) Program
- б) Var
- в) Begin
- г) Write

6. Слово var должно быть записано

- а) один раз перед описанием переменных
- б) каждый раз перед описанием каждой переменной
- в) каждый раз перед описанием переменных разных типов

7. Укажите строки, в которых правильно описаны целочисленные переменные:

- а) Var a, b, c : real;
- б) Var a : integer;
- в) Var d, e : integer;
- г) Var k : real;

8. Укажите строки, в которых правильно описаны вещественные переменные:

- а) Var a, b, c : real;
- б) Var a : integer;
- в) Var d, e : integer;
- г) Var k : real;

9. Какие из следующих величин (числовых констант) относятся к целому типу:

- а) 100 б) 5 в) 134.5 г) 0.345 д) -347 е) -2.5
- ж) 4.0 з) 0

10. Какие из следующих величин (числовых констант) относятся к вещественному типу:

- а) 100 б) 5 в) 134.5 г) 0.345 д) -347 е) -2.5
ж) 4.0 з) 0

Правильные ответы:

1 – а; 2 – а, г, д; 3 – а; 4 – в; 5 – б; 6 – а; 7 – б, в; 8 – а, г; 9 – а, б, д, з; 10 – в, г, е, ж.

Урок 13. ОРГАНИЗАЦИЯ ВВОДА И ВЫВОДА ДАННЫХ

Цели урока:

- закрепление материала предыдущего урока;
- формирование представлений о командах ввода и вывода данных;
- развитие логического и алгоритмического мышления; познавательного интереса, умений выделять главное.

Тип урока: комбинированный.

Учащиеся должны уметь составлять и выполнять программы с использованием команд ввода и вывода данных.

Программное и методическое обеспечение урока: система программирования Pascal ABC; учебное пособие «Информатика, 7», § 11; раздаточный материал (*приложения 1–3 к уроку 13*); ЭСУ Algo7.

Методические рекомендации к уроку

1. Проверка знаний и закрепление умений предыдущего урока.

Разделите учащихся на 3 группы (в каждой группе ученики разных способностей, уровня воспитанности, степени мотивированности). Предложите каждой группе учащихся выполнить свое задание (*приложение 1 к уроку 13*) и представить найденные совместно ответы в форме фронтального обсуждения.

Для представления результата выберите по одному ученику из каждой группы случайным образом. Обратите внимание учащихся на то, что оценка работы каждой группы в общем и каждого участника команды в отдельности будет зависеть от степени усвоения материала каждым, а также от оказания помощи друг другу.

2. Объяснение нового материала.

Сообщите учащимся, что составление алгоритмов решения задач на языке программирования включает в себя этапы определения исходных данных и вывода результатов.

Сообщите, что результатами выполнения программы могут быть сообщения (*строковые константы*), числа либо и то и другое.

Предложите учащимся назвать команду, с помощью которой они выводили на экран сообщение «Моя Родина – Беларусь».

Далее сообщите учащимся, что **Write** – одна из часто используемых команд для организации диалога между пользователем и компьютером, а также для сообщений пользователю о результатах выполнения программы (вывод данных).

Обсудите вместе с учащимися программу `privet_2` из § 11. Сообщите учащимся, что существует вторая разновидность команды **Write** – **WriteLn**. Поясните на примере программы `privet_3` из § 11 отличие команды **Write** от **WriteLn**.

Далее на примере программы `summa` познакомьте учащихся с новыми возможностями оператора **Write**. Обратите внимание учеников на то, что в программе `summa` записана не строковая константа `'2+3'`, а сумма двух числовых констант 2 и 3. Поэтому в результате выполнения программы `summa` на экране появится результат вычисления суммы – число 5.

Обратите внимание учащихся на то, что если записать команду **Write** (`'2+3'`), то результатом выполнения программы будет вывод (заклученной в апострофы) строки `2+3`.

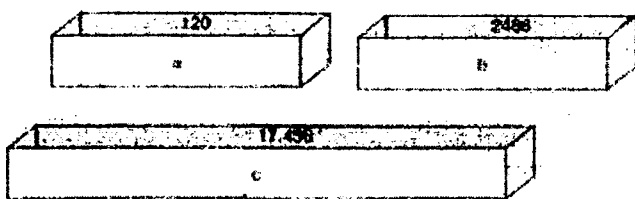
Обобщите вместе с учащимися, что на данном этапе урока они научились следующему: получать сообщения от компьютера; выводить результаты работы программы на экран.

Далее скажите о необходимости научиться отвечать компьютеру на его вопросы или сообщать ему исходные числа для вычислений.

Напомните учащимся, что исходные данные и результаты выполнения программ могут помещаться в переменные.

Предложите учащимся вспомнить, как описывается переменная в программе (следует указать ее имя и тип). Акцентируйте внимание учащихся на том, что переменная – это величина, которая может изменяться в процессе работы программы. Подчеркните, что следует соблюдать следующий принцип: «Использовать переменную можно лишь тогда, когда она описана и ей присвоено некоторое значение». Это позволит избежать возможных ошибок в работе программы.

Покажите с помощью рисунка, что наглядно переменную можно представить как коробочку, в которую можно положить на хранение что-либо. **Имя переменной** – это надпись на коробочке, **значение** – это то, что хранится в ней в данный момент, а **тип переменной** говорит о том, что допустимо класть в эту коробочку.



Вопрос:

А как же положить в коробочку число?

Сообщите, что положить в коробочку число (ввести значение в переменную) можно с помощью команды ввода **Read** либо **Readln**.

Покажите на примерах 1, 2 из § 11, как записывается и выполняется команда **Readln**.

Обратите внимание учащихся на то, что перед каждой командой ввода рекомендуется выводить на экран поясняющий текст с информацией о том, что именно нужно ввести.

Расскажите, чем отличается команда **Read** от **Readln**.

3. Закрепление нового материала (репродуктивный метод обучения, фронтальная, индивидуальная или парная работа).

Рассмотрите с учащимися программу `summa2` примера 3 из § 11. Используйте наглядность: нарисуйте на доске ячейки для размещения в памяти значений переменных `a`, `b` и `s`. Проследите за изменениями содержимого этих ячеек в ходе выполнения программы, например при `a=13`, `b=7`. Предложите учащимся занять свои места за компьютером, загрузить систему программирования **Pascal ABC** и файл с программой `summa2` примера 3 из § 11, которую следует выполнить для различных исходных данных. Покажите учащимся, что значения для переменных `a` и `b` ввести с клавиатуры можно двумя способами: набрать числа в одной строке через пробел (несколько пробелов) и нажать клавишу **ENTER** либо каждый раз при вводе очередного числа нажимать клавишу **ENTER**.

Акцентируйте внимание учащихся на том, что с помощью команды `Writeln` можно оформить вывод результата вычислений в программе с пояснением, например `Writeln ('Сумма=', a+b);`

Далее предложите учащимся изменить программу `summa2` так, чтобы она вычисляла сумму 3 целых чисел. При необходимости отстающим ученикам помогите выполнить задание или организуйте взаимопомощь учеников друг другу (парная работа: учащихся объединить в пары так, чтобы один был более сильный, а другой более слабый).

Дополнительно учащимся, способным к раннему усвоению учебного материала, можно предложить составить программу, которая запрашивает ввод сторон прямоугольника и находит его площадь и периметр. Результат вычислений с пояснением выводится на экран.

Возможное решение:

```
Program Rectangle;
```

```
Var a, b, p : real;
```

```
Begin
```

```
  Write(' Введите длины сторон прямоугольника: '); Readln(a,b);
```

```
  Writeln(' Площадь равна ',a*b);
```

```
  Writeln(' Периметр равен ',2*(a+b));
```

```
End.
```

Далее проведите индивидуальную коррекцию усвоения полученных знаний с помощью теста (приложение 3 к уроку 12), который соответствует целям урока. Предложите учащимся сверить результаты теста друг с другом и сравнить с эталоном. Отметку за тест выставляйте только по желанию учащегося.

4. Подведение итогов урока. Вместе с учащимися обобщите изученный на уроке материал, выделив главное: организация ввода данных, организация вывода данных в языке Паскаль; возможности команды Write (или Writeln). Подведите итоги групповой работы, отметьте группы, успешно справившиеся со своими заданиями. Дайте рекомендации для устранения проблем, возникших при работе в группах. Оцените успешность достижения цели урока.

5. Домашнее задание. Изучить материал §11, ответить на вопросы 1–4, выполнить упражнение 1.

Приложения к уроку 13

Приложение 1. Задания для работы в группах

Вариант 1

1. Запишите общую структуру программы.
2. Запишите, какого типа переменные потребуются для обозначения в программе указанных данных:

Номер дня недели _____
 Среднегодовая температура воздуха _____
 Порядковый номер ученика в классном журнале _____

Вариант 2

1. Что называют переменной?
2. Запишите, какого типа переменные потребуются для обозначения в программе указанных данных:

Количество взятых в библиотеке книг _____
 Температура тела человека _____
 Номер дня в году _____

Вариант 3

1. Что значит описать переменную?
2. Запишите, какого типа переменные потребуются для обозначения в программе указанных данных:

Средний балл успеваемости _____	БІБЛІОТЕКА Мегілеўскага дзяржаўнага універсітэта імя А. А. Кудяшова
Количество уроков в неделю _____	
Номер месяца _____	

Приложение 2.

Этапы решения задачи

с помощью языка программирования:

1. Определение исходных данных задачи.
2. Определение результатов.
3. Составление алгоритма решения задачи.
4. Определение типов данных для величин (исходных данных и результатов), используемых при реализации алгоритма; их описание в программе.
5. Составление и запись текста программы в окне редактора Pascal ABC.
6. Выполнение программы.

Приложение 3. Тест по итогам урока.

1. Чем является результат выполнения команды `write (' Привет! ')`?
 - а) строковой константой
 - б) целочисленной константой
 - в) вещественной константой
 2. Сколько констант записано в команде `write (2+3)`?
 - а) одна строковая константа '2+3'
 - б) две числовые константы: 2 и 3
 - в) одна числовая константа 5
 3. Можно ли ввести значения двух переменных `a` и `b` в одной строке?
 - а) нет, нельзя
 - б) можно разделяя пробелом
 4. Какая команда выведет значения двух переменных `a` и `b` в одной строке через пробел?
 - а) `Write(a, ' ', b);`
 - б) `Writeln(a, b);`
 - в) `Read(a, b);`
- Правильные ответы: 1 – а; 2 – б; 3 – б; 4 – а.

Урок 14. КОМАНДА ПРИСВАИВАНИЯ. АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ И ВЫРАЖЕНИЯ

Цели урока:

- закрепление материала предыдущего урока;
- формирование представлений о команде присваивания; об арифметических операциях и выражениях;
- развитие логического и алгоритмического мышления, памяти, внимания.

Тип урока: усвоение новых знаний и умений.

Учащиеся должны знать арифметические операции.

Учащиеся должны уметь составлять простейшие программы с использованием команды присваивания и арифметических операций.

Программное и методическое обеспечение урока: система программирования Pascal ABC; учебное пособие «Информатика, 7», § 12 (12.1, 12.2); раздаточный материал (*приложения 1, 2 к уроку 14*); ЭСУ Algo7.

Методические рекомендации к уроку

1. Проверка знаний и закрепление умений предыдущего урока. В ходе фронтальной беседы проверьте наличие и правильность выполнения домашнего задания (упражнение 1 § 11). Выявите проблемы, с которыми столкнулись учащиеся при выполнении домашнего задания. Дайте рекомендации по их устранению.

Предложите учащимся проверить степень усвоения материала предыдущего урока с помощью теста (*приложение 1 к уроку 14*) с последующей взаимопроверкой при фронтальном обсуждении правильных ответов с целью коррекции знаний.

2. Объяснение нового материала. Напомните учащимся, что в переменные определенного типа, используемые в программах, записываются соответствующие значения.

Задайте учащимся вопрос: *Какие арифметические операции им известны из курса математики?*

Обсудите вместе с учащимися, что над переменными целого и вещественного типа можно также выполнять арифметические операции. Сообщите, что конкретные значения переменных программы можно вводить разными способами, с одним из которых они уже знакомы.

Предложите учащимся вспомнить, с помощью какой команды (**Read** либо **Readln**) они вводили данные на предыдущем уроке. Обратите внимание учащихся на то, что значения переменных вводятся не только с клавиатуры во время выполнения программы. Сообщите, что записывать значения в переменные можно непосредственно во время составления программы с помощью команды присваивания, которая обозначается «:=» (двоеточие и равно вместе).

Познакомьте учащихся с общим видом команды присваивания. Приведите примеры (*приложение 2 к уроку 14*). Обратите внимание учеников на то, что при присваивании переменной какого-либо значения старое значение переменной стирается в памяти компьютера.

Познакомьте учащихся с синтаксическими правилами записи выражений в программах, с порядком выполнения арифметических операций в выражениях. Разберите вместе с учащимися примеры 1, 2 из § 12 (12.1), которые демонстрируют работу команды присваивания.

Далее познакомьте учащихся с операциями, которые можно выполнять над переменными целого и вещественного типа. Объясните работу операций `div` и `mod` с помощью примера 3 из § 12. С помощью примера 5 из §12 рассмотрите операцию обычного деления (`/`).

3. Закрепление нового материала. Разбейте учащихся на 3 группы. Каждой группе предложите выполнить задания (*приложение 3 к уроку 14*) и в ходе фронтальной беседы представить и объяснить найденный совместно результат.

Далее предложите группам учащихся занять места за компьютерами, загрузить систему программирования Pascal ABC и на основе алгоритма программы примера 3 из § 12 составить программу, решающую задачу одного из упражнений. В частности 1-й группе предложите выполнить упражнение 3, 2-й – упражнение 5, 3-й – упражнение 6.

Далее предложите каждой группе предоставить свои программы, обосновать алгоритм решения задач.

Если останется время, предложите учащимся выполнить упражнение 4 (§ 12).

Дополнительно учащимся, способным к раннему усвоению учебного материала, можно предложить составить программу, которая вычисляет, сколько в N сутках часов, минут, секунд.

Возможное решение:

```
Var n, nh, nm, ns : integer;
```

```
Begin
```

```
Write(' Введите количество суток: '); readln(n);
```

```
nh := n*24;           {в сутках 24 часа}
```

```
nm := nh*60;         {1 час = 60 минут}
```

```
ns := nm*60;         {1 минута = 60 секунд}
```

```
Writeln(n, ' суток = ',nh, ' часов = ',nm, ' минут = ', ns, ' секунд');
```

```
End.
```

4. Подведение итогов урока. Подведите итоги групповой работы, отметьте группы, успешно справившиеся со своими заданиями. Дайте рекомендации для устранения проблем, возникших при работе в группах. Оцените успешность достижения цели урока.

5. Домашнее задание. Изучить материал §12 (12.1, 12.2).

Приложения к уроку 14

Приложение 1. Тест к уроку (входной контроль).

1. Верно ли, что команда `Writeln` без параметров только переводит курсор в начало следующей строки экрана?

а) верно;

б) неверно.

2. Вывести на экран текст «Введи число» в процессе выполнения программы можно с помощью команды:

- а) Write ('Введи число');
- б) Read ('Введи число');
- в) Writeln ('Введи число');
- г) Write («Введи число»).

3. Ввести с клавиатуры значение переменной A во время выполнения программы можно с помощью команды:

- а) Write ('A');
- б) Read (A);
- в) Readln (A);
- г) Read ('A');

4. Алгоритмом какого типа является следующая программа:

```

Begin
    Readln (a);
    Writeln ('a=', a);
    Writeln ('a=', 2*a);

```

End.

- а) алгоритм с ветвлением
- б) алгоритм с повторением
- в) линейный алгоритм

5. Что отобразится в окне вывода в результате выполнения предыдущей программы при a=5?

