

Министерство образования и науки РФ
Отдел по управлению муниципальными учреждениями
администрации Юсьвинского муниципального района
МБОУ «Майкорская СОШ»

Рассмотрено на заседании ШМО Протокол № 1 от 28.08.2017 Руководитель ШМО: 	Согласовано. ЗУВР:  « 31 » 08 20 17 г.	Утверждено Приказ № 64 от 28.08.17 Директор школы:  
--	---	--

**Рабочая программа элективного курса по
программированию на языке Паскаль
для 10-11 класса**

Разработчик программы:

Накипов Павел Самигулович,
учитель информатики,
первая квалификационная категория

Содержание

Содержание.....	2
История создания языка Паскаль	3
Пояснительная записка.....	4
Тематическое планирование	6
Учебно-методический комплект.....	11
Приложение	12
Теоретические занятия	12
Практические занятия.....	50

История создания языка Паскаль

Язык программирования Паскаль был разработан профессором кафедры вычислительной техники Швейцарского Федерального института технологии Николасом Виртом в 1968 году как альтернатива существующим и все усложняющимся языкам программирования, таким, как PL/1, Algol, Fortran. Интенсивное развитие Паскаля привело к появлению уже в 1973 году его стандарта в виде пересмотренного сообщения, а число трансляторов с этого языка в 1979 году перевалило за 80.

В начале 80-х годов Паскаль еще более упрочил свои позиции с появлением трансляторов MS-Pascal и Turbo-Pascal для ПЭВМ. С этого времени Паскаль становится одним из наиболее важных и широко используемых языков программирования. Существенно то, что язык давно вышел за рамки академического и узко профессионального интереса и используется в большинстве университетов высокоразвитых стран не только как рабочий инструмент пользователя. Важнейшей особенностью Паскаля является воплощенная идея структурного программирования. Другой существенной особенностью является концепция структуры данных как одного из фундаментальных понятий.

Основные причины популярности Паскаля заключаются в следующем:

- простота языка позволяет быстро его освоить и создавать алгоритмически сложные программы;
- развитые средства представления структур данных обеспечивают удобство работы, как с числовой, так и с символьной и битовой информацией;
- наличие специальных методик создания трансляторов с Паскаля упростило их разработку и способствовало широкому распространению языка;
- оптимизирующие свойства трансляторов с Паскаля позволяют создавать эффективные программы. Это послужило одной из причин использования Паскаля в качестве языка системного программирования;
- в языке Паскаль реализуются идеи структурного программирования, что делает программу наглядной и дает хорошие возможности для разработки и отладки.

При описании достоинств Паскаля нужно сказать, что он является прототипом для языков нового поколения (среди которых следует отметить языки Ada и Modula-II). Паскаль дает очень много в понимании сущности программирования, он прививает хороший стиль программирования, тщательную разработку алгоритма. Преимущества этого языка особенно ощутимы при написании достаточно сложных и мобильных (т. е. легко переносимых на другие РС) программ.

Пояснительная записка

Изучение основ программирования связано с целым рядом умений и навыков (организация деятельности, планирование ее и т.д.), которые по праву носят общеинтеллектуальный характер и формирование которых – одна из приоритетных задач современной школы.

Очень велика роль изучения программирования для развития мышления школьников, формирования многих приемов умственной деятельности. Здесь роль информатики сродни роли математики в школьном образовании. Поэтому не использовать действительно большие возможности программирования, решения соответствующих задач для развития мышления школьников, формирования многих общеучебных, общеинтеллектуальных умений и навыков было бы, наверное, неправильно.

Изучая программирование на языке Паскаль, учащиеся прочнее усваивают основы алгоритмизации и программирования на языке Бейсик (предполагается изучение в курсе информатики двух языков программирования: один в основном курсе информатики, другой в старшей школе), приобщаются к алгоритмической культуре, познают азы профессии программиста.

Цель программы: формирование навыков программирования на языке Паскаль.

Задачи программы:

- Формирование интереса к изучению профессии, связанной с программированием.
- Формирование алгоритмической культуры.
- Формирование навыков грамотной разработки программ.
- Дать ученику возможность реализовать свой интерес к выбранному курсу.
- Приобретение знаний и навыков алгоритмизации учащимися в ее структурном варианте.
- Способствовать ученикам в освоении всевозможных методов решения задач, реализуемых на языке Паскаль.
- Развитие логического мышления учащихся.
- Развитие алгоритмического мышления учащихся.
- Углубление знаний, умений и навыков решения задач по программированию и алгоритмизации.
- Рассмотреть в 11 классе выполнение заданий ЕГЭ по программированию.

Темы программы

Алгоритмизация. Алгоритм. Свойства алгоритма. Алгоритмические структуры: линейная, ветвление, цикл, подпрограмма, массивы.

Введение в язык Паскаль. Система программирования ПаскальABC. Форма и управляющие элементы.

Операторы присваивания, ветвления, выбора, цикла, подпрограмм. Арифметические, логические, строковые выражения. Встроенные функции. Типы, имена, значения переменных.

Процедуры и функции.

Выполнение программ на компьютере.

Требования к знаниям и умениям

Учащиеся должны:

- Объяснять структуру основных алгоритмических конструкций и уметь использовать их для построения алгоритмов;
- Знать основные типы данных
- Знать операторы языка Паскаль;
- Уметь разрабатывать и записывать на языке Паскаль типовые алгоритмы;
- Уметь создавать программы с использованием языка программирования Паскаль.

Тематическое планирование в 10 классе

№ урока	Урок	Домашние задания
1	Знакомство с языком Паскаль. Алгоритм. Свойства алгоритма Презентация: Урок № 1 Знакомство с языком Паскаль Алгоритм. Свойства.ppt	№ 1 стр. 4
2	Составление линейных алгоритмов. Знакомство со средой программирования Паскаль ABC. Презентация: Урок № 2 Составление линейных алгоритмов.ppt	№ 2(№ 1-10) стр. 7
3	Составление линейных алгоритмов. Основные математические функции. (Выполнить задания для самостоятельной работы 1 – 5)	№ 3(№ 6а,б,в,г) стр. 10
4	Составление линейных алгоритмов. Ввод и вывод данных. Форматы вывода Просмотреть самостоятельно примеры 1, 2, 3 Выполнить задания для самостоятельной работы: 1 (а), 3, 4	№ 4(№ 5, 6) стр. 13
5	Составление линейных алгоритмов. Операции div и mod. Рассмотреть у доски пример про цифры трехзначного числа Выполнить задания для самостоятельной работы: № 4, 5, 11 b	№ 5(№ 3) стр. 16
6	Составление линейных алгоритмов. Операции div и mod. Рассмотреть у доски пример про цифры трехзначного числа Выполнить задания для самостоятельной работы: № 6, 7, 8, 10, 11 c, d	№ 6(№ 9) стр. 16
7	Составление линейных алгоритмов. Основные функции. M2_Блок № 3 Разобрать самую сложную задачу решаем задачи 1а, 2, 5 а	№ 7(№ 4) стр. 19
8	Решение прикладных задач. Решение задач по вариантам 1 вариант № 1, 2С, 4 А, 4С, 5 С, 6, 8, 10, 13, 12 2 вариант № 1, 2 D, 4 В, 4D, 5 D, 7, 9, 11, 14, 17	№ 8 по вариантам стр. 21 – 22
9	Решение прикладных задач. 2 урок Решение задач по вариантам 1 вариант № 1, 2С, 4 А, 4С, 5 С, 6, 8, 10, 13, 12 2 вариант № 1, 2 D, 4 В, 4D, 5 D, 7, 9, 11, 14, 17	№ 9 по вариантам стр. 21 – 22
10	Подготовка к Контрольной работе № 1 Дополнительные задачи 2, 4, 9	№ 10(№ 1, 3, 8) стр. 23 – 24
11	Решение дополнительных задач По выбору 5, 7, 10, 44, 50	
12	Контрольная работа № 1 по теме Решение линейных задач Файл Контрольная работа № 1 Решение линейных задач 2.doc	

13	<p>Условный оператор Презентация: урок № 13 14 Условный оператор.ppt На компьютерах рассмотреть задачу Две величины выбрать большее из них (Из двух числе выбрать большее) - записать в тетрадь сохранить ее к себе Сравнение возрастов - проверить работу программы Самостоятельная работа - 2 задачи</p>	Теория стр. 27 - 29 № 13 (по вариантам) стр. 29 - 30
14	<p>Логические операции и выражения Презентация: урок № 13 14 Условный оператор.ppt (2 часть) Рассмотреть и записать выражения и операции, пример двустороннего равенства и неравенства Задачи № 1 - 5(стр. 31)</p>	№ 14(№ 6) стр. 31
15	<p>Условный оператор. Структура условного оператора Задачи № 1 А, 3, 5, 9, 11 (стр. 34)</p>	№ 15(№ 8, 12) стр. 34
16	<p>Условный оператор. Простые и составные условия. Разобрать пример на стр 35 - 36 файл Заготовка 16.pas Задачи № 1 А, 3, 4, 7, 9 (стр. 37)</p>	№ 16(№ 2, 10) стр. 37
17	<p>Условный оператор. Простые и составные условия. Задачи № 5, 8, 11 в(объяснить а, б, г)Показать как работать с заштрихованной областью</p>	№ 17(№ 12) стр. 37
18	<p>Операторные скобки. Составной оператор Решаем вместе задачи 1, 3 Самостоятельно 2, 4, 6</p>	№ 18(№ 5) стр. 40
19	<p>Оператор выбора Презентация Урок № 19 Оператор выбора на Паскале.ppt слайды с 1 по 14 Задачи на стр. 43 – Дополнительные 1, 2, 3</p>	Теория, стр. 40 - 42
20	<p>Оператор выбора Решать задачи № 4, 6, 7, 8, 11, 12, 13</p>	№ 19(№ 2, 5, 9) стр. 42 - 43
21	<p>Оператор выбора Дорешить задачи № 4, 6, 7, 8, 11, 12, 13 или Дополнительно стр. 44 № 6, 7 светофор из презентации Оператор выбора</p>	Шуточный тест (показать пример) из папки Оператор выбора Шуточный тест.pas
22	<p>Решение задач Показать как работать с заштрихованной областью Показать уравнения прямой, окружности, параболы. Задачник практикум 1 стр224 № 1, 2 и 3 (с условным оператором) и № 5 (логическая переменная) - эти задачи разобрать Дать номера заданий по вариантам из МЗ_БЛОК5 объяснить когда сдавать Варианты 1 вариант 1а, 3, 6, 9, 11, 16а, 17а, 18а, 19, 23а, 24, 25б, 25в, 30а 2 вариант 1б, 4, 7, 12, 16б, 17б, 18б, 20, 23б, 25г, 25з, 26, 28а, 30б 3 вариант</p>	нет

	2, 5, 8, 10, 13, 17с, 18с, 21, 22, 25а, 25е, 25ж, 28b, 30с На следующем уроке Решение задач по вариантам	
23	Практикум по решению задач 14 задач – сдать 12 ???	По вариантам
24	Практикум по решению задач	По вариантам
25	Контрольная работа № 2 по теме Решение задач с развилкой Файл Контрольная работа № 2 Решение задач с развилкой 2.doc	
26	Циклы Презентация Урок № 26 Операторы цикла на Паскале.ppt из папки 10 класс\курс программирование + 5 примеров для презентации из папки Урок № 26 Операторы цикла на Паскале Решаем на компьютерах № 1–7 на стр. 51	№ 26(№ 8) стр. 51
27	Цикл с параметром FOR Решаем на компьютерах № 1, 2, 3а, 6, 7 а на стр. 54 - 55	№ 27(№ 7b, 8 d, 11) стр. 54
28	Цикл с параметром FOR Разобрать 8 А Решаем на компьютерах № 8а, 12, 14, 18, 21 а на стр. 54 - 55	№ 28(№ 16, 22) стр. 55
29	Циклы с предусловием WHILE и постусловием REPEAT 1 А объяснить – 2 способами Решаем на компьютерах № 1 Б, 4, 5 (два варианта решения)	№ 29(№ 6) стр. 58 (два варианта решения)
30	Циклы с предусловием WHILE и постусловием REPEAT 2 б объяснить на доске Решаем на компьютерах № 2 а, е (два варианта решения)	№ 30(№ 7) стр. 58(два варианта решения)
31	Циклы с предусловием WHILE и постусловием REPEAT Решаем на компьютерах № 1, 2, 3а, 6а, 7а на стр. 54 (два варианта решения)	№ 31(№ 8) стр. 58 (два варианта решения)
32	Циклы с предусловием WHILE и постусловием REPEAT Решаем на компьютерах № 8а, 12, 14, 18, 21 а на стр. 54 – 55 (два варианта решения)	№ 32(№ 9) стр. 58 (два варианта решения)
33	Вложенные циклы Решаем на компьютерах № 1, 2а на стр. 61 (три варианта решения)	№ 33(№ 2b) стр. 61
34	Вложенные циклы Решаем на компьютерах № 3а, 3b, 5, 6 на стр. 61 (три варианта решения)	№ 34(№ 3е) стр. 61