- а) В линейном алгоритме: каждая команда выполняется один раз
- б) некоторые команды один раз выполняются
- в) некоторые команды выполняются несколько раз
- г) все команды выполняются несколько раз

Правильные ответы:

1 – а; 2 – б, в; 3 – б, в; 4 – в; 5 –

5
a=5
a=10

; 6 – а.

Приложение 2. Оператор присваивания

<Имя переменной>:=<выражение>;

Примеры:

a:=5; {переменной a присвоить значение 5}
b:=6; {переменной b присвоить значение 6}
c:=a+b; {переменной c присвоить значение, равное сумме значений переменных a и b}
f:=10;
f:=f+2; {увеличить значение переменной f на 2}

Приложение 3. Теоретические задания для групповой работы

Что будет выведено на экран в результате выполнения следующей программы?

Вариант 1	Вариант 2
<pre>Var a, b, c : integer; Begin a:= -4; a:= a*2+1; b:= -a; c:= b mod 2; Writeln('c=', c); End.</pre>	<pre>Var a, b, c : integer; Begin a:= 3; a:= a*2+1; c:= a div 2; b:= c*c; Writeln ('b=', b); End</pre>
Ответ: c=1	Ответ: b=9

Вариант 3
<pre>Var a, b, c : integer; Begin a:= 5; a:= a*3+2; b:= a mod 5; c:= b*b; Writeln ('c=', c); End.</pre>
Ответ: c=4

**Урок 15. СТАНДАРТНЫЕ ФУНКЦИИ.
ПРЕОБРАЗОВАНИЕ ТИПОВ ЧИСЛОВЫХ ДАННЫХ**

Цели урока:

- закрепление материала предыдущего урока;
- формирование представления о стандартных функциях;
- развитие логического и алгоритмического мышления.

Тип урока: комбинированный.

Учащиеся должны знать арифметические операции и функции над переменными целого и вещественного типов.

Учащиеся должны уметь составлять и реализовывать простейшие программы с использованием арифметических операций и стандартных функций.

Программное и методическое обеспечение урока: система программирования Pascal ABC; учебное пособие «Информатика, 7», § 12 (12.3, 12.4); раздаточный материал (приложение к уроку 15); ЭСУ Algo7.

Методические рекомендации к уроку

1. Проверка знаний и закрепление умений предыдущего урока (индивидуальная работа). Предложите учащимся проверить степень усвоения мате-

риала предыдущего урока с помощью заданий в двух вариантах (приложение к уроку 15).

2. Объяснение нового материала (объяснительно-иллюстративный метод обучения в сочетании с частично-поисковым методом, фронтальная форма работы). Напомните учащимся, что в программах можно использовать переменные целого и вещественного типов. Предложите учащимся ответить на вопросы:

Какие арифметические операции можно выполнять над переменными целого типа?

Какие арифметические операции можно выполнять над переменными вещественного типа?

Сообщите учащимся, что переменные обоих типов можно обрабатывать с помощью стандартных функций. Введите понятие *стандартной функции* (Многие действия с числовыми данными выполняются путем вызова функций, встроенных в язык программирования. Такие функции называются стандартными.). Вместе с учащимися разберите пример 4 из §12. Обратите внимание учащихся на то, что **аргументы функции всегда пишутся в скобках**.

Далее познакомьте учащихся со стандартными функциями, с помощью которых можно обрабатывать переменные вещественного типа. Обратите внимание учащихся на особенности их применения с целью преобразования числовых данных для совместимости типов числовых данных. Разберите и обсудите примеры 6, 7 из § 12, которые демонстрируют преобразование и совместимость различных типов числовых данных. При разборе примера 7 обратите внимание учащихся на результат вычисления дробной части (*frac*) – он будет вещественным (в учебнике опечатка: переменной *u* типа *integer* нельзя присвоить значение: $y := \text{frac}(b)$, поскольку функция *frac* всегда дает вещественный результат). Можно задать учащимся вопрос:

Какие значения будут выведены после выполнения команд: `writeln(frac(7/2)); writeln(frac(1/5)); writeln(frac(1));` ?

Правильные ответы: 0.5 , 0.2 и 0

3. Закрепление нового материала. Предложите учащимся занять свои места за компьютером, загрузить систему программирования Pascal ABC, программу примера 6 из § 12 и выполнить ее. Задайте учащимся вопросы:

Какие переменные в этой программе являются исходными данными, а какие – результатами? (a и c – исходные данные, a, b, c, d – результаты).

*Какие изменения следует внести в программу, чтобы ее можно было выполнить для различных исходных данных? (вместо операторов присваивания записать операторы (как минимум, оператор `read`), чтобы программа запрашивала значения исходных данных – переменных *a* и *c*)*

Предложите учащимся внести изменения в программу и выполнить ее для различных исходных данных.

Далее предложите учащимся изменить набранную программу примера 8 так, чтобы она решала задачи сначала из упражнения 7, затем из упражнения 8 (§ 12). Проверьте работу учащихся. Выявите пробелы в усвоении материала. При необходимости отстающим учащимся окажите помощь либо организуйте взаимопомощь учащихся друг другу.

Выполните вместе с учащимися упражнение 1 (§ 12). Обратите внимание учеников на то, что в первом выражении возможны два варианта записи выражения на языке Паскаль:

$4+(2*3) \text{ div } 3$ – даст результат целого типа;

$4+(2*3) / 3$ – даст результат вещественного типа.

Во 2-м и 3-м выражениях тип результата real. В 4-м – integer/

Выполните вместе с учащимися упражнение 2 (§ 12). Обратите внимание учеников на то, что в программе `int_g1` ошибок нет: результат деления по модулю является целочисленным, его можно присвоить вещественной переменной *d*.

Дополнительно учащимся, способным к раннему усвоению учебного материала, можно предложить составить программу для решения следующей задачи.

Расстояние между двумя населенными пунктами S (км). Вычислите, сколько литров топлива потребуется на дорогу туда и обратно и сколько денег необходимо уплатить за это топливо, если расход топлива на 100 км составляет K (л), а 1 л топлива стоит P (руб.).

Возможное решение:

```
Var s,K,v_2s : real;
```

```
    p : integer;
```

```
Begin
```

```
Write( 'Введите расстояние в км: S = ');
```

```
Readln(S);
```

```
Write( 'Введите расход топлива (л на 100 км): ');
```

```
Readln(K);
```

```
Write( 'Введите стоимость 1 л топлива: ');
```

```
Readln(P);
```

```
v_2s := 2*s/100*K; {расход топлива на весь путь}
```

```
Writeln( 'На дорогу потребуется ',v_2s:4:2,' литров топлива');
```

```
p := trunc( p*v_2s)+1; {стоимость топлива на весь путь}
```

```
Writeln( 'За ',v_2s,' литр(ов) топлива придется уплатить ',p,' рублей');
```

```
End.
```

4. Подведение итогов урока. Подведите итоги индивидуальной или парной работы, выявите проблемы, при необходимости проведите коррекцию усвоения новых знаний. Дайте оценку успешности достижения цели на уроке.

5. Домашнее задание. Изучить материал § 12 (12.3, 12.4), выполнить упражнение 2 (если не успели обсудить на уроке).

Приложение к уроку 15. Тест к уроку (входной контроль).

**Задания по теме «Команда присваивания.
Операции над переменными целого типа»**

Вариант 1

1. Запишите следующие выражения на языке программирования Паскаль.

а) $5a$; б) $|-x + 5|$.

в) $\frac{a-b}{2}$; г) $\frac{a}{bc}$.

2. В программе описаны переменные:

var a, b : integer; c : real;

Какой переменной можно присвоить значение выражения a/b?

а) переменной c

б) переменной a

в) переменной b

3. В программе описаны переменные:

var a: integer; b: real;

Какая команда недопустима в программе?

а) $a:=\text{frac}(b)$;

б) $a:=\text{trunc}(b)$;

в) $b:=a \text{ div } 4$;

4. Укажите неверный оператор:

а) $a = 19$;

б) $b:=4*a$;

в) $5:=m$.

5. Установите соответствие:

1) End а) Присвоить

2) := б) Описание переменных

3) Var в) Написать (вывести)

4) Write г) Конец

6. Что будет выведено на экран в результате выполнения следующей программы:

Var a,b,c: integer;

Begin

a:=1; b:=2; c:=3;

Writeln(a, b, c);

Writeln(a, ' ', b, ' ', c); End.

Правильные ответы:

2 – а; 3 – а; 4 – в; 5 – г, 2а, 3б, 4в; 6 –

123
1 2 3

Вариант 2

1. Запишите следующие выражения на языке программирования Паскаль:

а) $x^2 + 1$; б) $-5a$

в) $\frac{2}{bc}$; г) $\frac{a-b}{2}$

2. В программе описаны переменные:

`var a, b : integer; c : real;`

Какой переменной можно присвоить значение выражения $a * 1.0$?

- а) переменной a
- б) переменной b
- в) переменной c

3. В программе описаны переменные:

`var a: integer; b: real;`

Какая команда недопустима в программе?

- а) `a:= trunc (b);`
- б) `a:=frac(b);`
- в) `a:=a mod 4;`

4. Укажите неверный оператор:

- а) `c:=5-f;`
- б) `100+5*a:= p;`
- в) `f:=p;`

5. Установите соответствие:

- | | |
|----------|------------------------|
| 1) Var | а) Начало |
| 2) Read | б) Описание переменных |
| 3) Begin | в) Присвоить |
| 4) := | г) Прочитать (ввести) |

6. Что будет выведено на экран в результате выполнения следующей программы:

```
var a,b,c: integer;
Begin
a:=3; b:=4; c:=5;
Writeln(a, ' ', b, ' ', c);
Writeln (a, b, c);
End.
```

Правильные ответы: 2 – в; 3 – б; 4 – б; 5 – 1б, 2г, 3а, 4в; 6 –

3 4 5

345

Урок 16. РЕШЕНИЕ ЗАДАЧ НА СОСТАВЛЕНИЕ ЛИНЕЙНЫХ АЛГОРИТМОВ

Цели урока:

- закрепление материала предыдущих уроков;
- формирование умений составлять простейшие линейные алгоритмы и программы;
- развитие логического и алгоритмического мышления;

Тип урока: закрепление изучаемого материала.

Учащиеся должны уметь составлять простейшие программы с использованием арифметических операций и выражений.

Программное и методическое обеспечение урока: система программирования Pascal ABC; учебное пособие «Информатика, 7», § 13; раздаточный материал (*приложение к уроку 16*); ЭСУ Algo7.

Методические рекомендации к уроку

1. Проверка знаний и закрепление умений предыдущего урока. Предложите учащимся проверить степень усвоения материала предыдущего урока с помощью теста (*приложение к уроку 16*) с последующей коррекцией знаний при фронтальном сравнении ответов учеников с эталоном.

В ходе фронтальной беседы проверьте наличие и правильность выполнения домашнего задания (упражнение 2 § 12) с целью выявления проблем, с которыми столкнулись учащиеся при выполнении домашнего задания. Дайте рекомендации по их устранению.

Вопросы на закрепление (развивающие логическое и алгоритмическое мышление):

- Часто ли мы используем алгоритмы? (*ежедневно по несколько раз*)
- Вам нужно объяснить шестилетнему ребенку (младшему брату или сестре) как купить мороженое в магазине. Является ли ваша инструкция-объяснение алгоритмом? (*да*)
- Легко ли составить «универсальный» алгоритм покупки мороженого (который поймут все шестилетние дети)? (*нет*) Почему? (*трудно предусмотреть все варианты развития событий, поведение исполнителя (ребенка)*)
- Что проще сделать: напрямую «управлять» исполнителем (сделай это, теперь сделай то, и т.д.) или придумать полную инструкцию на будущее, предусмотрев все возможные варианты развития событий? (*управлять проще, чем составлять алгоритм действий*)

• Для чего же нужно составлять алгоритмы? (*напрямую управлять можно только одним исполнителем, а, составив алгоритм, можно управлять несколькими исполнителями одновременно*)

• Являются ли алгоритмом вся инструкция по технике безопасности? (*нет, поскольку в ней есть требование типа «запрещается ...», но часть инструкции может содержать алгоритм*)

• Является ли алгоритмом следующая сокращенная запись инструкции для вызова программы управления звуковыми устройствами (на с. 30 учебника):
Пуск → Все программы → Стандартные → Развлечения → Громкость?
(*да*)

• Является ли алгоритмом следующая инструкция по работе в окне редактора Pascal ABC (на с.57 учебника):

- 1) Выбрать меню **Файл**
- 2) Выбрать команду **Новый**
- 3) Набрать текст программы в окне редактора

(*да, является*)

• Что такое алгоритм? (*Алгоритм – это описание последовательности действий исполнителя, направленных на решение поставленной задачи.*)

• Какие вы знаете виды алгоритмов? (*линейные, с ветвлением, с повторением*)

• Какие вы знаете формы записи алгоритмов? (*словесная, графическая (блок-схема), программа*)

• Можно ли составить алгоритм, не зная, кто будет исполнителем? (*нет, но если исполнитель явно не указан, то обычно подразумевается человек*)

• Кто является исполнителем программы, запущенной на выполнение в системе Pascal ABC? (*компьютер*)

2. Составление линейных программ. Разберите вместе с учащимися алгоритмы решения задачи 1 (программа `сifr_p`).

Вопросы:

• Для чего в программе записаны комментарии? (*для разъяснения отдельных предложений, фрагментов программы*)

• Нужно ли писать комментарии к готовой чужой программе (из учебника)? (*не обязательно*)

• Нужно ли писать комментарии к программе, когда Вы решаете задачу по программированию? (*желательно, так как это проясняет, что Вы как автор программы имеете в виду*)

• Как следует изменить программу `сifr_p`, чтобы она выводила первую цифру числа? вычисляла сумму цифр числа?

Разберите вместе с учащимися алгоритмы решения задачи 2 из § 13 (программа `summa_deneg`).

3. Самостоятельная работа. Предложите учащимся выполнить программу `summa_deneg` из § 13 для различных исходных данных. Далее предложите учащимся изменить программу `summa_deneg` так, чтобы она решала следующую задачу.

Петя купил в магазине N порций мороженого по цене P рублей и K плиток шоколада по цене S рублей. Сколько всего денег потратил Петя?

Сильным учащимся предложите самостоятельную работу на составление программ для решения задачи упражнения 6 (§13).

Дополнительно учащимся, способным к раннему усвоению учебного материала, можно предложить составить программу, которая вычисляет сумму цифр данного трехзначного числа и выводит на экран результат.

Ответ:

```
Var a,c1,c2,c3:integer;
Begin
  Write( 'введите трехзначное число ' );
  Readln (a);
  c1:=a div 100; {выделение первой цифры числа}
  c3:=a mod 10; {выделение последней цифры}
  a:=a div 10;
  c2:=a mod 10; {выделение второй цифры}
  Writeln( 'сумма цифр= ',c1+c2+c3);
End.
```

Дополнительное задание к задаче: вывести результат в виде:

сумма цифр числа ... равна ...

(вместо многоточий должны выводиться число и его сумма цифр соответственно)

4. Подведение итогов урока. Подведите итоги самостоятельной работы.
5. Домашнее задание. Изучить материал §13, выполнить упражнение 1.

Приложение к уроку 16. Тест по теме «Стандартные функции. Совместимость типов» (входной контроль)

Вариант 1

1. Какие действия можно выполнить над переменными a , b типа `real`? **Выберите несколько вариантов ответа:**

- 1) $a+b$;
- 2) $a-b$;
- 3) $a*b$;
- 4) a/b ;
- 5) $a \text{ div } b$.

2. Какие действия можно выполнить над переменными a, b типа integer?

Выберите несколько вариантов ответа:

- 1) $a+b$;
- 2) $a-b$;
- 3) $a*b$;
- 4) a/b ;
- 5) $a \bmod b$.