Тематическое планирование в 11 классе

№ урока	Урок	Домашние задания																																	
1	Операторы Паскаля (повторение). ПТБ <u>Презентация: повторение паскаль.ppt</u>																																		
2	Операторы Паскаля (повторение) Тест основы языка паскаль вариант 2																																		
3	Решение задач по вариантам <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td>1</td><td>3А</td><td>4А</td><td>6</td><td>9</td><td>12</td><td>16</td><td>18</td><td>22</td><td>21</td><td>1 вариант</td> </tr> <tr> <td>2А</td><td>3В</td><td>4В</td><td>7</td><td>11</td><td>13</td><td>15</td><td>20</td><td>23</td><td>29</td><td>2 вариант</td> </tr> <tr> <td>2В</td><td>4С</td><td>5</td><td>8</td><td>10</td><td>14</td><td>17</td><td>19</td><td>24</td><td>27</td><td>3 вариант</td> </tr> </table>	1	3А	4А	6	9	12	16	18	22	21	1 вариант	2А	3В	4В	7	11	13	15	20	23	29	2 вариант	2В	4С	5	8	10	14	17	19	24	27	3 вариант	По вариантам
1	3А	4А	6	9	12	16	18	22	21	1 вариант																									
2А	3В	4В	7	11	13	15	20	23	29	2 вариант																									
2В	4С	5	8	10	14	17	19	24	27	3 вариант																									
4	Решение задач по вариантам проверочная работа <u>Файл: 10 Циклы Проверочная 2012.doc</u> из папки 10 класс 09 контрольные и тесты	По вариантам																																	
5	Решение задач по вариантам	По вариантам																																	
6	Контрольная работа № 3 по теме Решение задач с повторениями <u>Файл: Контрольная работа № 3 Решение задач с повторениями 2.doc</u>																																		
7	Понятие массива. Ввод и вывод элементов массива. Задание массива в разделе констант <u>Презентация: Одномерные массивы.ppt</u> Решаем на компьютерах № 1 а, 1 с, 2 б, Заготовка массив1.pas и массив2.pas	№ 40 (№ 3) стр. 6																																	
8	Одномерные массивы Выполнить задания на карточках файл: Понятие массива карточки.doc из папки массивы стр 6 Решаем на компьютерах № 4 – 9	№ 41 (№ 10) стр. 6																																	
9	Действия с одномерными массивами стр 7 Решаем на компьютерах № 2,3, 4, 6 Разобрать 1 задачу	№ 42 (№ 5) стр. 7																																	
10 - 14	Поиск в массивах стр 10 – 11 Решаем на компьютерах: 1 урок – задачи 1 – 4 2 урок – задачи 6, 7, 8, 11 3 урок – задачи 12 – 15 4 урок – задачи 17 – 20 5 урок – задачи 23, 24, 26, 29	№ 43 (№ 5) стр. 10 № 44 (№ 10) стр. 10 № 45 (№ 16) стр. 10 № 46 (№ 22) стр. 11 № 47 (№ 25) стр. 11																																	
15	Замена в массивах Разобрать № 1	№ 48 (№ 8) стр. 12																																	

	стр 13 Решаем на компьютерах № 2 – 7	
16	Задачи с несколькими массивами Разобрать № 1 стр 15 Решаем на компьютерах № 1 – 5	№ 49 (№ 6) стр. 15
17	Контрольная работа № 4 по теме Одномерные массивы <u>Файл:</u> Контрольная работа № 4 Решение задач с одномерными массивами 2.doc	
18	Двумерные массивы Файлы на компьютеры: двумерный массив 1.pas Разобрать задачу 1, решать 2-4	№ 50
19	Решение задач с двумерными массивами Задачи 7 - 11	№ 51
20	Решение задач с двумерными массивами Задачи 13 - 18	№ 52
21	Решение задач с двумерными массивами Задачи 20 - 25	№ 53
22 - 26	Заполнение матриц	№ 54
23	Заполнение матриц	№ 55
24	Заполнение матриц	№ 56
25	Заполнение матриц	№ 57
26	Заполнение матриц	№ 58
27	Контрольная работа № 5 по теме Двумерные массивы Файл Контрольная работа № 5 Решение задач с двумерными массивами 2.doc	
28 - 31	Задачи по программированию из ЕГЭ	
32 - 34	Повторение базового курса информатики	

Учебно-методический комплект

1. Рапаков Г. Г., Ржеуцкая С. Ю. Программирование на языке Pascal, учебное пособие, С-Пб. БХВ-Петербург, 2012.
2. Семакин И. Г. и другие Базовый курс 7 – 9, учебник для 8 – 9 классов, М. ЛБЗ, 2012.
3. Семакин И. Г. И другие Информатика. Задачник-практикум в 2 т., М. ЛБЗ, 2012.
4. Семакин И. Г. и другие Структурированный конспект базового курса, учебное пособие для учителей, М. ЛБЗ, 2010.
5. Накипов П. С. (составитель). Язык программирования «Паскаль». Методическое пособие для учеников 10 – 11 классов, 2012.

Приложение

Разработки уроков

Теоретические занятия

Урок 1 Тема: Правила ТБ. Этапы решения задач на компьютере.

Цель: повторить правила ТБ при работе в компьютерном классе,

Определить этапы решения задач на компьютере.

1. Правила ТБ. Заполнить журнал по ТБ.

2. Новый материал

- Задача в информатике
- Этапы решения задачи
 1. Постановка задачи
 2. Математическая формализация
 3. Составление алгоритма
 4. Запись алгоритма на языке программирования
 5. Исполнение программы
 6. Анализ результатов и корректировка алгоритма, в случае необходимости.

3. Решение задач.

- 1) Составить алгоритм, как набрать из реки 7л воды, если имеются сосуды емкостью 3л и 8л.