3. Выберите верное описание переменных A и B, если они используются в следующей команде присваивания: $B:=A/2$;

1) Var A, B: Integer;

2) Var A: Real;
B: Integer;

3) Var A: Integer;
B: Real.

4. Алгоритм – это:

- 1) набор команд
- 2) последовательность действий
- 3) описание последовательности действий
- 4) процесс решения поставленной задачи

5. Какая команда допустима в линейном алгоритме:

- 1) ожидать пока загорится зеленый сигнал светофора
- 2) взбивать до однородности
- 3) добавить одну чашку молока

6. В алгоритме с ветвлением:

- 1) из двух команд выполняется только одна
- 2) каждая команда выполняется один раз
- 3) некоторая команда выполняется несколько раз

Правильные ответы:

1 – 1, 2, 3, 4; 2 – 1, 2, 3, 5; 3 – 3; 4 – 3; 5 – 3; 6 – 1.

Вариант 2

1. Какие действия можно выполнить над переменными a, b типа real? **Выберите несколько вариантов ответа:**

- 1) $a+b$;
- 2) $a \bmod b$;
- 3) $a-b$;
- 4) $a*b$;
- 5) a/b .

2. Какие действия можно выполнить над переменными a, b типа integer?

Выберите несколько вариантов ответа:

- 1) $a+b$; 4) $a*b$;
2) $a-b$; 5) $a \text{ div } b$.
3) a/b ;

3. Выберите верное описание переменных A и B, если они используются в следующей команде присваивания: $B := A \text{ div } 2$;

1) Var A: Integer;
 B: Real;

2) Var A, B: Real;

3) Var A, B: Integer;

4. Выполнение алгоритма – это:

- 1) последовательность команд исполнителю
- 2) система команд исполнителя
- 3) набор действий исполнителя
- 4) последовательность действий исполнителя

5. В линейном алгоритме:

- 1) из двух команд выполняется только одна
- 2) каждая команда выполняется один раз
- 3) некоторая команда выполняется несколько раз

6. В алгоритме с повторением:

- 1) некоторая команда выполняется несколько раз
- 2) каждая команда выполняется один раз
- 3) из двух команд выполняется только одна

Правильные ответы:

1 – 1, 3, 4, 5; 2 – 1, 2, 4, 5; 3 – 3; 4 – 4; 5 – 2; 6 – 1.

Урок 17. РЕШЕНИЕ ЗАДАЧ НА СОСТАВЛЕНИЕ АЛГОРИТМОВ

Цели урока:

- закрепление материала предыдущих уроков;
- развитие логического и алгоритмического мышления;
- использование элементов опережающего обучения для подведения к команде ветвления;
- развитие умений составлять линейные программы.

Тип урока: комбинированный.

Учащиеся должны уметь составлять программы.

Программное и методическое обеспечение урока: система программирования Pascal ABC; учебное пособие «Информатика, 7», § 13; ЭСУ Algo7.

Методические рекомендации к уроку

1. Проверка знаний и закрепление умений предыдущего урока. В ходе фронтальной беседы проверьте наличие и правильность выполнения домашнего задания (упражнение 1 §13); выявите проблемы, с которыми столкнулись учащиеся при выполнении домашнего задания с целью коррекции знаний. Осуществите актуализацию знаний и мотивацию учащихся на практическое использование учебного материала с опорой на знания и умения, полученные на предыдущих уроках.

2. Составление линейных программ. Разберите с учениками задачу 3 (программу vv1) из § 13. Вспомните назначение стандартной функции `sqrt`. Как можно переписать выражение для вычисления значения переменной `V` с использованием функции `sqrt`?

3. Составление алгоритмов с ветвлением. (фронтальная форма работы).

Вспомните с учениками виды алгоритмов: линейные, с ветвлением, с повторением.

Вопросы:

I. Какие алгоритмы называют линейными? (в которых все команды выполняются последовательно (линейно) одна за другой) Приведите примеры.

II. Допустимы ли в линейном алгоритме для исполнителя Человек следующие команды:

- сделать шаг (да)
- повернуть налево (да)
- развернуться на 180° (да)
- добавить одну чашку молока (да)
- посолить по вкусу (нет)
- взбивать до однородности (нет)
- выложить на блюдо (да)

III. Какие алгоритмы называют алгоритмами с ветвлением? (в которых нужно совершать различные действия, в зависимости от выполнения или невыполнения некоторых условий). Приведите примеры (собираясь на прогулку, если идет дождь, то нужно взять зонт; если поднялась температура, то нужно обратиться к врачу).

IV. Задача-игра. Один ученик называет произвольное число, а другой должен определить, является ли названное число четным или нечетным. Можно ли составить алгоритм для определения четности или нечетности числа? (да) Составьте такой алгоритм.

Решение (словесный алгоритм):

I) узнать число;

II) найти остаток от деления этого числа на 2;

III) если остаток равен 0, то число четное, иначе число нечетное

К какому виду алгоритмов относится этот алгоритм? (алгоритм с ветвлением)

В чем основное отличие алгоритма с ветвлением от линейного алгоритма? (В алгоритме с ветвлением выполнение некоторой команды зависит от соблюдения (истинности) условия.)

Какие команды должен уметь выполнять исполнитель, чтобы правильно выполнить этот алгоритм? (Какова система команд исполнителя?) {делить с остатком; сравнить с 0}

Вспомните с учениками, как эти операции реализуются в среде Pascal ABC.

* В классе, где большинство учащихся способны к раннему усвоению учебного материала, можно, используя методику опережающего обучения, заменить русские слова в словесном алгоритме на английские и, таким образом, показать команду ветвления. Далее можно предложить составить программу на языке Паскаль для решения этой задачи.

Программа:

```
var a : integer;
begin
  write(' Введите целое число: ');
  read(a);
  if a mod 2 = 0
  then writeln(' Число четное ')
  else writeln(' Число нечетное ')
end.
```

V. Задача-игра. Один ученик называет произвольное число, а другой должен определить, является ли названное число целым или дробным. Можно ли составить алгоритм, который будет определять, является ли заданное число целым или дробным? (да) Составьте такой алгоритм.

Решение (словесный алгоритм):

I) узнать число;

II) найти дробную часть числа;

III) если дробная часть числа равна 0, то число целое, иначе число дробное

Вспомните с учениками, как эти операции реализуются в среде Pascal ABC.

* Программа:

```
var    r : real;
begin
  write( ' Введите число: ');
  read(r);
  if frac(r) =0
    then writeln( ' Число целое ')
    else writeln( ' Число дробное ')
end.
```

4. Самостоятельная работа. Предложите ученикам самостоятельно разобраться с решением задачи 4 из § 13. Предложите учащимся самостоятельно или под руководством учителя выполнить программу skor из § 13 для различных исходных данных.

Могут ли s и t быть вещественными числами?

Далее предложите учащимся на основе алгоритма программы skor выполнить упражнение 7 (§ 13).

Дополнительно учащимся можно предложить составить программу для решения следующей задачи.

Дорожка шириной в одну плитку и длиной L м выложена квадратными плитками с длиной стороны T см. Сколько целых плиток понадобится? Какой длины кусок плитки надо отрезать, чтобы замостить ту часть дорожки, на которую уже не поместится целая плитка?

Ответ:

Var

L, T : integer;

Begin

Write(' Введите длину дорожки (м): ');

Readln(L);

Write(' Введите длину стороны плитки (см): ');

Readln(T);

Writeln(' Количество целых плиток: ', L*100 div T); // 1м=100см

Writeln(' Отрезать надо ', L*100 mod T, ' см ');

End.

5. Подведение итогов урока. Оцените работу учащихся и успешность достижения цели на уроке.

6. Домашнее задание. Повторить материалы § 12–13, записи в тетради, выполнить упражнение 3 §13.

Урок 18. РЕШЕНИЕ ЗАДАЧ НА СОСТАВЛЕНИЕ АЛГОРИТМОВ

Цели урока:

- развитие логического и алгоритмического мышления;
- использование элементов опережающего обучения для подведения к команде повторения.
- закрепление материала предыдущих уроков;
- проверка уровня усвоения материала.

Тип урока: комбинированный

Учащиеся должны уметь составлять программы.

Программное и методическое обеспечение урока: система программирования Pascal ABC; учебное пособие «Информатика, 7», § 13; раздаточный материал (*приложение к уроку 18*).

Методические рекомендации к уроку

1. Проверка знаний и закрепление умений предыдущего урока. В ходе фронтальной беседы проверьте наличие и правильность выполнения домашнего задания (упражнение 3 §13); выявите проблемы, с которыми столкнулись учащиеся при выполнении домашнего задания с целью коррекции знаний. Осуществите актуализацию знаний и мотивацию учащихся на практическое использование учебного материала с опорой на знания и умения, полученные на предыдущих уроках.

2. Составление алгоритмов с повторением.

Вопросы:

I. Какие алгоритмы называют циклическими? (в которых содержатся повторяющиеся действия) Приведите примеры (утренняя зарядка, работа первоклассника в прописи, работа над ошибками и др.).

II. Задача-игра. Для исполнителя Ученик составить алгоритм рисования пунктирной линии, состоящей из 5 черточек.

Пример такого алгоритма:

Повторить 5 раз следующие команды:

- 1) опустить карандаш на бумагу (мел на доску);
- 2) сместиться на 1 см вправо;
- 3) поднять карандаш (мел);
- 4) сместиться на 1 см вправо.

К какому виду алгоритмов относится этот алгоритм? (алгоритм с повторением)

III. * В чем основное отличие между линейными, циклическими и ветвящимися алгоритмами? (В линейном алгоритме каждая команда выполняется ровно один раз. В алгоритме с повторением некоторая команда выполняется

несколько раз. В алгоритме с ветвлением выполнение некоторой команды зависит от соблюдения (истинности) условия.)

* Можно ли составить алгоритм для рисования пунктирной линии из 5 черточек для исполнителя Чертежник? (да)

IV. Предложите учащимся составить программы для решения упражнений 3–6 к §13

3. Закрепление изученного материала. Проведите индивидуальную коррекцию усвоения полученных знаний с помощью теста (*приложение 1 к уроку 18*), который соответствует целям урока. Предложите учащимся сверить результаты теста друг с другом и сравнить с эталоном. Отметку за тест выставляйте только по желанию учащегося.

4. Составление линейных программ. Предложите учащимся самостоятельно или под руководством учителя составить программы для решения упражнений 4 – 6 к §13

Если останется время, предложите учащимся составить программы для решения следующей задачи.

Размер файла составляет N байт. Определить, сколько килобайт и байт занимает этот файл ($1 \text{ Кбайт} = 1024 \text{ байт}$).

5. Подведение итогов урока. Оцените работу учащихся и успешность достижения целей на уроке. Дайте рекомендации по устранению пробелов в знаниях и умениях.

6. Домашнее задание. Повторить материалы § 10–13, записи в тетради. Подготовиться к самостоятельной работе.

Приложение к уроку 18. Тест по итогам урока

Вариант 1	Вариант 2
<p>1. Алгоритм – это:</p> <ul style="list-style-type: none"> а) набор команд б) последовательность действий в) описание последовательности действий г) процесс решения поставленной задачи <p>2. В линейном алгоритме:</p> <ul style="list-style-type: none"> а) некоторая команда выполняется, если истинно условие б) некоторая команда выполняется несколько раз в) каждая команда выполняется один раз <p>3. Можно ли составить алгоритм с повторением для рисования пунктирной линии?</p> <ul style="list-style-type: none"> а) да, можно б) нет, нельзя <p>4. В алгоритме с ветвлением:</p> <ul style="list-style-type: none"> а) каждая команда выполняется один раз б) некоторая команда выполняется, если истинно условие в) некоторая команда выполняется несколько раз г) все команды выполняются несколько раз <p>5. Какой формы записи алгоритмов не существует?</p> <ul style="list-style-type: none"> а) фрагментарная (фрагменты некоторых команд) б) словесная (на родном языке) в) графическая (блок-схема) г) на языке программирования (программа) 	<p>1. Что является алгоритмом?</p> <ul style="list-style-type: none"> а) перечень устройств компьютера б) процесс приготовления бутербродов в) действия повара, делающего торт г) описание порядка действий для приготовления клея <p>2. Какая команда (команды) допустима в линейном алгоритме:</p> <ul style="list-style-type: none"> а) сделать шаг б) перемешать до однородности в) перейти дорогу, если загорелся зеленый свет <p>3. Можно ли составить алгоритм с ветвлением для определения четности или нечетности числа?</p> <ul style="list-style-type: none"> а) нет, нельзя б) да, можно <p>4. В алгоритме с повторением:</p> <ul style="list-style-type: none"> а) некоторая команда выполняется несколько раз б) каждая команда выполняется один раз в) некоторая команда выполняется, если истинно условие г) все команды выполняются несколько раз <p>5. Какой формы записи алгоритмов не существует?</p> <ul style="list-style-type: none"> а) словесная (на родном языке) б) графическая (блок-схема) в) на языке программирования (программа) г) фрагментарная (фрагменты некоторых команд)

Правильные ответы:

Вариант 1: 1 – в, 2 – в, 3 – а, 4 – б, 5 – а.

Вариант 2: 1 – г, 2 – а, 3 – б, 4 – а, 5 – г.

Урок 19. САМОСТОЯТЕЛЬНАЯ РАБОТА

Цели урока:

- развитие логического и алгоритмического мышления;
- проверка уровня усвоения теоретического материала, сформированности умений по составлению программ, реализующих линейные алгоритмы.

Тип урока: проверка и оценка знаний и умений.

Учащиеся должны уметь составлять линейные программы.

Программное и методическое обеспечение урока: система программирования Pascal ABC; раздаточный материал.

Методические рекомендации к уроку

1. Проверка усвоения учебного материала.

Вариант 1

Теоретическая часть

Уровень 1

1. (1 балл) Величины, значения которых не изменяются в процессе выполнения программы, называют:

- а) переменными
- б) константами

2. (1 балл) Переменные в программе описываются после зарезервированного слова

- а) begin
- б) var

3. (1 балл) Какая функция вычисляет квадрат числа?

- а) abs
- б) sqr

Уровень 2

4. (2 балла) Укажите те команды, которые позволяют ввести данные во время выполнения программы:

- а) read
- б) readln
- в) write
- г) writeln

5. (2 балла) Переменные типа real используют для хранения

- а) только целых чисел
- б) вещественных чисел
- в) символов

Уровень 3

6. (3 балла) В алгоритме с ветвлением:

- а) каждая команда выполняется один раз
- б) некоторая команда выполняется, если истинно условие
- в) некоторая команда выполняется несколько раз

Практическая часть

Уровень 4

7. (7 баллов). Составьте программу для решения задачи.

На банковском счете в начале месяца находилась сумма L тыс. рублей, а в конце увеличилась до F тыс. рублей. Составить программу вычисляющую, на сколько тыс. рублей изменилась сумма на счете.

Уровень 5

8 (10 баллов). Составьте программу для решения задачи.

На кондитерской фабрике конфеты из бункера, вмещающего S килограммов конфет, упаковывают в коробки по N граммов. Сколько коробок будет заполнено? Сколько граммов конфет останется?

Вариант 2

Теоретическая часть

Уровень 1

1. (1 балл) Величины, значения которых изменяются в процессе выполнения программы, называют:

- а) переменными
- б) константами

2. (1 балл) Команды в программе записываются после зарезервированного слова

- а) begin
- б) var

3. (1 балл) Какая функция вычисляет модуль числа?

- а) abs
- б) sq

Уровень 2

4. (2 балла) Укажите те команды, которые позволяют вывести результат выполнения программы:

- а) read
- б) readln

- в) write
- г) writeln

5. (2 балла) Переменные типа integer используют для хранения
- а) только целых чисел
 - б) вещественных чисел
 - в) символов

Уровень 3

6. (3 балла) В алгоритме с повторением:
- а) каждая команда выполняется один раз
 - б) некоторая команда выполняется, если истинно условие
 - в) некоторая команда выполняется несколько раз

Практическая часть

Уровень 4

7. (7 баллов). Составьте программу для решения задачи.

После окончания 7-го класса в классе было S отличников, а после окончания 9-го класса уменьшилось до V отличников. Составить программу вычисляющую, на сколько изменилось количество отличников.

Уровень 5

8. (10 баллов). Составьте программу для решения задачи.

Для пошива полотенец рулон ткани длиной D м нарезают по M см. Сколько отрезков получится? Какой будет длина остатка?

Критерий оценивания работ:

Сумма набранных баллов	Отметка
1	1
2	2
3 – 4	3
5 – 6	4
7 – 9	5
10 – 12	6
13 – 15	7
16 – 20	8
21 – 25	9
26 – 27	10

2. Домашнее задание. Повторить материалы § 10–13.

Урок 20. ГРАФИЧЕСКИЕ ВОЗМОЖНОСТИ ЯЗЫКА ПРОГРАММИРОВАНИЯ PASCAL ABC

Цели урока:

- закрепление материала предыдущих уроков;
- формирование представления о графических возможностях языка программирования, о командах рисования разноцветных точек и линий;
- развитие познавательного интереса, логического и алгоритмического мышления.

Тип урока: усвоение новых знаний и умений.

Учащиеся должны уметь: реализовывать линейные программы с использованием графических примитивов.

Программное и методическое обеспечение урока: система программирования Pascal ABC, учебное пособие «Информатика 7», § 14 (14.1–14.3); раздаточный материал – памятка по работе с графикой в Pascal ABC (см. приложение к уроку, дополненный вариант программы *Linii*); ЭСУ *Algo7*.

Методические рекомендации к уроку

1. Актуализация знаний и мотивация учащихся на изучение учебного материала. Проведите анализ результатов самостоятельной работы предыдущего урока, акцентируйте внимание учащихся на типичных ошибках практической работы, проведите коррекцию знаний.

Вспомните с учениками о возможностях рисования с помощью компьютера. Можно вспомнить некоторые инструменты для рисования графических примитивов: карандаш (позволяет рисовать отдельные точки), линия, овал, прямоугольник, текст (позволяет добавлять текстовое сообщение); возможности использования цвета рисования. Поясните, что рисование в Paint представляет собой вариант прямого управления исполнителем – графическим редактором. Расскажите о том, что система программирования Pascal ABC позволяет использовать графические возможности компьютера, рисовать при помощи программ. Какие это дает преимущества? Можно создать программу рисования некоторой фигуры, например, ракеты, а затем использовать эту программу для рисования ракет разного размера, расположенных (повернутых) под разными углами и т.п. Фигуры могут быть и более сложными. Кроме того, можно написать программу для создания анимированного изображения – например, движущейся ракеты.

2. Объяснение нового материала.

Поясните, что для работы с графикой в системе Pascal ABC необходимо вызвать графический модуль *GraphAbc* (подключить библиотечный модуль). Предложите ученикам вспомнить из материала 6 класса, как подключали исполнителя Чертежник (*uses Drawman*). Какая команда служит для подключения имеющихся библиотечных модулей? (*uses*). Значит, как подключить графический модуль *GraphAbc*? (*uses GraphAbc*)

Обратите внимание учащихся на то, что рисование осуществляется в специальном графическом окне (возможность рисовать одновременно в нескольких окнах в Pascal ABC отсутствует). Размеры этого окна можно задать с помощью команды `SetWindowSize (X,Y)`, где X – размер (в пикселях) по горизонтали, Y – размер по вертикали.

Вспомните с учащимися (из темы «Обработка графической информации» 6 класса), из каких элементов строится графическое изображение (из точек – пикселей). Поясните, что каждый пиксель имеет свои координаты. Начало координат находится в левом верхнем углу. То есть в левом верхнем углу находится пиксель с координатами $(0, 0)$. Координаты остальных пикселей – это их номера по горизонтали (слева направо) и по вертикали (сверху вниз) – см. рис. 3.4 § 14 в учебнике [2, с. 86]. Очевидно, что координаты пикселя – целые числа.

Попросите учеников привести примеры координат пикселей.

Познакомьте учеников с командой рисования точек `SetPixel`. Разберите вместе с учащимися пример 1 из § 14.

Расскажите о способе задания цвета линии с помощью символических констант, познакомьте с командой `SetPenColor`. Далее разберите вместе с учащимися пример 2 (с. 89).

Вопрос: Можно ли в команде `Line` использовать переменные вместо констант? (Да) Какого типа должны быть эти переменные? (*integer*)

Почему программа в примере 2 называется *Linii*? Дополним программу *Linii*. Предложите ученикам загрузить готовую программу и выполнить ее.

```
Program Linii;  
uses GraphABC;  
var x,y1,y2: integer;  
begin  
  SetWindowSize(640,480);  
  SetPenColor(clRed);  
  
  x:=20;  
  y1:=10;  
  y2:=20;  
  
  Line(x,y1,x,y2);  
  y1:=y1+20;  
  y2:=y2+20;  
  Line(x,y1,x,y2);  
  y1:=y1+20;  
  y2:=y2+20;  
  Line(x,y1,x,y2);  
  y1:=y1+20;
```

```
y2:=y2+20;  
Line(x,y1,x,y2);  
y1:=y1+20;  
y2:=y2+20;  
Line(x,y1,x,y2);  
end.
```

Каков результат выполнения программы? (*вертикальная пунктирная линия из 5 черточек*)

Содержит ли программа повторяющиеся части? Какие команды повторяются?

Составьте словесный алгоритм к программе. Как можно сократить словесный алгоритм?

* Учитель может познакомить учеников с блок-схемой данного алгоритма и ввести команду повторения `for` и составной оператор `begin ... end`

```
uses GraphABC;  
var x,y1,y2,i: integer;  
begin  
  SetWindowSize(640,480);  
  SetPenColor(clRed);  
  
  x:=20;  
  y1:=10;  
  y2:=20;  
  Line(x,y1,x,y2);  
  for i:=1 to 4 do  
    begin  
      y1:=y1+20;  
      y2:=y2+20;  
      Line(x,y1,x,y2);  
    end;  
  end.
```

Сколько черточек пунктира нарисует программа? Почему?

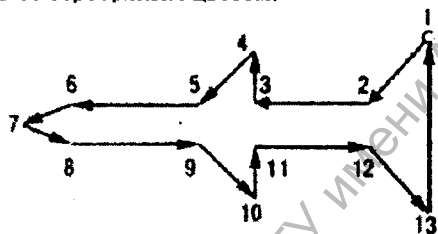
Расскажите о способе задания толщины линий с помощью символических констант, познакомьте с командой `SetPenWidth`. Далее разберите вместе с учащимися пример 3 из § 14.

Расскажите о способах задания стиля линии (пера). Обсудите, как закрашиваются замкнутые фигуры с помощью команды `FloodFill`. Обратите внима-

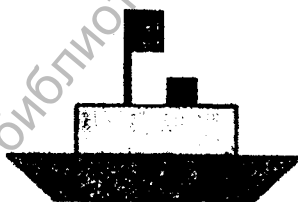
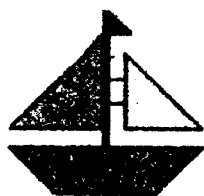
ние учеников на то, что закрашивать можно только замкнутые фигуры, контур которых нарисован одним цветом.

3. **Закрепление нового материала.** Предложите учащимся занять свои места за компьютером, загрузить систему программирования Pascal ABC, программы примеров 3, 4 из § 14 и выполнить их, меняя цвет пера и координаты точек. Проверьте сделанное задание.

Если останется время, предложите учащимся нарисовать ракету (см. рисунок) и закрасить ее серебряным цветом.



Дополнительно учащимся можно предложить составить программу, которая позволит нарисовать кораблик (на выбор учащихся):



4. **Подведение итогов урока. Рефлексия.**

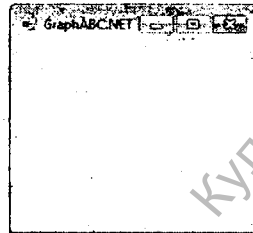
5. **Домашнее задание.** Изучить материал §14 (14.1–14.3), ответить на вопросы 1–4, нарисовать кораблик (на выбор учащихся).

Приложение к уроку 20. Памятка по работе с графикой в Pascal ABC.

Модуль GraphABC предназначен для создания графических и анимационных программ. Рисование происходит в специальном *графическом окне*. Начало координат (0,0) находится в левом верхнем углу. Координаты пикселей – целые положительные числа.

Для работы с графикой используется программа вида:

```
uses GraphABC;
begin
  команды
end.
```



Изображение можно построить из простых геометрических фигур (графических примитивов): точек, отрезков, прямоугольников, окружностей и т.д. Рисование различных геометрических фигур осуществляется с помощью специальных команд.

Команды для рисования

Команда	Действие
SetWindowSize (X, Y);	Установить размеры графического окна (в пикселях)
ClearWindow (c).	Очистить (залить) окно цветом c
SetPixel (x, y, c).	Нарисовать точку с координатами (x,y) цветом c
Line (x1, y1, x2, y2);	Нарисовать отрезок от точки (x1,y1) до точки (x2,y2)
FloodFill (x, y, c)	Закрасить замкнутую область, содержащую точку (x, y) цветом c
SetPenColor (c)	Установить цвет пера (рисование линий и контуров)
SetPenWidth (n)	Установить толщину пера в пикселях
SetPenStyle (s)	Установить стиль пера
SetBrushColor (c)	Установить цвет кисти (заливка фигур)
Rectangle (x1, y1, x2, y2)	Нарисовать прямоугольник, заданный координатами противоположных вершин (x1,y1) и (x2,y2)

Окончание табл

Команда	Действие
Circle (x, y, r)	Нарисовать окружность с центром в точке (x,y) и радиусом r
TextOut (x, y, 'текст')	Вывести текст в прямоугольник, где (x, y) – координаты левого верхнего угла прямоугольника
SetFontStyle (fs)	Установить стиль шрифта
SetFontSize (n)	Установить размер шрифта в пикселях

Цвета:

c	Цвет	c	Цвет	c	Цвет
clBlack	Черный	clFuchsia	Сиреневый	clPurple	Фиолетовый
clBlue	Синий	clSkyBlue	Голубой	clNavy	Темно-синий
clGreen	Зеленый	clAqua	Бирюзовый	clTeal	Сине-зеленый
clYellow	Желтый	clLime	Салатный	clOlive	Оливковый
clRed	Красный	clMaroon	Темно-красный	clBrown	Коричневый
clGray	Серый	clLtGray	Светло-серый	clDkGray	Темно-серый
clWhite	Белый	clSilver	Серебристый	clCream	Кремовый

Стили пера:

s	Стиль
psSolid	Сплошное
psDash	Штриховое
psDot	Пунктирное
psDashDot	Штрихпунктирное
psDashDotDot	Альтернативное штрихпунктирное
psClear	Прозрачное

Стили шрифта:

fs	Стиль
fsBold	Жирный
fsItalic	Курсив
fsUnderline	Подчеркнутый
fsNormal	Обычный

Урок 21. РЕШЕНИЕ ЗАДАЧ НА СОСТАВЛЕНИЕ ЛИНЕЙНЫХ АЛГОРИТМОВ С ИСПОЛЬЗОВАНИЕМ ГРАФИЧЕСКИХ ВОЗМОЖНОСТЕЙ

Цели урока:

- закрепление материала предыдущих уроков;
- формирование умений составлять линейные программы с использованием команд рисования разноцветных прямоугольников и окружностей;
- обобщение и систематизация знаний и умений;
- развитие логического и алгоритмического мышления.

Тип урока: комбинированный.

Учащиеся должны уметь составлять линейные программы с использованием графических примитивов.

Программное и методическое обеспечение урока: система программирования Pascal ABC; учебное пособие «Информатика, 7») § 14 (п. 14.4); раздаточный материал – памятка по работе с графикой в Pascal ABC (приложение к уроку 20); ЭСУ Algo7.

Методические рекомендации к уроку

1. Актуализация знаний предыдущего урока. Предложите ученикам проверить степень усвоения изученного материала, выполнив тестовые задания (приложение к уроку).

Предложите ученикам проверить результаты теста друг у друга. Обсудите возможные ошибки.

Предложите учащимся ответить на вопрос: *С помощью каких команд можно нарисовать закрашенный треугольник?*

2. Объяснение нового материала.

Расскажите ученикам о возможности рисования прямоугольников и окружностей с помощью команд `Rectangle` и `Circle` соответственно. Поясните, что цвет контура этих фигур нужно задать заранее с помощью команды `SetPenColor`. Разберите с учениками пример 5 из учебника.

Сообщите, что прямоугольники и окружности можно рисовать закрашенными. Для этого нужно предварительно указать цвет кисти командой `SetBrushColor`. Обсудите, какие следует внести изменения в программу из примера 5, чтобы прямоугольник был закрашен зеленым цветом, а окружность – красным.

Расскажите ученикам о возможности вывода текстовых сообщений командой `TextOut`, с помощью которых можно подписать рисунок.

Разберите вместе с учащимися алгоритм примера 6 из § 14.

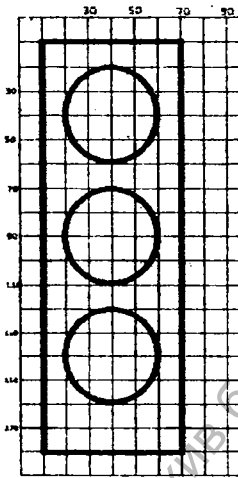
3. **Закрепление нового материала** Предложите учащимся загрузить и выполнить программу примера 6 из § 14. Предложите изменить программу, меняя размеры фигур и цвета раскраски.

Предложите ученикам самостоятельно загрузить и выполнить программу примера 7. Предложите изменить программу, меняя цвета раскраски фигур.

Далее предложите учащимся ответить на вопросы упражнения 1.

4. Подведение итогов урока. Рефлексия.

5. **Домашнее задание.** Изучить материал §14 (14.4), ответить на вопросы 1-6, выполнить упражнение 2 (один рисунок на выбор). Можно предложить составить алгоритм для рисования светофора:



Возможное решение:

```
Program svetofor;  
uses GraphABC;  
var x,y,r,i: integer;
```

```
begin  
  SetWindowSize(640,480);  
  SetPenColor(clgreen);  
  
  setpenwidth(2);  
  rectangle(10,10,70,180);  
  x:=40;y:=40; r:=20;  
  for i:=1 to 3 do  
    begin  
      circle(x,y,r);  
      y:=y+50;  
    end;  
end.
```

Приложение к уроку 21. Тест к уроку (входной контроль).

1. Где записывается команда Uses GraphABC для подключения графического модуля?

- а) перед служебным словом Program
- б) после служебного слова Program
- в) после служебного слова Begin

2. Установите соответствие между командой и ее назначением:

- | | |
|---------------------------|--|
| 1) SetWindowSize (X, Y); | а) Залить окно цветом с |
| 2) ClearWindow (c) | б) Нарисовать отрезок от точки (x1,y1) до точки (x2,y2) |
| 3) SetPixel (x, y, c) | в) Установить размеры графического окна |
| 4) Line (x1, y1, x2, y2); | г) Установить цвет пера |
| 5) FloodFill (x, y, c) | д) Нарисовать точку с координатами (x,y) цветом с |
| 6) SetPenColor (c) | е) Закрасить замкнутую область, содержащую точку (x, y) цветом с |

Правильные ответы: 1 – б; 2 – 1в, 2а, 3д, 4б, 5е, 6г.