	1	2	3	4	5	6	7	8
3л	0	3	0	3	0	2	2	3
8л	8	5	5	2	2	0	8	7

- 2) Алгоритм Евклида (НОД)

1. Если числа равны, то взять любое из них в качестве ответа, иначе продолжить выполнение алгоритма.
2. Найти большее из чисел.
3. Заменить большее число разностью большего и меньшего чисел.
4. Начать сначала.

4. Итог урока.

5. Домашнее задание. Составить алгоритм.

Некий человек должен перевезти в лодке через реку волка, козу и капусту. Каждый раз он может перевезти только либо волка, либо козу, либо капусту. На одном берегу нельзя оставить вместе козу и волка, а также козу и капусту. Составить алгоритм переправы на другой берег.

Урок 2 Тема: Основы Алгоритмизации. Алфавит языка Паскаль. Свойства алгоритма.

Цель: Ввести понятие алгоритма и его свойств;

познакомить с алфавитом языка Паскаль.

1. Опрос.

- Назовите этапы решения задач на ЭВМ.
- Прочитать алгоритм к задаче про волка, козу, капусту.

2. Новый материал.

- Свойства алгоритма
 - Дискретность
 - Точность
 - Понятность
 - Конечность

– Массовость

- Алфавит
 - Латинские прописные и строчные буквы
 - Цифры 0-9
 - Знаки + - * / \ < > ^ , . ; : ' () _
 - Служебные слова.
- Константы – числа, символы, строки, которые не изменяют своего значения в процессе выполнения программы.
CONST e=2.71 (число Эйлера)
P=3.14
- Переменные – величины, которые меняются в процессе выполнения программы. Объявляя переменную заданного типа, вы отводите в памяти место, где будет храниться ее значение.

3. Решение задач.

Исполнитель умеет:

Умножать число на 2;

Увеличивать число на 1.

Составить для этого исполнителя алгоритм получения числа 100 из единицы.

$1 \rightarrow 1 * 2 \rightarrow 2 * 2 \rightarrow 4 + 1 \rightarrow 5 * 2 \rightarrow 10 * 2 \rightarrow 20 + 1 \rightarrow 21 + 1 \rightarrow 22 + 1 \rightarrow 23 + 1 \rightarrow 24 + 1$
 $\rightarrow 25 * 2 \rightarrow 50 * 2 \rightarrow 100$

1. Исполнитель умеет из дроби a/b получать любую из дробей $(a-b)/b$, $(a+b)/b$, b/a . Как получить из дроби $1/2$ дробь $1/4$?

$1/2 \rightarrow 2/1 \rightarrow 3/1 \rightarrow 4/1 \rightarrow 1/4$

4. Итог урока.

5. Домашнее задание: Записи в тетради.

Урок 3 Тема: Типы языка Паскаль. Арифметические операции, функции.

Цель: познакомить с основными типами данных и основными операциями над данными

1. Опрос

Дайте определение алгоритма. Назовите свойства алгоритма. Привести пример алгоритма.

2. Новый материал

Типы данных в Паскале:

1. Простые
2. Структурированные
3. Ссылочные
4. Процедурные
5. Объектные

Простые типы данных

Порядковые

Целый
Логический
Символьный
Ограниченный
Перечисляемый

Вещественные

Целочисленный тип

Тип		Диапазон значений	Формат
Byte	Байтовый	0 – 255	1 байт
Shorting	Короткий	-128..127	1 байт
Integer	Целый	-32768..32867	2 байта
Word	Слово	0 – 65535	2 байта
Longint	Длинный	-2 147 483 648..2 147 483 647	4 байта

Вещественный тип

Тип		Точность	Формат
Single	С одинарной точностью	7-8 знаков	4 байта
Real	Вещественный	11-12 знаков	6 байт
Double	Двойной точности	15-16 знаков	8 байт
Extended	С двойной точностью	19-20 знаков	10 байт
Comp	Сложный	19-20 знаков	8 байт

- Арифметические выражения и операции.
Арифметическое выражение состоит из операндов, скобок, знаков операций.

Сложение	+
Вычитание	-
Умножение	*
Деление	/
Целочисленное деление $11 \text{ div } 5=2$	Div
Остаток от деления $11 \text{ mod } 5=1$	Mod

Арифметические функции

$ x $	abs(x)
x^2	sqr(x)
\sqrt{x}	sqrt(x)
π	pi
$\sin x$	sin(x)
$\cos x$	cos(x)

3. Решение задач.

Записать математические выражения на языке Паскаль:

$$\frac{\sqrt{x+1} + \sqrt{x-1}}{2\sqrt{x}}, \quad |x| + |x+1|, \quad \sqrt{x^2 + y^2}.$$

Перевести из линейной записи в обычную.

- | | | | | |
|-----------------------------|--|----------|------------|------------|
| $a/b/c;$ | $a*b/c;$ | $a+b/c;$ | $(a+b)/c;$ | $a+b/b+c;$ |
| а) $a/b**2;$ | л) $5*\arctg(x)-\arctg(y)/4;$ | | | |
| б) $a+b/c+1;$ | м) $\lg(u*(1/3)+\sqrt{v}+z);$ | | | |
| в) $1/a*b/c;$ | н) $\ln(y*(-\sqrt{\text{abs}(x)}));$ | | | |
| г) $a**b**c/2;$ | о) $\text{abs}(x**(y/x)-(y/x)**(1/3));$ | | | |
| д) $(a**b)**c/2;$ | п) $\sqrt{(x1-x2)**2+(y1-y2)**2};$ | | | |
| е) $a/b/c/d*p*q;$ | р) $\exp(\text{abs}(x-y))*(\text{tg}(z)**2+1)**x;$ | | | |
| ж) $x**y**z/a/b;$ | с) $\lg(\sqrt{\exp(x-y)+x**\text{abs}(y)+z});$ | | | |
| з) $4/3*3.14*r**3;$ | т) $\sqrt{\exp(a*x)*\sin(x)**n}/\cos(x)**2;$ | | | |
| и) $b/\sqrt{a*a+b};$ | у) $\sqrt{\sin(\arctg(u))**2+\text{abs}(\cos(v))};$ | | | |

к) $d*c/2/R+a**3$;

ф) $abs(cos(x)+cos(y))*(1+sin(y)**2)$;

4. Итог урока.

5. Домашнее задание: $\frac{1}{1+\frac{a+b}{2}}, (a+b)/(b+c)$.

Урок 4 Тема: Структура программы на языке Паскаль. Операторы ввода, вывода и присваивания

Цель: определить структуру программы, познакомить с операторами ввода, вывода, присваивания.