Урок 22. САМОСТОЯТЕЛЬНАЯ РАБОТА НА СОСТАВЛЕНИЕ ЛИНЕЙНЫХ АЛГОРИТМОВ С ИСПОЛЬЗОВАНИЕМ ГРАФИЧЕСКИХ ВОЗМОЖНОСТЕЙ ЯЗЫКА ПРОГРАММИРОВАНИЯ

Цели урока:

- проверка уровня усвоения материала предыдущих уроков, уровня сформированности требуемых знаний и умений.

Тип урока: проверка и оценка знаний и умений.

Учащиеся должны уметь составлять линейные алгоритмы с использованием графических примитивов.

Программное и методическое обеспечение урока: система программирования Pascal ABC; раздаточный материал.

Методические рекомендации к уроку

1. Проверка усвоения учебного материала. Проверку усвоения учебного материала осуществите с помощью теоретических (в форме теста) и практических уровневых индивидуальных заданий для самостоятельной работы.

Вариант 1

Теоретическая часть

Уровень 1

1. (1 балл) Для подключения библиотечных модулей служит команда:

- a) Uses
- б) GraphABC

2. (1 балл) Какая команда служит для задания размеров графического окна:
- SetWindowSize (X, Y)
 - ClearWindow

Уровень 2

3. (3 балла) Что будет выведено на экране в результате выполнения программы:

```
Program v1;  
Uses GraphABC;  
Begin  
    SetPenColor (clYellow);  
    SetBrushColor (clRed);  
    Rectangle (10, 10, 100,100);  
End.
```

Уровень 3

4. (5 баллов). Установите соответствие между командой и ее назначением:

- | | |
|----------------------------|---|
| 1) SetWindowSize (X, Y) ; | а) Нарисовать отрезок |
| 2) Line (x1, y1, x2, y2) ; | б) Нарисовать окружность |
| 3) FloodFill (x, y, c) | в) Установить цвет пера для рисования линий |
| 4) SetPenColor (c) | г) Закрасить замкнутую область |
| 5) Circle (x, y, r) | д) Установить размеры графического окна |

Практическая часть

Уровень 4

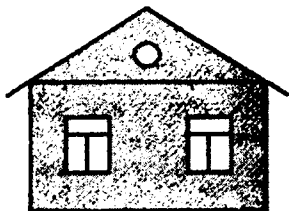
4. (7 баллов) Напишите программу, которая изобразит треугольник красного цвета; закрашенную зеленым цветом окружность с контуром синего цвета, отметит центр окружности. Толщину линии установите в 5 пикселей.

Сохраните программу на указанном учителем диске в файле с именем PR4_Фамилия.pas.

Уровень 5

5. (12 баллов) Напишите программу, которая изобразит домик (см. рисунок). Используемые цвета: желтый, черный, серый.

Сохраните программу на указанном учителем диске в файле с именем PR5_Фамилия.pas.



Вариант 2

Теоретическая часть

Уровень 1

1. (1 балл) Библиотечный модуль для работы с графикой называется:

- a) Uses
- б) GraphABC

2. (1 балл) Какая команда служит для очистки графического окна:

- a) SetWindowSize (X, Y)
- б) ClearWindow

Уровень 2

3. (3 балла) Что будет выведено на экране в результате выполнения программы:

```
Program v1;  
Uses GraphABC;  
Begin  
    SetPenColor (clGreen);  
    SetBrushColor (clBlue);  
    Circle (100,100, 70);  
End.
```

Уровень 3

4. (5 баллов). Установите соответствие между командой и ее назначением:

- | | |
|-------------------------------|--|
| 1) ClearWindow (c) | а) Установить цвет кисти для заливки фигур |
| 2) SetPixel (x, y, c) | б) Нарисовать прямоугольник |
| 3) SetBrushColor (c) | в) Вывести текст |
| 4) Rectangle (x1, y1, x2, y2) | г) Залить окно определенным цветом |
| 5) TextOut (x, y, 'текст') | д) Нарисовать точку |

Практическая часть

Уровень 4

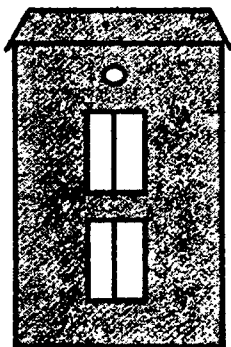
4. (7 баллов) Напишите программу, которая изобразит треугольник красного цвета; закрашенную зеленым цветом окружность с контуром синего цвета, отметит центр окружности. Толщину линии установите в 5 пикселей.

Сохраните программу на указанном учителем диске в файле с именем PR4_Фамилия.pas.

Уровень 5

5. (12 баллов) Напишите программу, которая изобразит домик (см. рисунок). Используемые цвета: желтый, черный, коричневый.

Сохраните программу на указанном учителем диске в файле с именем PR5_Фамилия.pas.



Критерий оценивания работ:

Сумма набранных баллов	Отметка
1	1
2 – 3	2
4 – 5	3
6 – 8	4
9 – 12	5
13 – 16	6
17 – 20	7
21 – 24	8
25 – 27	9
28 – 29	10

2. Домашнее задание. Повторить § 14.

УРОКИ ПО ТЕМЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ» В 8 КЛАССЕ

Урок 1. ПРАВИЛА РАБОТЫ И БЕЗОПАСНОГО ПОВЕДЕНИЯ В КОМПЬЮТЕРНОМ КЛАССЕ. ПОВТОРЕНИЕ ТЕМЫ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

Цели урока:

- повторение структуры программы, типов данных, арифметических операций, команд ввода-вывода;
- развитие логического и алгоритмического мышления;
- воспитание безопасного поведения в кабинете информатики.

Тип урока: урок повторения знаний и умений.

Учащиеся должны знать: правила безопасного поведения в компьютерном классе.

Учащиеся должны уметь: составлять линейные программы.

Программное и методическое обеспечение урока: система программирования Pascal ABC, раздаточный материал (тексты задач, прил. к уроку 1).

Методические рекомендации к уроку

1. Актуализация знаний и мотивация учащихся на изучение учебного материала. Напомнить учащимся об основных правилах безопасного поведения в кабинете информатики.

Задавая соответствующие вопросы, повторить основные сведения по теме «Основы алгоритмизации и программирования» из 6, 7 классов: понятие алгоритма, виды и способы записи алгоритмов; общий вид программы на языке Паскаль, данные (переменные и константы), типы данных, описание числовых переменных; графические возможности среды Pascal ABC.

Следует напомнить ученикам, что программа – это алгоритм, записанный на языке программирования. *Исполнителем* такого алгоритма является *компьютер*.

Предложите ученикам ответить на вопросы теста (см. приложение к уроку)

2. Самостоятельная работа.

Предложите ученикам вспомнить, как загрузить систему программирования Pascal ABC; предложите решить следующие задачи на составление и реализацию алгоритмов:

1) Составить программу, которая запрашивает два числа и находит их сумму, разность и произведение.

3) Составить программу, которая нарисует разноцветные фигуры: треугольник, квадрат и окружность.

2) Составить программу, которая запрашивает количество суток N и вычисляет, сколько в N сутках часов, минут и секунд.

3. Подведение итогов урока. Рефлексия. Обсудите решение задач, подведите итоги урока.

4. Домашнее задание: повторить виды алгоритмов, структуру программы, команды ввода, вывода, присваивания, арифметические операции и функции над числовыми данными.

Задача. Составить программу, вычисляющую модуль разности двух чисел.

Приложение к уроку 1. Тест по итогам урока

Вариант 1	Вариант 2
<p>1. Алгоритм – это:</p> <ul style="list-style-type: none">а) набор командб) последовательность действийв) описание последовательности действийг) процесс решения поставленной задачи <p>2. В линейном алгоритме:</p> <ul style="list-style-type: none">а) некоторая команда выполняется если истинно условиеб) некоторая команда выполняется несколько разв) каждая команда выполняется один раз <p>3. В алгоритме с ветвлением:</p> <ul style="list-style-type: none">а) каждая команда выполняется один разб) некоторая команда выполняется, если истинно условиев) некоторая команда выполняется несколько разг) все команды выполняются несколько раз <p>4. Исполнителем программы является:</p> <ul style="list-style-type: none">а) компьютерб) автор учебникав) ученик	<p>1. Что является алгоритмом?</p> <ul style="list-style-type: none">а) перечень устройств компьютераб) процесс приготовления бутербродовв) действия повара, делающего тортг) описание порядка действий для приготовления клея <p>2. Какая команда (команды) допустима в линейном алгоритме:</p> <ul style="list-style-type: none">а) сделать шагб) переменить до однородностив) перейти дорогу, если загорелся зеленый свет <p>3. В алгоритме с повторением:</p> <ul style="list-style-type: none">а) некоторая команда выполняется несколько разб) каждая команда выполняется один разв) некоторая команда выполняется, если истинно условиег) все команды выполняются несколько раз <p>4. Программа – это:</p> <ul style="list-style-type: none">а) один из способов самовыраженияб) инструкция по разработке алгоритмав) один из способов записи алгоритма

Вариант 1	Вариант 2
5. Какой формы записи алгоритмов не существует? а) фрагментарная (фрагменты некоторых команд) б) словесная (на родном языке) в) графическая (блок-схема) г) на языке программирования (программа)	5. Какого вида алгоритмов не существует? а) линейный алгоритм б) алгоритм с ветвлением в) алгоритм с повторением г) алгоритм с поворотом

Правильные ответы:

Вариант 1: 1 – в, 2 – в, 3 – б, 4 – а, 5 – а.

Вариант 2: 1 – г, 2 – а, 3 – а, 4 – в, 5 – г.

Урок 2. ОПЕРАТОР ЦИКЛА С ПАРАМЕТРОМ

Цели урока:

- формирование представления об операторе цикла for-to;
- развитие логического и алгоритмического мышления, познавательного интереса, внимания.

Тип урока: урок усвоения новых знаний и умений.

Учащиеся должны знать: оператор цикла for-to.

Учащиеся должны уметь: составлять простейшие циклические программы.

Программное и методическое обеспечение урока: система программирования Pascal ABC, учебное пособие «Информатика. 8 класс», § 3 (п. 3.1–3.2), раздаточный материал (тексты задач), ЭСУ Algo8.

Методические рекомендации к уроку

1. **Актуализация знаний и мотивация учащихся на изучение учебного материала.** Предлагается напомнить учащимся, что центральным понятием программирования является алгоритм. Напомнить, что программа – это алгоритм, записанный на языке программирования. Обратить внимание учащихся на то, что до сих пор они преимущественно составляли линейные программы, в которых команды выполнялись последовательно, одна за другой. Целесообразно обратить внимание учащихся на то, что в жизни имеют место случаи, когда необходимо выполнять один и тот же набор действий многократно: шагать по ступенькам, пока не поднимешься на нужный этаж, работа первоклассника в прописи, выполнение спортивных упражнений (утренняя зарядка). Предложите ученикам привести свои примеры.

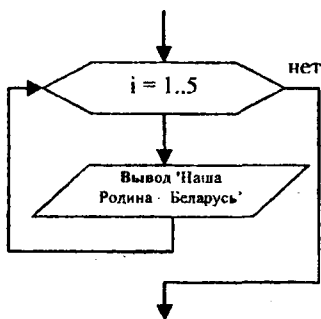
Вопрос: *Как называются алгоритмы, в которых содержатся повторяющиеся действия?* (алгоритмы с повторением).

2. Объяснение и закрепление нового материала.

Предложите решить следующую задачу, подводящую к команде цикла с параметром.

1) Составить программу, которая 5 раз выведет на экран фразу «Наша Родина – Беларусь!» (§3.2 пример 1)

Блок-схема:



Объясните ученикам, что при реализации повторяющихся действий (команд) в Паскале используются операторы цикла.

Приведите решение задачи на Паскале:

```
var i: integer;
begin
for i:=1 to 5 do writeln ('Наша Родина – Беларусь!')
end.
```

Обратите внимание учеников на необходимость описания переменной i при реализации программы. Введите понятия *заголовок оператора (цикла)*, *параметр цикла*, *тело цикла* (с. 28).

Рассмотрите по шагам порядок выполнения оператора for.

2) Составить программу, которая выведет на экран целые числа от 0 до 7. Вопрос: *Сколько раз выполниться тело цикла?*

3) Составить программу, которая выведет на экран целые числа от 5 до 9 (Пример 6).

Познакомьте учеников с записью оператора for - to в общем виде (с. 29):

```
for i:=N1 to N2 do <оператор>;
```

и блок-схемой (с. 30).

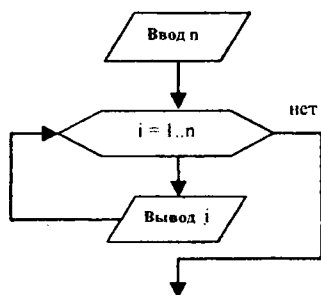
Обратите внимание учеников на то, что начальное значение параметра цикла (счетчика) должно быть меньше конечного значения. Объясните, как определить количество повторов тела цикла: $N2-N1+1$.

4) Разберите с учениками решение задачи нахождения суммы чисел от 1 до 10 (§3.1) и запись ее решения на языке Паскаль – §3.2 пример 2.

3. **Самостоятельная работа.** Для закрепления нового материала предложите ученикам самостоятельно решить следующую задачу.

5) Составьте блок-схему и напишите программу вывода на экран n первых натуральных чисел (упражнение 36).

Блок-схема к задаче:



Если останется время, решите с учениками задачу из примера 5 (с. 32).

4. Подведение итогов урока. Рефлексия.

5. Домашнее задание: §3 (до примера 3). Вопросы 1-3. Упражнения 1, 2.

Урок 3. СОСТАВНОЙ ОПЕРАТОР. СОСТАВЛЕНИЕ И РЕАЛИЗАЦИЯ АЛГОРИТМОВ С ИСПОЛЬЗОВАНИЕМ ЦИКЛА С ПАРАМЕТРОМ

Цели урока:

- закрепление материала предыдущего урока;
- формирование представления об операторе цикла for-downto, составном операторе;
- формирование навыков составления алгоритмов с повторением;
- развитие логического и алгоритмического мышления.

Тип урока: комбинированный урок.

Учащиеся должны знать: оператор цикла с параметром, составной оператор.

Учащиеся должны уметь: реализовывать и составлять программы с использованием оператора цикла с параметром.

Программное и методическое обеспечение урока: система программирования Pascal ABC, учебное пособие «Информатика. 8 класс», § 4, раздаточный материал (тексты задач, прил.к уроку), ЭСУ Algo8.

Методические рекомендации к уроку

1. Актуализация знаний и мотивация учащихся на изучение учебного материала. Задавая соответствующие вопросы, вспомните материал прошлого урока.

Целесообразно ввести составной оператор, предложив ученикам решить задачу, подводящую к его использованию.

1) Составить программу рисования пунктирной линии красного цвета

Решение:

<pre>Program Punktir; uses GraphABC; var x1,x2,y: integer; begin SetWindowSize(640,480); SetPenColor(clRed); x1:=10; x2:=20; y:=20; Line(x1,y,x2,y); x1:=x1+20; x2:=x2+20; Line(x1,y,x2,y); x1:=x1+20; x2:=x2+20; Line(x1,y,x2,y); x1:=x1+20; x2:=x2+20; Line(x1,y,x2,y); x1:=x1+20; x2:=x2+20; Line(x1,y,x2,y); x1:=x1+20; x2:=x2+20; Line(x1,y,x2,y); end.</pre>	<pre>Program Punktir; uses GraphABC; var x1,x2,y: integer; i: integer; begin SetWindowSize(640,480); SetPenColor(clRed); SetPenWidth(2); x1:=10; x2:=20; y:=20; Line(x1,y,x2,y); for i:=1 to 5 do begin x1:=x1+20; x2:=x2+20; Line(x1,y,x2,y); end; end.</pre>
--	---

После разбора линейного алгоритма решения задачи следует задать вопросы:

Какая часть программы повторяется? Сколько раз? Какой оператор можно использовать?

Предложите внести изменения в программу.

2. Объяснение и закрепление нового материала (составной оператор).

Поясните, что если в теле цикла содержится несколько команд, то их следует записывать между (операторными скобками) **begin** и **end**; – такая конструкция называется составным оператором. Расскажите ученикам об использовании составного оператора в теле цикла, дайте его запись в общем виде на языке Паскаль:

```
begin
  <Оператор1>;
  <Оператор2>;
  ...
end;
```

Предложите ученикам самостоятельно решить следующие задачи на использование составного оператора в теле цикла:

2) Составить программу, которая выводит на экран 5 первых нечетных натуральных чисел (пример 4 §3).

3) Составить блок-схему и программу, которая выводит на экран 5 первых нечетных натуральных чисел (упражнение 3а). *Какое число является наименьшим четным? Какие изменения следует внести в предыдущую программу?*

3. Объяснение нового материала (оператор **for – downto**).

4) Для подведения учащихся к использованию оператора в форме **for – downto** следует предложить решить соответствующую задачу – Пример 3 из §3.

Познакомьте учеников с записью оператора *for-downto* в общем виде (с. 29):
for i:=N1 downto N2 do <оператор>;

и блок-схемой на с. 30.

Объясните, как определить количество повторений тела цикла $(N1-N2+1)$.

4. Закрепление изученного материала.

Дайте задание ученикам самостоятельно решить задачи (прошлого урока) с использованием **for-downto**:

5) нахождение суммы чисел от 1 до 10;

6) многократный вывод некоторого сообщения.

Предложите выполнить задания теста по теме «Цикл с параметром» (см. приложение к уроку)

5. Подведение итогов урока. Рефлексия.

6. Домашнее задание: §3.1, 3.2. упражнение 4.

Приложение к уроку 3. Тест по материалам двух уроков

Вариант 1	Вариант 2
<p>1. В алгоритме с повторением:</p> <ul style="list-style-type: none"> а) некоторая команда выполняется несколько раз б) каждая команда выполняется один раз в) из двух команд выполняется только одна г) все команды выполняются несколько раз <p>2. Что называют телом цикла?</p> <ul style="list-style-type: none"> а) базовую алгоритмическую конструкцию б) последовательность повторяющихся действий в) повторение действий <p>3. Параметр цикла может иметь тип данных</p> <ul style="list-style-type: none"> а) real б) integer в) оба типа <p>4. Что сделает оператор for i:=4 to 7 do write ('Привет!');</p> <ul style="list-style-type: none"> а) три раза выведет слово «Привет!» б) четыре раза выведет слово «Привет!» в) ничего не выведет <p>5. Что сделает оператор for i:=10 to 1 do writeln (i);</p> <ul style="list-style-type: none"> а) выведет числа от 1 до 10 б) выведет числа от 10 до 1 в) ничего не выведет 	<p>1. В алгоритме с повторением:</p> <ul style="list-style-type: none"> а) некоторая команда выполняется несколько раз б) каждая команда выполняется один раз в) из двух команд выполняется только одна г) все команды выполняются несколько раз <p>2. Оператор for .. do называется</p> <ul style="list-style-type: none"> а) цикл с параметром б) цикл со счетчиком в) цикл с условием <p>3. Параметр цикла является</p> <ul style="list-style-type: none"> а) заголовком цикла б) условием каких-либо действий в) счетчиком выполнения команд тела цикла <p>4. Что сделает оператор for i:=9 downto 7 do write ('Привет!');</p> <ul style="list-style-type: none"> а) два раза выведет слово «Привет!» б) три раза выведет слово «Привет!» в) ничего не выведет <p>5. Что сделает оператор for i:=1 downto 7 do writeln (i);</p> <ul style="list-style-type: none"> а) выведет числа от 1 до 7 б) выведет числа от 7 до 1 в) ничего не выведет

Правильные ответы:

Вариант 1: 1 – а, 2 – б, 3 – б, 4 – б, 5 – в.

Вариант 2: 1 – а, 2 – а, 3 – в, 4 – б, 5 – в.

Урок 4. ОПЕРАТОР ВЕТВЛЕНИЯ

Цели урока:

- формирование представления об операторе ветвления;
- развитие логического и алгоритмического мышления, познавательного интереса.

Тип урока: урок усвоения новых знаний и умений.

Учащиеся должны знать: оператор ветвления в сокращенной и полной формах.

Учащиеся должны уметь: составлять и выполнять программы с использованием оператора ветвления.

Программное и методическое обеспечение урока: учебное пособие «Информатика. 8 класс», § 1 [2], система программирования Pascal ABC, тексты заданий, ЭСУ Algo8.

Методические рекомендации к уроку

1. Актуализация знаний и мотивация учащихся на изучение учебного материала.

Знакомство с командой ветвления следует начать с примеров из жизни. В жизни часто приходится действовать в зависимости от сложившихся обстоятельств, условий. Например, если начался дождь, то следует взять зонт на прогулку; если загорелся зеленый свет светофора, то можно переходить улицу; если поднялась температура, то нужно обратиться к врачу. Предложите ученикам привести свои примеры.

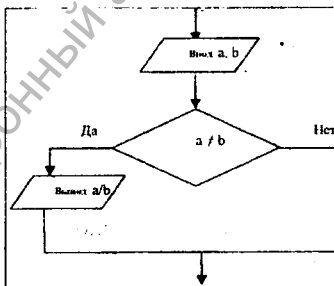
При написании программ также используются команды, которые позволяют компьютеру принимать решения о дальнейших действиях в зависимости от некоторого условия.

2. Объяснение нового материала.

Предложите ученикам решить следующую задачу, подводящую к использованию команды ветвления (в сокращенной форме).

1) Составить программу, которая вычислит частное двух введенных чисел a и b , если это возможно ($b \neq 0$).

Решение.



```
var a, b, c: real;
begin
  write('a= ');
  read(a);
  write('b= ');
  read(b);

  if b <> 0
    then write(a/b);

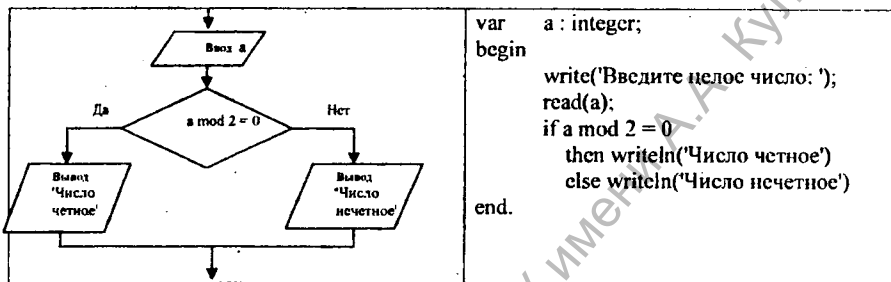
end;
```

Продемонстрируйте ученикам блок-схему и запись оператора ветвления в сокращенной форме [4, с. 10]. Расскажите о простых условиях, операциях сравнения [4, с. 7].

Задача, подводящая к использованию команды ветвления в полной форме:

2) Составить программу, которая определит, является ли введенное целое число четным или нечетным.

Решение.



Продемонстрируйте ученикам блок-схему и запись оператора ветвления в полной форме [4, с. 5, 7]. Подчеркните, что нельзя ставить «;» перед else.

3. Закрепление нового материала.

Предложите ученикам решить следующие задачи.

3) Составить программу, которая запрашивает два различных целых числа и определяет большее из них (пример 1 из § 2).

4) Составить программу, которая определит, является ли введенное целое число целым или дробным.

Решение.

```

var   r : real;
begin
write(' Введите число: ');
read(r);
if frac(r) = 0
then writeln(' Число целое ')
else writeln(' Число дробное ')
end.

```

5) Составить программу, которая определит, является ли введенное не равное нулю число положительным или отрицательным.

6) Сформулируйте условие задачи примера 5 из § 1. При решении этой задачи обратите внимание учеников на использование составного оператора.

7) Ученикам, способным к раннему усвоению материала, предложите самостоятельно изучить решение задачи на проверку знания фрагмента таблицы умножения – пример 1 из § 1. Предложите изменить программу таким образом, чтобы она проверяла знание таблицы умножения до $10 \cdot 10$.

Подсказка: использовать две вспомогательные переменные и функцию `random(10)`, известную из урока 3 (с. 32).

4. Домашнее задание: § 1.1, 1.2.-1.4, вопросы 1-5, упражнение 4 (составить блок-схему и программу).

5. Подведение итогов урока. Рефлексия.

Урок 5. ОПЕРАТОР ВЕТВЛЕНИЯ. ПРОСТЫЕ И СОСТАВНЫЕ УСЛОВИЯ

Цели урока:

- формирование представления об использовании составных условий в операторе ветвления;
- развитие логического и алгоритмического мышления, познавательного интереса.

Тип урока: урок усвоения новых знаний и умений.

Учащиеся должны знать: оператор ветвления, простые и составные условия, вложенные ветвления.

Учащиеся должны уметь: составлять и выполнять программы с использованием оператора ветвления.

Программное и методическое обеспечение урока: учебное пособие «Информатика. 8 класс», § 1, 2, система программирования Pascal ABC, тексты заданий, ЭСУ Algo8.

Методические рекомендации к уроку

1. Актуализация знаний и мотивация учащихся на изучение учебного материала.

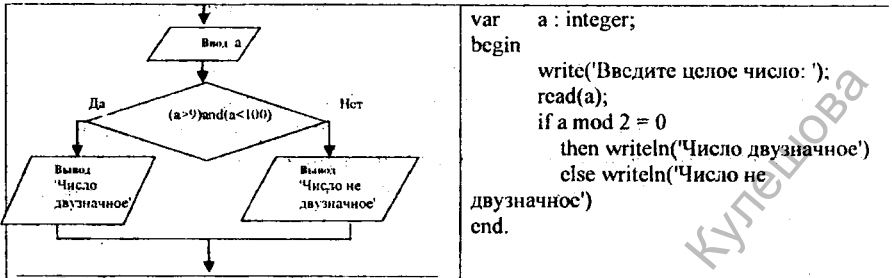
Сформулируйте задачу, подводящую к использованию составного условия:

1) Составить программу, которая определит, является ли введенное целое число двузначным (пример 2 из § 1).

2. Объяснение нового материала.

Поясните, что для того, чтобы определить, является ли число двузначным, нужно выполнить две операции сравнения: $a > 9$ и $a < 100$ и убедиться, что выполняются (истинны) оба эти условия.

Решение.



Объясните ученикам, что такое составные условия, какие логические операции используются в составных условиях, правила записи составных условий § 1.3.

3. Закрепление нового материала. Предложите решить следующие задачи.

2) Составьте программу, которая выводит число сотен в заданном целом числе, если оно трехзначное, в противном случае сообщает, что число не трехзначное.

3) Пример 2 из § 2 – узнать пору года по номеру месяца.

Какой результат выдаст программа при случайном вводе $nm=13$?

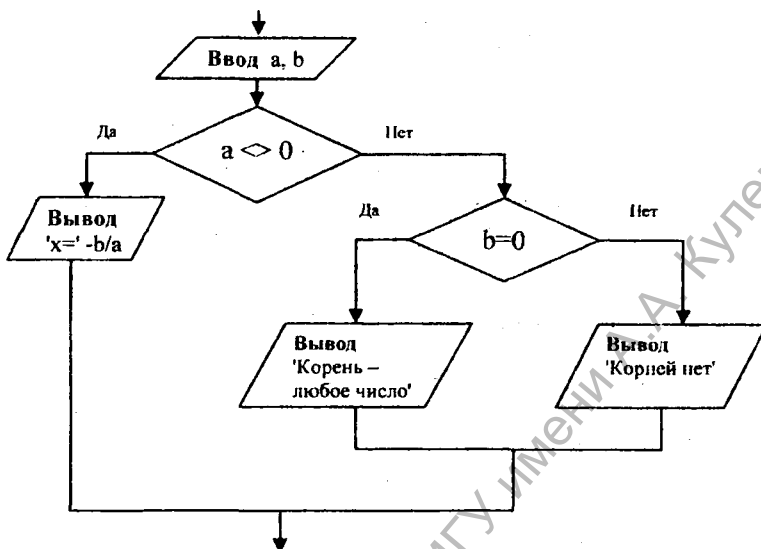
Дополните программу таким образом, чтобы при ошибочном вводе месяца ($nm < 1$ или $nm > 12$) выдавалось сообщение «Это не номер месяца».

* Составьте алгоритм для решения данной задачи, используя вложенные ветвления.

Далее можно решить с учениками задачи примеров 3–5 из §2.

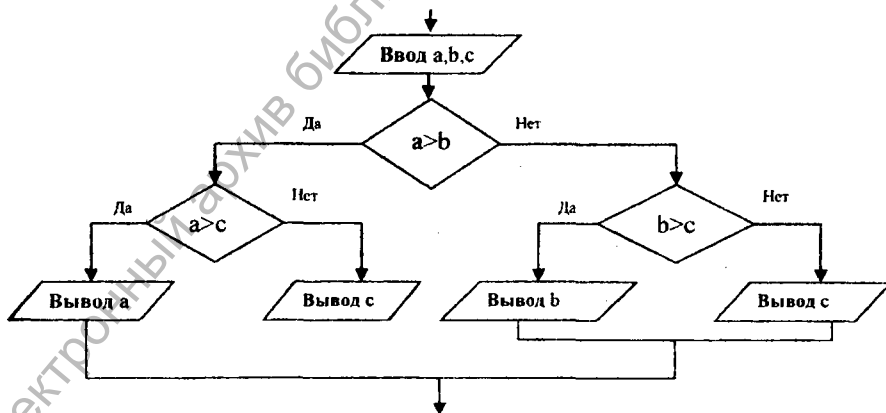
4) При решении задачи примера 3 (определить, поместиться ли круг в квадрат) следует пояснить ученикам назначение новой для них команды `SetBrushStyle(bsClear)`, которая устанавливает рисование фигур с прозрачной внутренней областью (иначе они закрашиваются цветом фона).

5) При решении задачи примера 4 целесообразно построить блок-схему:



6) При решении задачи примера 5 §2 желательно обратить внимание учеников на то, что одна и та же задача может иметь несколько решений. Можно предложить также следующее решение задачи.

Блок-схема:



Программа:

Program max3;

var a,b,c : real;

begin

writeln(' Введите рост каждого из трех учеников: ');

readln (a,b,c);

if a>b then if a>c then writeln(' Рост самого высокого - ',a)

else writeln(' Рост самого высокого - ',c)

else if b>c then writeln(' Рост самого высокого - ',b)

else writeln(' Рост самого высокого - ',c)

end.

Предложите ученикам ответить на вопросы теста (см. приложение к уроку).

4. Домашнее задание: §1,2. Упражнение 1 из § 2 – задача на кратность числу 3 (блок-схема), упражнение 5 из §2 (наименьшее из 3 чисел).

5. Подведение итогов урока. Рефлексия.

Приложение к уроку 5. Тест по материалам двух уроков.