1. Опрос

Перечислить основные типы данных

Перечислить арифметические операции

На доске вычислить $25 \div 4$, $41 \bmod 3$

2. Новый материал

- Основные операторы.

- *Read* (a_1, a_2, \dots, a_n) – оператор ввода данных

Readln (a_1, a_2, \dots, a_n)

Readln – переход на новую строку при вводе данных, используют, если необходимо задержать исполнение программы до нажатия *Enter*.

- *Write* (a_1, a_2, \dots, a_n) – оператор вывода данных

Writeln (a_1, a_2, \dots, a_n)

Writeln – переход на следующую строку.

- $:=$ оператор присваивания. Вычисляет значение выражения, стоящего справа и присваивает переменной стоящей слева. Нельзя с помощью одного оператора присваивания присвоить нескольким переменным одно и то же значение. Например: нельзя записать $i:=j:=k:=0$; необходимо: $i:=0$; $j:=0$; $k:=0$.

- Структура программы.

program имя программы; {заголовок программы}

uses имя модуля1,..; {раздел библиографии}

label имя метки 1,..; {раздел меток}

const имя константы = значение константы; {раздел констант}

type ; {раздел типов}

var ; {раздел переменных}

begin

операторы

end.

3. Решение задач.

Составить программу вычисления периметра и площади прямоугольника.

```
Program z1;
```

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

```
Begin
```

```
  Writeln('b');
```

```
  Readln(a, b);
```

```
  P:= 2*(a+b);
```

```
  S:=a*b;
```

```
  Writeln(p, s);
```

```
  Readln;
```

End.

4. Итог урока.

5. Домашнее задание: Записи в тетради.

Урок 5 – 7 Тема: Простейшие программы на языке Паскаль.

Цель: : Научить составлять линейные алгоритмы

1. Письменный опрос

I вариант

1. Назовите этапы решения задач с помощью ЭВМ.
2. Перечислите стандартные функции на Паскале
3. Запишите выражение на языке

Паскаль $\frac{\sin x + \sqrt{y}}{3,5xy}$

II вариант

1. Алгоритм и его свойства.
2. Перечислите основные операторы на Паскале
3. Запишите выражение на языке

Паскаль $\frac{|5 + \cos x|}{7x + \sqrt{y}}$

2. Решение задач.

1. Вычислить сторону ромба, если известны две его диагонали.

```
Program z2;
```

```
Var a, d1, d2: real;
```

```
Begin
```

```
Writeln ('Введи диагонали');
```

```
Readln (d1, d2);
```

```
a:= sqrt(sqr(d1/2)+ sqr(d2/2));
```

```
Writeln ('a=',a);
```

```
Readln;
```

```
End.
```

2. Вычислить периметр и площадь ромба, у которого сторона и меньшая диагональ

равны a . ($S = \frac{a^2 \sqrt{3}}{2}$)

3. Итог урока.

4. **Домашнее задание:** Вычислить длину окружности и площадь круга по заданному радиусу.

Вычислить площадь прямоугольного треугольника по его катетам.

Составить линейный алгоритм на языке Паскаль, для вычисления по формуле: $y = x + 4a - 7$

В программе предусмотреть ввод переменных и результат.

Урок 8 Тема: Решение задач на составление программ

Цель: Научить составлять линейные алгоритмы, реализовывать их на компьютере,

Научить тестировать программу, находить ошибки, корректировать программу.

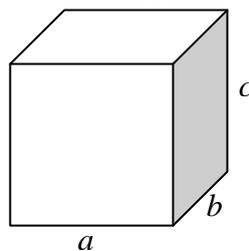
1. Домашнее задание

2. Решение задач.

- 1) Даны два целых числа. Найти их среднее арифметическое и среднее геометрическое.

- 2) Даны длины сторон прямоугольного параллелепипеда. Найти объем и площадь боковой поверхности.

$$V = abc$$
$$S = 2(a+b)c$$



- 3) Найти сумму кубов корней уравнения $x^2 - ax - 2 = 0$.

$$x_1^3 + x_2^3 = (x_1 + x_2)(x_1^2 - x_1x_2 + x_2^2) = a((x_1 + x_2)^2 - 3x_1x_2) = a(a^2 + 6)$$

3. Итог урока.

4. Домашнее задание: Задача

Вычислить периметр прямоугольного треугольника по его катетам.

Урок 9 Тема: Зачетная работа по теме «Основы языка Паскаль»

Цель: Контроль знаний учащихся

Вариант № 1

1. Назначение программирования – это ...

- а) разработка программ управления компьютером;
- б) разработка программ для решения задач;
- в) разработка программ управлением операционной системой;
- г) создание и редактирование задач на паскале;

2. Фиксированная система обозначений для описания алгоритмов – это

- а) система программирования;
- б) язык программирования;
- в) система команд исполнителя;
- г) алфавит языка;

3. Выберите из списка правильные языки программирования:

- а) паскаль, египетский;
- б) фортран, бейсик;
- в) си, формулы, паскаль;
- г) английский, французский;

4. Команды: присваивание, ввод, вывод, цикл относятся к ...

- а) простым командам;
- б) командам присваивания;
- в) системе данных;
- г) системе команд;

5. Выберите из списка правильное описание команды присваивания:

- а) $x := \text{варежка} + 1$;
- б) $\text{Rez} = a + 5$;
- в) $y := 1$;
- г) $N := x - 1$;

6. Команда ввода информации на Паскале выглядит:

- а) `Reedln`;
- б) `Readln`;
- в) `Write`;
- г) `Writelene`;

7. Команда вывода информации на Паскале выглядит:

- а) `Reedln`;
- б) `Readln`;
- в) `Write`;
- г) `Writelene`;

8. Результат решения задачи сообщается пользователю путём:

- а) команды вывода;
- б) команды ввода;
- в) вывода на экран;
- г) ввода на экран;

9. Результат решения задачи выводится на:

- а) внешнюю память;
- б) экран;
- в) принтер;
- г) всё перечисленное выше;

10. Данные в программе сохраняются ...

- а) до конца выполнения программы;
- б) до их пересваивания;
- в) после выполнения программы;
- г) во внешней памяти;

11. Отметьте целые типы

- а) integer, real
- б) integer, longint
- в) single, double
- г) extended, string

12. Отметьте логический тип

- а) integer
- б) word
- в) string
- г) boolean

13. Отметьте вещественные типы

- а) extended, string
- б) real, byte
- в) single, extended
- г) string, Boolean

14. Значения True, False относятся к типу

- а) integer
- б) word
- в) string
- г) boolean

15. Выберите из списка правильное описание переменных

- а) x, y : integer;
- б) Var x y : integer;
- в) Var x, y : integer;
- г) Var x, y integer;

16. Основная программа на языке Паскаль заключена между:

- а) Program и end
- б) Begin и end
- в) Var и end
- г) Readln и writeln

17. Раздел описания модулей начинается со слова:

- а) Program
- б) Uses
- в) Label
- г) Begin

18. Операция DIV обозначает

- а) деление
- б) целочисленное деление
- в) вещественное деление
- г) остаток от деления

19. Операция MOD обозначает

- а) остаток от целочисленного деления
- б) остаток от деления
- в) деление
- г) целочисленное деление

20. Блок обозначает

- а) вывод данных

- б) ввод данных
- в) какое-то вычисление
- г) условие

21. Блок  обозначает

- а) обращение к вспомогательному алгоритму
- б) ввод данных
- в) какое-то вычисление
- г) условие

22. Модуль x на Паскале выглядит так

- а) $|x|$
- б) `modul(x)`
- в) `abs(x)`
- г) `exp(x)`

23. Выражение $\frac{a+1}{\sqrt{2}}$ на языке Паскаль записывается так

- а) `a + 1 / sqrt 2`
- б) `a + 1 / (sqrt 2)`
- в) `(a + 1) / sqrt 2`
- г) `(a + 1) / sqrt (2)`

24. Результат выражений $a*b/c$ и $(a*b)/c$ будет

- а) разным
- б) одинаковым
- в) не знаю ответа

25. В арифметические выражения могут входить

- а) команды операционной системы
- б) машинные коды
- в) указатели
- г) переменные

26. Что можно отнести к названию типа переменных?

- а) переменные
- б) целые
- в) числовые
- г) указатели

27. Чему равна переменная K после следующих действий

$$S := 50;$$

$$K := S/2 - 25;$$

$$S := S - 10;$$

$$K := K + S$$

- а) 40
- б) 30
- в) -40
- г) 0

28. Найдите правильно записанную в линейной форме формулу: $\frac{\sqrt{\frac{1}{x} + x^3}}{2x}$

- а) $(\sqrt{1/x + x*x*x}) / (2 * x)$
- б) $(\sqrt{1/x + x*x*x}) / 2 * x$
- в) $\text{qrt } 1/x + x*x*x) / (2 x)$
- г) $\text{sqrt } (1/x + x*x*x) / (2 * x)$

29. В каких единицах задается аргумент для SIN

- а) в градусах
- б) в радианах
- в) в секундах
- г) в любых

30. Чему равно значение выражения:

$$y := 12 \bmod 4$$

- а) 12
- б) 4
- в) 3
- г) 0

31. Чему равно значение выражения:

44. Если две логические переменные X и Y ложны, то какое значение примет выражение

$p := \text{not} (X \text{ and } \text{not}(Y))$

- a) True
- б) False
- в) not
- г) end

45. Определите результат работы программы:

Writeln(' 5 + 5 = ', 5 + 5)

- a) 10 = 10
- б) 5 + 5 = 10
- в) 20
- г) 5 + 5 = 5 + 5

Таблица ответов вариант 1

Вопрос	Ответ	Вопрос	Ответ
1.	А	24	Б
2.	Г	25	Г
3.	Б	26	Б
4.	Г	27	А
5.	В	28	Г
6.	Б	29	Б
7.	В	30	Г
8.	А	31	Б
9.	Г	32	В
10.	А	33	А
11.	Б	34	В
12.	Г	35	-
13.	В	36	В
14.	Г	37	Г
15.	В	38	А
16.	Б	39	Г
17.	Б	40	Б
18.	Б	41	Г
19.	А	42	Г
20.	В	43	Б
21.	Б	44	А
22.	В	45	Б
23.	Г		

Критерии оценки:

- 5 – от 37 до 45
- 4 –от 29 до 36
- 3 – от 22 до 28
- 2 – менее 22

Урок 10 Тема: Формулы

Цель: Научится составлять формулы на языке Паскаль

1. Анализ зачетной работы
2. Решение задач.

Значения переменных x, b, c, d, e, k вводятся с клавиатуры. Составить программы для вычисления следующих выражений:

1) $y = |x| + x^4$

- 2) $y = |x| + 4x^3 - 7x^2$
- 3) $y = |x - 2| + 3x^8$
- 4) $y = 6b^2 + |b - 3|^3 - 15$
- 5) $y = (3x^3 + 18x^2)x + 12x^2 - 5$
- 6) $y = (d + c + b)e^{-5}k - 1$
- 7) $y = 3c^3 + |c^2 - 4c + 7| - 5c$
- 8) $y = |x + 4| - |x^2 - 3x + 6|$

3. Итог урока.

4. Домашнее задание: Задача

Составить линейный алгоритм на языке Паскаль, для вычисления по формуле: $x = 4y + 2c - 9$. В программе предусмотреть ввод переменных и результат.

Урок 11 Тема: Логические выражения

Цель: Научится составлять логические выражения

1. Опрос

1. Устная работа

- Найти ошибки:

a) $A := 25 + B;$

b) $4 - B := A;$

c) $ABS(A) := B;$

d) $4 := B;$

e) $B := SQR(A);$

f) $B := A;$

Ошибки в задании b), c), d).

- Определить значения x, y.

$x := 1.2; y := x * 2;$

$y := y - 5 * x;$

$x := ABS(y);$

$Writeln(x, y);$

Ответ: $x = 3.6; y = -3.6.$

- Определить значения x, y.

$a := -7; b = a / 2;$

$a := (a + b) / 2;$

$b := a - b;$

$Writeln(a, b);$

Ответ: $a = -5.25; b = -1.75.$

2. Новый материал

Логические выражения и операции.

Результатом выполнения логического (булевого) выражения является логическое значение **TRUE** или **FALSE**. Операндами служат данные только булевого типа.

Простейшими видами логических выражений являются следующие: логическая константа, логическая переменная, элемент массива логического типа, логическая функция, выражение отношения.

Другие логические выражения строятся из вышеперечисленных путем применения логических операций и круглых скобок.