Вариант 1	Вариант 2
1. В алгоритме с ветвлением: а) некоторая команда выполняется несколько раз б) каждая команда выполняется один раз в) из двух команд выполняется только одна	1. В алгоритме с ветвлением: а) некоторая команда выполняется несколько раз б) каждая команда выполняется один раз в) из двух команд выполняется только одна
2. Условие – это логическое выражение, которое записывается в условном операторе после слова а) if б) then в) else	2. Если условие ложно, то выполняется оператор, записанный после слова а) then б) else в) if
3. Простое условие – это а) сравнение нескольких величин б) сравнение двух выражений по величине в) сравнение двух и более выражений по величине	3. Составные условия строятся из простых с помощью а) арифметических операций +, -, *, / б) логических операций and, or, not в) операций, перечисленных в пунктах а и б
4. Укажите условия, которые верно записаны на языке Паскаль: а) $a < 0$ б) $0 < x < 10$ в) $a + 3 * b \geq 100$ г) $a = 0$ and $b = 0$ д) $a, b > 1$ е) $(a > 2 + b)$ or $(4 < b)$	4. Укажите условия, которые верно записаны на языке Паскаль: а) $5 + c < 90$ б) $b > a$ в) $a = b = c$ г) $(2 > a)$ and $(b + 3 = c)$ д) not $(a = 0)$ е) $a < 0$ or $b < 0$
5. Укажите истинные выражения а) $(2 + 3 > 5)$ and $(4 < 5)$ б) $(2 - 1 > 0)$ or $(4 > 5)$ в) not $(2 > 3 - 1)$	5. Укажите истинные выражения а) $(4 < 7)$ and $(3 + 2 > 4)$ б) $(4 * 2 < 10)$ or $(5 > 2 + 3)$ в) not $(1 + 2 > 0)$

Правильные ответы:

Вариант 1: 1 – в; 2 – а; 3 – б; 4 – а, в, е; 5 – б, в.

Вариант 2: 1 – в; 2 – б; 3 – б; 4 – а, г, д; 5 – а, б.

Урок 6. САМОСТОЯТЕЛЬНАЯ РАБОТА. СОСТАВЛЕНИЕ И РЕАЛИЗАЦИЯ АЛГОРИТМОВ С ИСПОЛЬЗОВАНИЕМ ВЕТВЛЕНИЯ

Цели урока:

- проверка уровня усвоения материала уроков по теме «Оператор ветвления»;
- развитие логического и алгоритмического мышления, самоконтроля, самооценки.

Тип урока: проверка и оценка знаний и умений.

Учащиеся должны уметь: составлять алгоритмы программ с ветвлением.

Программное и методическое обеспечение урока: система программирования Pascal ABC, учебное пособие «Информатика. 8 класс», раздаточный материал (задания по вариантам).

Методические рекомендации к уроку

1. Проверка усвоения учебного материала

Вариант 1

Теоретическая часть

Задание 1. Укажите условия, истинные при $x = -4$, $y = 2$.

а) $y = x$

б) $x = -2 * y$

в) $x \text{ div } 2 = -y$

г) $x = 2 + y$

д) $y > x$

Ответ: б, в, д.

Задание 2. Запишите на языке Паскаль следующие условия:

а) число d – положительное;

б) $20 > b > 10$;

в) число a неположительное;

г) a – отрицательное либо b – нечетное.

Ответ: а) $d > 0$;

б) $(b < 20) \text{ and } (b > 10)$;

в) $a \leq 0$;

г) $(a < 0) \text{ or } (b \text{ mod } 2 <> 0)$.

Практическая часть

Задание 3. Составьте программу, которая по заданному a исследует корни уравнения $ax=10$ и выводит на экран значение вычисленного корня либо сообщение «корней нет».

Ответ может быть таким:

```
var a:integer;  
x:real;  
begin  
write('a='); readln(a);  
if a <> 0 then writeln('x=', 10/a)  
  else writeln('корней нет');  
end.
```

Задание 4. Составьте программу, которая вводит с клавиатуры длину и ширину прямоугольника и определяет, является ли прямоугольник квадратом. Постройте данный прямоугольник (единицу длины показать 5 пикселями).

Ответ может быть таким:

```
uses crt, graphabc;  
var a,b,dl:byte;  
begin  
writeln('Введите длину прямоугольника: ');  
readln(a);  
writeln('Введите ширину прямоугольника: ');  
readln(b);  
if a=b then writeln('данный прямоугольник – квадрат');  
SetBrushColor(clSilver);  
Rectangle(20,100,20+a*5,100+b*5);  
end.
```

Вариант 2

Теоретическая часть

Задание 1. Укажите условия, истинные при $x=6$, $y=-3$.

- а) $y < x$
- б) $x = -2 * y$
- в) $x \text{ div } 2 = -y$
- г) $x = 3 + y$
- д) $x < y$

Ответ: а, б, в.

Задание 2. Напишите на языке Паскаль следующие условия:

- а) число d – отрицательное;
- б) $5 > b > 0$;

- в) число c неотрицательное;
г) a – положительное либо b – четное.

Ответ: а) $d < 0$;
б) $(b < 5)$ and $(b > 0)$;
в) $c \geq 0$;
г) $(a > 0)$ or $(b \bmod 2 = 0)$.

Практическая часть

Задание 3. Составьте программу, которая по заданному a исследует корни уравнения $ax - 5 = 0$ и выводит на экран значение вычисленного корня либо сообщение «корней нет».

Ответ может быть таким:

```
var a: integer;  
x: real;  
begin  
write('a= '), readln(a);  
if a <> 0 then writeln('x=', 5/a)  
  else writeln('корней нет');  
end.
```

Задание 4. Составьте программу, которая вводит с клавиатуры три числа a , b , c и определяет, могут ли эти числа быть длинами сторон треугольника.

Ответ может быть таким:

```
uses crt, graphabc;  
var a,b,c:byte;  
begin  
writeln('Введите числа: ');  
write('a= '); readln(a);  
write('b= '); readln(b);  
write('c= '); readln(c);  
if (a < b + c) and (b < a + c) and (c < a + b)  
  then writeln('Могут быть длинами сторон треугольника')  
  else writeln('Не могут быть длинами сторон треугольника');  
end.
```

2. Домашнее задание: §1, 2.

Урок 7. ОПЕРАТОР ЦИКЛА С ПРЕДУСЛОВИЕМ

Цели урока:

- закрепление материала предыдущих уроков;
- формирование представления об операторе цикла while;
- развитие логического и алгоритмического мышления, познавательного интереса, внимания.

Тип урока: урок усвоения новых знаний и умений.

Учащиеся должны знать: оператор цикла while.

Учащиеся должны уметь: реализовывать и составлять программы с использованием оператора While.

Программное и методическое обеспечение урока: система программирования Pascal ABC, учебное пособие «Информатика. 8 класс», § 3 (п. 3.3), раздаточный материал (тексты заданий, прил. к уроку 7), ЭСУ Algo8.

Методические рекомендации к уроку

1. Актуализация знаний и мотивация учащихся на изучение учебного материала. Предложите ученикам решить следующую задачу, подводщую к использованию цикла с условием.

1) Составьте программу, которая запрашивает целое число N и выводит на экран все нечетные натуральные числа, не превосходящие N.

Обсуждение можно организовать в форме игры: вызвать 2-х учеников, один из которых задумает число N, а второй – будет записывать на доске нечетные числа до N.

Вопросы: Пусть задумано число. Известно ли, сколько нечетных чисел нужно вывести? (нет). Обозначим выводимое число, например буквой x. Какое первое нечетное число нужно вывести? (1) Верно ли, что $x \leq N$? Выводим? (да) Какое следующее нечетное? (3) Как найти следующее нечетное число? (прибавить 2) Как узнать, что больше выводить числа не нужно? (x стало больше N) Значит, при каком условии нужно выводить число x? ($x \leq N$)

Решение

Составим словесный алгоритм:

1. Узнать число N

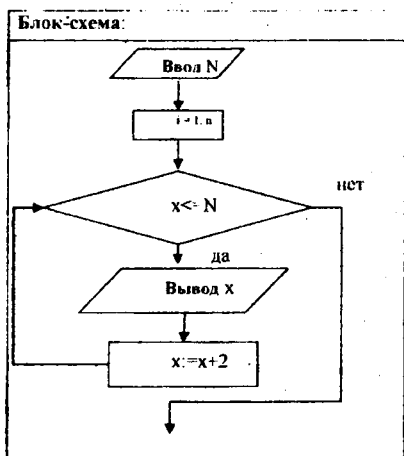
2. Взять число $x=1$

3. Пока $x \leq N$ выполнять:

3.1. вывести x

3.2. $x:=x+2$

3.3. вернуться к пункту 3 для проверки условия



Программа на Паскале:

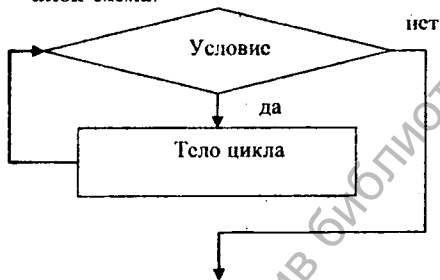
```

program nechet;
var N,x : integer;
begin
  write(' N= '); readln(N);
  x:=1;
  while x<=N do // пока...выполнять
  begin
    write(x, ' ');
    x:=x+2
  end;
  writeln;
end.
  
```

2. Объяснение нового материала.

Познакомьте учеников с оператором цикла с предварительным условием (предусловием), его блок схемой и записью на языке Паскаль (с. 34):

Блок-схема:



Запись на Паскале:

```
while <условие> do <оператор>
```

составной оператор в теле цикла:

```

while <условие> do
begin
  <Оператор1>;
  <Оператор2>;
  ...
end;
  
```

Объясните, как работает цикл с предусловием (семантику). Поясните, в каких случаях используют цикл с параметром, а в каких – с предусловием. Укажите на возможность использования составного оператора в теле цикла.

3. Закрепление изученного материала

2) Составить программу, которая выводит на экран все числа, которые не превосходят заданного числа N и кратны заданному числу k.

Решение.

```
Program kratno;
```

```
var N,x,k : integer;
```

```
{Вывести все целые числа, не превосходящие введенного N, которые делятся на k}
```



```

begin
write( 'Введите N: '); readln (N);
write( 'Введите k: '); readln (k);
x:=k;
while x<=N do
begin
writelн(x);
x:=x+k
end
end.

```

Задайте вопрос: *Можно ли использовать цикл с условием, если число повторений известно заранее?*

3) Разберите с учениками задачу из примера 7 § 3 – сумма N первых натуральных чисел.

4) Разберите (решите) с учениками задачу из примера 8 § 3 – сколько слагаемых должно быть в сумме первых натуральных четных чисел, чтобы эта сумма оказалась больше заданного N .

5) Составить программу, вычисляющую сумму чисел до ввода отрицательного числа (пример 1 из § 4):

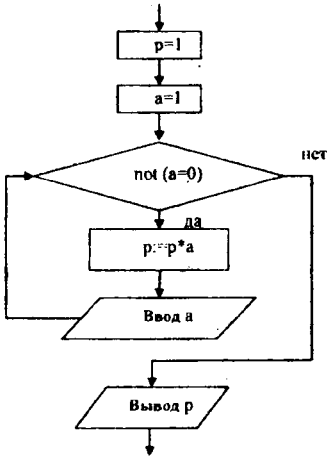
Вопросы: Известно ли заранее, сколько будет введено слагаемых? (нет) Как узнать, что больше суммировать не нужно? (введут отрицательное число) Значит какое условие нужно проверить? (введенное слагаемое неотрицательное)

Необходимо обратить внимание учеников на то, что условие $a \geq 0$ равносильно условию $\text{not}(a < 0)$. В цикле с условием можно использовать составные условия, используя логические операции **and**, **or**, **not** (Придумать простую задачу для 8 класса, где в цикле с условием будут необходимы **and** или **or** довольно сложно).

6) Составить программу, которая запрашивает числа и перемножает их до тех пор, пока не введут ноль.

Решение.

Блок-схема:



Запись на Паскале:

```
var a, p : real;
begin
  p:=1; // начальное значение произведения
  a:=1; // первый множитель
  while not (a=0) do // пока не ввели 0
  begin
    p:=p*a; // находим произведение
    write( 'следующий множитель: ');
    readln(a); // следующий множитель
  end;
  writeln( 'произведение: ', p); // результат
end.
```

Предложите ученикам выполнить тестовые задания (см. приложение к уроку).

4. Домашнее задание: §3 (до примера 9), вопросы 4-6, упражнение 5.

5. Подведение итогов урока. Рефлексия.

Приложение к уроку 7. Тест по итогам урока.

1. Можно ли использовать цикл с условием, если число повторений известно?

- а) можно
- б) нельзя

2. В цикле с предусловием **while** условие проверяется:

- а) перед выполнением тела цикла
- б) после выполнения тела цикла
- в) в процессе выполнения тела цикла

3. В цикле с предусловием тело цикла:

- а) выполняется определенное количество раз
- б) выполняется пока условие истинно
- в) выполняется пока условие ложно

4. Можно ли в цикле с предусловием записывать составные условия, используя логические операции **and**, **or**, **not**?

- а) можно
- б) нельзя

Правильные ответы: 1 – а, 2 – а, 3 – б, 4 – а.

Урок 8. СОСТАВЛЕНИЕ И РЕАЛИЗАЦИЯ АЛГОРИТМОВ С ИСПОЛЬЗОВАНИЕМ ОПЕРАТОРОВ ЦИКЛА

Цели урока:

- закрепление материала предыдущих уроков;
- формирование навыков составления алгоритмов с повторением;
- развитие логического и алгоритмического мышления.

Тип урока: урок закрепления знаний и умений.

Учащиеся должны знать: операторы цикла.

Учащиеся должны уметь: реализовывать и составлять программы с использованием операторов цикла.

Программное и методическое обеспечение урока: система программирования Pascal ABC; учебное пособие «Информатика. 8 класс», § 3, 4, раздаточный материал (тексты заданий, прил. к уроку 7), ЭСУ Algo8.

Методические рекомендации к уроку

1. Проверка знаний и закрепление умений предыдущих уроков

Обсудите с учениками выполнение домашнего задания – решение задачи из упражнения 5 §3. Вспомните материал прошлого урока: оператор цикла с предусловием. Можно задать вопросы из теста прошлого урока.

Предложите ученикам решить следующие задачи, при необходимости окажите помощь.

1) Составить программу, которая выводит на экран таблицу значений функции $y=x+1$, вычисленных в каждой точке x на отрезке $[-2; 2]$ с шагом 0,5. (Пример 8 из §4)

Вопрос: какой цикл лучше использовать: for или while?

Расскажите ученикам о том, что процесс вычисления значений функции в каждой точке на заданном отрезке с заданным шагом называется *табулированием функции*.

Закрепите понятие – выполните табулирование функции при решении следующей задачи.

2) Составить программу, которая выводит на экран таблицу значений функции $y=x^2$, вычисленных в каждой точке x на отрезке $[-7; 7]$ с шагом 1.

Вопрос: какой цикл лучше использовать: for или while?

5) Вспомните с учениками некоторые команды работы с графикой. Обсудите решение задачи примера 9 из §3. Обратите внимание учеников на размер графического окна – $640*480$.

6) Какие изменения следует внести в программу предыдущего задания таким образом, чтобы рисовались круги заданного радиуса? Предложите внести изменения.

7) Какие изменения следует внести в программу предыдущего задания таким образом, чтобы круги рисовались вдоль левого края окна? Предложите внести изменения.

2. Подведение итогов урока. Рефлексия.
3. Домашнее задание: §3, вопросы, упражнение 6 из §3.

Урок 9. СОСТАВЛЕНИЕ И РЕАЛИЗАЦИЯ АЛГОРИТМОВ С ИСПОЛЬЗОВАНИЕМ КОМАНДЫ ПОВТОРЕНИЯ

Цели урока:

- закрепление материала предыдущих уроков;
- формирование навыков составления алгоритмов с повторением;
- развитие логического и алгоритмического мышления.

Тип урока: урок закрепления знаний и умений.

Учащиеся должны знать: операторы цикла.

Учащиеся должны уметь: реализовывать и составлять программы с использованием операторов цикла.

Программное и методическое обеспечение урока: система программирования Pascal ABC, учебное пособие «Информатика. 8 класс», § 3, 4, раздаточный материал (тексты заданий), ЭСУ Algo8.

Методические рекомендации к уроку

1. Проверка знаний и закрепление умений предыдущих уроков

1) Составить программу, которая вычисляет произведение n первых натуральных чисел и выводит результат на экран (пример 2 §4).

2) Составить программу, которая находит значение выражения x^{10} , где x – целое число меньше 5.

3) Составить программу, которая находит значение выражения a^n , где a – целое число, n – натуральное (Пример 3 §4).

4) Составить программу, которая запрашивает целое число $a < 5$, находит и выводит на экран наименьшее натуральное число n , при котором $a^n \geq 50$ (пример 4 §4).

Предложите решить следующие задачи с использованием графики.

5) Составить программу, которая нарисует 3 окружности с центром в точке (320, 240) и радиусами 30, 60, 90 (цвет – по желанию)

6) Обсудите решение примера 10 из §3.

7) Задайте ученикам вопрос: какие изменения нужно внести в программу предыдущей задачи, чтобы она решала задачу упражнения 7 из §3? Предложите решить эту задачу.

2. Подведение итогов урока. Рефлексия.

3. Домашнее задание: §3, вопросы; упражнение 5 из §4.

Задача. Составить программу, которая находит и выводит на экран наименьшее натуральное число n , при котором $3^n \geq 100$.

Урок 10. САМОСТОЯТЕЛЬНАЯ РАБОТА НА СОСТАВЛЕНИЕ АЛГОРИТМОВ С ПОВТОРЕНИЕМ

Цели урока:

- проверка уровня усвоения материала предыдущих уроков, уровня сформированности требуемых умений и навыков;
- развитие логического и алгоритмического мышления.

Тип урока: проверка и оценка знаний и умений.

Учащиеся должны знать: операторы цикла.

Учащиеся должны уметь: составлять программы с использованием операторов цикла.

Программное и методическое обеспечение урока: система программирования Pascal ABC, раздаточный материал.

Методические рекомендации к уроку

1. Проверка усвоения учебного материала.

Вариант 1	Вариант 2
<p>1. Каков результат выполнения программы:</p> <pre>var i, sum : integer; begin sum:=0; for i:=1 to 4 do sum:=sum+i; writeln(sum) end.</pre> <p><i>Правильный ответ: 10</i></p>	<p>1. Каков результат выполнения программы:</p> <pre>var i, sum : integer; begin sum:=0; i:=1; while i<= 3 do begin sum:=sum+i; i:=i+1; end; writeln(sum) end.</pre> <p><i>Правильный ответ: 6</i></p>
<p>2. Составить программу, которая находит и выводит на экран значение выражения 3^n, где n – натуральное число, которое вводится с клавиатуры.</p>	<p>2. Составить программу, которая находит и выводит на экран наименьшее натуральное число n, при котором $2^n \geq 20$.</p>
<p>3. Составить программу, которая рисует по вертикали вдоль левого края графического окна ряд красных кругов радиусом R, находящихся на расстоянии 10 пикселей друг от друга.</p>	<p>3. Составить программу, которая рисует 5 окружностей голубого цвета с центром в точке (200, 200) и радиусами 30, 40, 50, 60, 70.</p>

2. Домашнее задание: §1, 3, вопросы.

Урок 11. СОСТАВЛЕНИЕ И РЕАЛИЗАЦИЯ АЛГОРИТМОВ С ИСПОЛЬЗОВАНИЕМ ВЕТВЛЕНИЯ И ПОВТОРЕНИЯ

Цели урока:

- закрепление материала предыдущих уроков;
- формирование навыков составления алгоритмов с ветвлением и повторением;
- развитие логического и алгоритмического мышления.

Тип урока: комбинированный урок закрепления изучаемого материала.

Учащиеся должны знать: операторы ветвления и повторения.

Учащиеся должны уметь: реализовывать и составлять программы с использованием операторов ветвления и повторения.

Программное и методическое обеспечение урока: система программирования Pascal ABC, учебное пособие «Информатика. 8 класс», раздаточный материал § 4, ЭСУ Algo8.

Методические рекомендации к уроку

1. Актуализация знаний и мотивация учащихся на изучение учебного материала. Анализ и оценка результатов самостоятельной работы прошлого урока. Сообщить ученикам, что существует ряд задач, решение которых требует в циклах использования операторов ветвления.

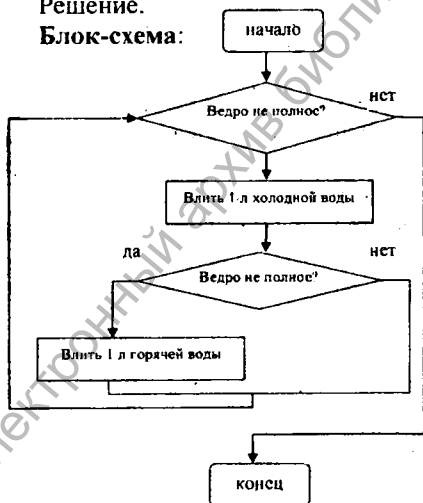
2. Составление и реализация алгоритмов.

Предложите ученикам решить следующую задачу.

1) Имеется пустое 7-литровое ведро, литровая кружка и две емкости: с горячей и холодной водой. Составить словесный алгоритм и блок-схему наполнения ведра теплой водой, поочередно добавляя горячую и холодную воду.

Решение.

Блок-схема:



Словесный алгоритм:

1. пока Ведро не полно делать:
 - 1.1. Влить 1 л холодной воды
 - 1.2. если Ведро не полно то Влить 1 л горячей воды

Обсудите с учениками решение следующей задачи.

2) Составить программу, которая запрашивает натуральные числа и находит максимальное среди введенных чисел до тех пор, пока не введут ноль, а результат – максимальное из введенных натуральных чисел выводит на экран.

Решение.

```
var n,max:integer;
begin
  write( 'Введите натуральное число: ');
  readln(n);
  max:=n;

  while n>0 do
  begin
    if n>max then max:=n;
    write( 'Введите следующее число: ');
    readln(n);
  end;
  writeln( 'Наибольшее из введенных натуральных чисел: ',max);
end.
```

Предложите ученикам решить (или обсудите решения) задач примеров 5-8 из § 4.

Ученикам, которые легко справятся с вышеуказанными заданиями, предложите также решить задачу нахождения НОД двух чисел (пример 9 § 4), попытаться самостоятельно решить или изучить решение задач из примеров 10-12 § 4.

3. Домашнее задание: §1, 3. Упражнение 4 §4.

Задача. Составить программу, которая запрашивает натуральные числа и находит минимальное среди введенных чисел до тех пор, пока не введут ноль, а результат – минимальное из введенных натуральных чисел выводит на экран.

4. Повторение и обобщение изученного материала.

Повторите основные понятия, предложите ученикам выполнить тестовые задания (см. приложение к уроку).

Приложение к уроку 11. Тест по изученному материалу

Вариант 1	Вариант 2
<p>1. В алгоритме с повторением:</p> <p>а) некоторая команда выполняется несколько раз</p> <p>б) каждая команда выполняется один раз</p> <p>в) из двух команд выполняется только одна</p> <p>2. Параметр цикла может иметь тип данных</p> <p>а) real</p> <p>б) integer</p> <p>в) оба типа</p> <p>3. Простое условие – это</p> <p>а) сравнение нескольких величин</p> <p>б) сравнение двух выражений по величине</p> <p>в) сравнение двух и более выражений по величине</p> <p>4. Укажите условия, которые верно записаны на языке Паскаль:</p> <p>а) $5+c<90$</p> <p>б) $b><a$</p> <p>в) $a=b=c$</p> <p>г) $(2>a) \text{ and } (b+3=c)$</p> <p>д) $\text{not } (a=0)$</p> <p>е) $a<0 \text{ or } b<0$</p> <p>5. Укажите истинные выражения</p> <p>а) $(4<7) \text{ and } (3+2>4)$</p> <p>б) $(4*2<10) \text{ or } (5>=2+3)$</p> <p>в) $\text{not}(1+2>0)$</p> <p>6. Условие записывается в условном операторе после слова</p> <p>а) if</p> <p>б) then</p> <p>в) else</p> <p>7. В цикле с предусловием while условие проверяется</p> <p>а) перед выполнением тела цикла</p> <p>б) после выполнения тела цикла</p> <p>в) в процессе выполнения тела цикла</p>	<p>1. Можно ли в теле цикла for изменять параметр цикла?</p> <p>а) можно</p> <p>б) нельзя</p> <p>2. В алгоритме с ветвлением:</p> <p>а) некоторая команда выполняется несколько раз</p> <p>б) каждая команда выполняется один раз</p> <p>в) из двух команд выполняется только одна</p> <p>3. Если условие ложно, то выполняется оператор, записанный после слова</p> <p>а) then</p> <p>б) else</p> <p>в) if</p> <p>4. Составные условия строятся из простых с помощью</p> <p>а) арифметических операций +, -, *, /</p> <p>б) логических операций and, or, not</p> <p>в) операций, перечисленных в пунктах а и б</p> <p>5. Укажите условия, которые верно записаны на языке Паскаль:</p> <p>а) $a<0$</p> <p>б) $0<x<10$</p> <p>в) $a+3*b>=100$</p> <p>г) $a=0 \text{ and } b=0$</p> <p>д) $a, b>1$</p> <p>е) $(a>2+b) \text{ or } (4<b)$</p> <p>6. Укажите истинные выражения</p> <p>а) $(2+3>5) \text{ and } (4<5)$</p> <p>б) $(2-1>0) \text{ or } (4>5)$</p> <p>в) $\text{not}(2>3-1)$</p> <p>7. В цикле с предусловием while тело цикла</p> <p>а) выполняется определенное количество раз</p> <p>б) выполняется пока условие истинно</p> <p>в) выполняется пока условие ложно</p>

Правильные ответы:

Вариант 1: 1 – а; 2 – б; 3 – б; 4 – а, г, д; 5 – а, б; 6 – а; 7 – а.

Вариант 2: 1 – б; 2 – в; 3 – б; 4 – б; 5 – а, в, е; 6 – б, в; 7 – б.

ЛИТЕРАТУРА

1. Кушниренко, А.Г. Информатика 7 – 9 классы / А.Г. Кушниренко, Г.В. Лисбедев, Я.П. Зайдельман. – М. : Дрофа, 2002. – 336 с. : ил.
2. Заборовский, Г. А. Информатика : учеб. пособие для 7-го кл. общеобразоват. учреждений с рус. яз. обучения / Г. А. Заборовский [и др.] ; под ред. Г. А. Заборовского. – Минск : Нар. асвета, 2009. – 159 с. : ил.
3. Заборовский, Г. А. Информатика в 7 классе : учеб.-метод. пособие для учителей учреждений общ. сред. образования с белорус. и рус. яз. обучения / Г. А. Заборовский [и др.] ; под ред. Г. А. Заборовского. – Минск : Нар. асвета, 2011. – 104 с. : ил.
4. Миняйлова, Е. Л. Информатика : учеб. пособие для 8-го кл. общеобразоват. учреждений с рус. яз. обучения / Е. Л. Миняйлова [и др.]. – Минск : Нар. асвета, 2010. – 189 с. : ил.
5. Миняйлова, Е. Л. Информатика в 8 классе : учеб.-метод. пособие для учителей учреждений общ. сред. образования с белорус., и рус. яз. обучения / Е. Л. Миняйлова [и др.]. – Минск : Адукацыя і выхаванне, 2011. – 120 с. : ил.
6. Пупцев, А. Е. Информатика : учеб. пособие для 6-го кл. общеобразоват. учреждений с белорус. и рус. яз. обучения с 11-летним сроком обучения / А. Е. Пупцев, П. П. Макарова, А. И. Лапо. – Минск : Нар. асвета, 2008. – 126 с. : ил.
7. Макарова, Н. П. Информатика в 6 классе : учеб.-метод. пособие для учителей общеобразоват. учреждений с белорус. и рус. яз. обучения / Н. П. Макарова, А. Е. Пупцев, А. И. Лапо. – Минск : Изд. центр БГУ, 2010. – 133 с. : ил.
8. Терещук, В. А. Информатика в школе: Pascal ABC в теории и на практике / В. А. Терещук, Г. А. Филипова. – Минск : Аверсэв, 2009. – 128 с. : ил. – (Школьникам, абитуриентам, учащимся).

СОДЕРЖАНИЕ

ОТ АВТОРА	3
УРОКИ ПО ТЕМЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ» В 7 КЛАССЕ	5
Урок 11. Среда программирования	5
Приложения к уроку 11	8
Урок 12. Основы языка программирования	8
Приложения к уроку 12	10
Урок 13. Организация ввода и вывода данных	14
Приложения к уроку 13	17
Урок 14. Команда присваивания. Арифметические операции и выражения	18
Приложения к уроку 14	20
Урок 15. Стандартные функции. Преобразование типов числовых данных	22
Приложение к уроку 15. Тест (входной контроль)	25
Урок 16. Решение задач на составление линейных алгоритмов	27
Приложение к уроку 16. Тест (входной контроль)	29
Урок 17. Решение задач на составление алгоритмов	31
Урок 18. Решение задач на составление алгоритмов	35
Приложение к уроку 18. Тест по итогам урока	37
Урок 19. Самостоятельная работа	38
Урок 20. Графические возможности языка программирования Pascal ABC	41
Урок 21. Решение задач на составление линейных алгоритмов с использованием графических возможностей	47
Урок 22. Самостоятельная работа на составление линейных алгоритмов с использованием графических возможностей языка программирования	49
Приложение к уроку 22. Самостоятельная работа (2 варианта)	49
УРОКИ ПО ТЕМЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ» В 8 КЛАССЕ	53
Урок 1. Правила работы и безопасного поведения в компьютерном классе. Повторение темы «Основы алгоритмизации и программирования»	53
Приложение к уроку 1. Тест по итогам урока	54
Урок 2. Оператор цикла с параметром	55

Урок 3. Составной оператор. Составление и реализация алгоритмов с использованием цикла с параметром	57
Приложение к уроку 3. Тест по материалам двух уроков	60
Урок 4. Оператор ветвления	61
Урок 5. Оператор ветвления. Простые и составные условия	63
Приложение к уроку 5. Тест по материалам двух уроков	66
Урок 6. Самостоятельная работа. Составление и реализация алгоритмов с использованием ветвления	67
Урок 7. Оператор цикла с предусловием	70
Приложение к уроку 7. Тест по итогам урока	73
Урок 8. Составление и реализация алгоритмов с использованием операторов цикла	74
Урок 9. Составление и реализация алгоритмов с использованием команды повторения	75
Урок 10. Самостоятельная работа на составление алгоритмов с повторением	76
Урок 11. Составление и реализация алгоритмов с использованием ветвления и повторения	77
Приложение к уроку 11. Тест на закрепление материала	79
ЛИТЕРАТУРА	80



Учебное издание

Баган Лариса Валерьевна

**ОСНОВЫ АЛГОРИТМИЗАЦИИ
И ПРОГРАММИРОВАНИЯ
В СРЕДНЕЙ ШКОЛЕ**

Методические материалы

Технический редактор *А.Л. Позняков*
Компьютерная верстка *А.Л. Позняков*

Подписано в печать *12.09.2012*.
Формат 60x84/16. Гарнитура Times New Roman Cyr.
Усл.-печ. л. 4,9. Уч.-изд. л. 3,8. Тираж *32* экз. Заказ № *34С*.

Учреждение образования "Могилевский государственный университет
имени А.А. Кулешова", 212022, Могилев, Космонавтов, 1
ЛИ № 02330/278 от 30.04.2004 г.

Отпечатано в отделе оперативной полиграфии
УО "МГУ им. А.А. Кулешова". 212022, Могилев, Космонавтов, 1