Паскаль позволяет обрабатывать данные не только числового типа, но и других типов, например логического. Для обозначения логического типа используется ключевое слово *boolean*. Булевы (логические) переменные могут иметь одно из двух значений: *true* (истина)

или *false* (ложь). Булевский тип является порядковым типом и упорядочен так, что *false* < *true*. По сравнению с типом *real*, который допускает миллионы различных значений, может показаться, что тип *boolean* имеет ограниченную область применения. Однако, как это ни странно, именно ограниченность диапазона значений булевых переменных придает булевым выражениям их значимость.

Булевы выражения могут принимать несколько различных форм. Во-первых, они могут быть просто константами *true* (истина) или *false* (ложь). Оператор присваивания, использующий эту форму, аналогичен арифметическому оператору. Например, пусть переменные *a* и *b* логического типа:

```
var a,b:boolean;
```

тогда можно написать операторы:

```
a:=true;
```

```
b:=false.
```

Булевы выражения можно использовать для проверки отношений между двумя переменными: $a > b, c \leq d, k = m$ и т.д. Таким образом, условие в условном операторе является выражением логического типа. Кроме того, булевы выражения могут конструироваться с помощью булевых операций. Эти операции образуют инструментальный фундамент булевой логики, алгебры логики, разработанной в XIX в. математиком Джорджем Булем. Рассмотрим три основные булевы операции.

Операция *and* – логическое пересечение (умножение, операция "и"). Выражение *a and b* дает значение *true* только в том случае, если *a* и *b* имеют значения *true*, в остальных случаях – *false*:

$true \text{ and } true = true$

$true \text{ and } false = false$

$false \text{ and } false = false$

Операция *or* – логическое сложение (объединение, операция "или"). Выражение *a or b* дает значение *false* в том и только в том случае, если *a* и *b* имеют значения *false*, в остальных случаях – результат *true*:

$true \text{ or } true = true$

$true \text{ or } false = true$

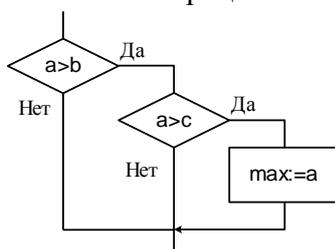
$false \text{ or } false = false$

Операция *not* – отрицание (операция "не"). Выражение *not a* имеет значение, противоположное значению *a*:

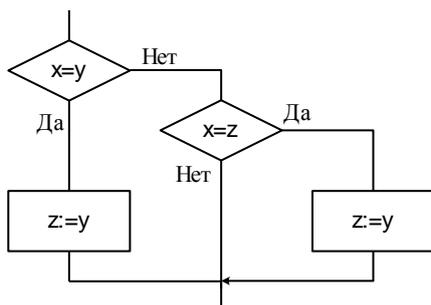
$not \ true = \ false$

$not \ false = \ true$

Эти операции полезны, если нужно проверить сложное условие:



$if \ (a > b) \ and \ (a > c)$
 $then \ max := a;$



$if \ (x=y) \ or \ (x=z)$
 $then \ z := y;$

3. Решение задач

Составить логические выражения из задачника Семакина

4. Итог урока.

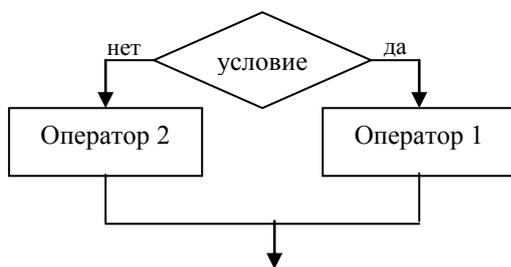
5. Домашнее задание: Конспект из тетради.

Урок 12 – 13 Тема: Условный оператор

Цель: Познакомить с оператором условного перехода, правилами записи условий.

1. Новый материал

Если в программе после выполнения очередного оператора надо выполнить не следующий по порядку, а другой, помеченный для этого меткой, используется оператор безусловного перехода **GOTO**. Он осуществляет переход к инструкции, перед которой стоит метка, объявленная в разделе меток Label.



Условный оператор

Условный оператор имеет *полную* и *краткую* формы.

Полная форма условного оператора выглядит следующим образом:

```
if условие then оператор1  
else оператор2;
```

В качестве условия указывается операция отношения.

Сложные условия составляются из простых с помощью логических операций *and*, *or*. Если условие

оказывается истинным, то выполняется оператор1, в противном случае выполняется оператор2.

Краткая форма условного оператора имеет вид:

```
if условие then оператор;
```

Если условие оказывается истинным, то выполняется оператор, в противном случае происходит переход к следующему оператору программы.

Если в качестве оператора выполняются несколько операторов, то они заключаются в операторные скобки `begin...end`.

3. Решение задач.

Задача 1. Даны два числа. Найти большее из них.

Задача 2. Найти большее из трех чисел.

```
program z1;  
uses crt;  
var a,b,c,max:real;  
begin  
  writeln ('Введи числа');  
  readln(a,b,c);  
  if a>b then max:=a else max:=b;  
  if c>max then max:=c;  
  writeln(max);  
end.
```

Задача 3. Принадлежит ли число n интервалу (a, b) .

Задача 4. Вычислить значение функции: $y = \begin{cases} x^2 + 5, & x > 3 \\ x - 8, & x \leq 3 \end{cases}$.

4. Итог урока.

5. Домашнее задание.

1. Могут ли углы α , β , γ быть углами одного треугольника.

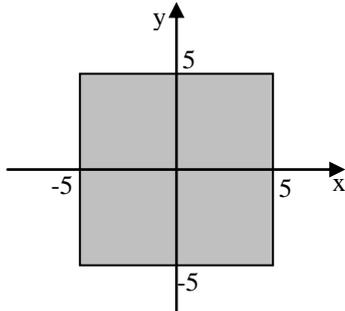
2. Даны точки $M(1,2)$ и $P(3, a)$. Лежит ли отрезок MP в I четверти.

Дополнительно.

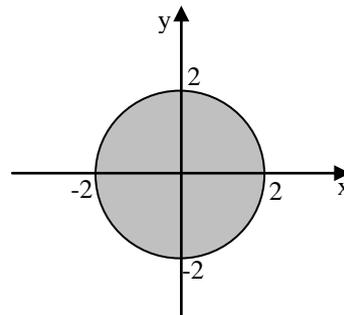
1. Составьте программу, которая уменьшает первое число в пять раз, если оно больше второго по абсолютной величине.

2. По заданному значению x вычислить значение $y = \begin{cases} x^3 - 4, & x \leq -1; \\ \cos x, & -1 < x \leq 5; \\ \frac{1}{x-5}, & x > 5. \end{cases}$

3. Определить принадлежит ли точка (x, y) фигуре



$$F = \begin{cases} |x| \leq 5; \\ |y| \leq 5. \end{cases}$$



$$x^2 + y^2 \leq 2^2.$$

4. Дано двузначное число. Определить кратна ли трем сумма его цифр.

Урок 14 Тема: Оператор выбора

Цель: Познакомить учащихся с форматом оператора выбора; добиваться понимания учениками особенностей применения оператора.

1. Проверка домашнего задания.

1. Выяснить оканчивается ли заданное число нулем.
2. Выяснить взаимное расположение прямой OX и точки (a,b) .

2. Самостоятельная работа.

I вариант	II вариант
1. Дано натуральное число. Верно ли, что оно заканчивается нечетной цифрой.	1. Дано натуральное число. Верно ли, что оно заканчивается четной цифрой.
2. Известен рост трех человек. Определить, одинаков ли их рост.	2. Известны длины трех сторон треугольника. Является ли треугольник разносторонним.

3. Новый материал

Оператор выбора

Не рекомендуют использовать многократно вложенный друг в друга условный оператор *if*. Если необходимо проверить много условий и в зависимости от них выполнять те или иные действия, то используют оператор выбора *case*.

Оператор выбора выполняет одно действие из нескольких в зависимости от значения некоторого выражения, называемого *переключателем (селектором)*. Он имеет следующий вид:

```
case переключатель of  
  список выбора 1: оператор1;  
  ...  
  список выбора N: операторN;  
  else оператор0  
end;
```

Переключатель представляет собой выражение порядкового типа (целого, символьного, перечислимого или интервального), а списки выбора содержат константы совместимого типа. Как и в операторе **if**, ветка **else** может отсутствовать.

Оператор **case** работает следующим образом. Если в одном из списков выбора найдено текущее значение переключателя, то выполняется оператор, соответствующий данному списку. Если же значение переключателя не найдено ни в одном списке, то выполняется оператор по ветке **else** или, если ветка **else** отсутствует, оператор **case** не выполняет никаких действий.

Список выбора состоит либо из одной константы, либо из диапазона значений вида *a..b* (константа *a* должна быть меньше константы *b*); можно также перечислить несколько констант или диапазонов через запятую:

```
case DayOfWeek of  
  1..5: writeln('Будний день');  
  6,7: writeln('Выходной день');  
end;
```

3. Решение задач.

Задача 1. Составить программу, которая в зависимости от порядкового номера дня недели выводит на экран его название.

```
Program z1;  
Var n: integer;  
Begin  
  Read (n);  
  Case n of           или n mod 7  
    1: Write ('понедельник');  
    2: Write ('вторник');  
    3: Write ('среда');  
    4: Write ('четверг');  
    5: Write ('пятница');  
    6: Write ('суббота');  
    7(0): Write ('воскресенье');  
  end;  
end.
```

Задача 2. Выполнить заданную арифметическую операцию с двумя числами.

```
program z2;  
uses crt;  
var op:char;  
a,b,s:real;  
begin  
  write('введи числа '); readln(a,b);  
  write('введи знак операции '); readln(op);  
  case op of  
    '+': s:=a+b;
```

```

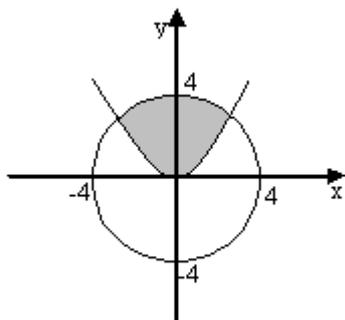
': s:=a-b;
*': s:=a*b;
/': s:=a/b;
end;
write(a, op, b,'=',s);
end.

```

4. Итог урока.

5. Домашнее задание:

1. Принадлежит ли точка фигуре.



Урок 15 – 16 Тема: Решение задач с использованием оператора выбора

Цель: Научить учащихся составлять программы на использование оператора условия и выбора.

1. Проверка домашнего задания

2. Решение задач.

Задача 3. По количеству набранных при тестировании баллов выставить соответствующую отметку. Пусть введена 100-бальная система и заданы следующие критерии оценивания:

Сумма баллов	90-100	80-89	65-79	<65
Отметка	5	4	3	2

```

program z2;
uses crt;
var s : integer;
    b: 2..5;
begin
write('Введите балл '); readln(s);
case s of
90..100: b:=5;
80..89: b:=4;
65..79: b:=3;
else b:=2
end;
write('Отметка ', b);
end.

```

Задача 4. В старояпонском календаре принят 60-летний цикл, состоящий из пяти 12-летних подциклов. Подциклы обозначались названиями цветов: зеленый, красный, желтый, белый и черный. Внутри каждого подцикла годы носили названия животных: крыса, корова, тигр,

заяц, дракон, змея, лошадь, овца, обезьяна, курица, собака и свинья. Например: 1984 год – год начала очередного цикла – назывался Годом Зеленой Крысы. Составить программу, которая по заданному номеру года нашей эры n напечатает его название по старояпонскому календарю. Рассмотреть два случая:

- а) Значение $n \geq 1984$;
- б) Значение n может быть любым натуральным числом.

```

program z2;
uses crt;
var n : integer;
begin
write('введи год '); readln(n);
n:=(n-1984) mod 12;
case n of
0: write('крыса');
1: write('корова');
2: write('тигр');
3: write('заяц');
4: write('дракон');
5: write('змея');
6: write('лошадь');
7: write('овца');
8: write('обезьяна');
9: write('курица');
10: write('собака');
11: write('свинья');
end;
end.

```

```

program z2;
uses crt;
var n,m : integer;
begin
write(' введи год '); readln(n);
if n>= 1984 then begin m:=(n-1984) mod 5;
n:=(n-1984) mod 12;end
else begin m:= 5-(1984-n) mod 5;
n:= 12-(1984-n) mod 12; end;
case m of
0: write('зеленый ');
1: write('красный ');
2: write('желтый ');
3: write('белый ');
4: write('черный ');
end;
case n of
0: write('крыса ');
1: write('корова ');
2: write('тигр ');
3: write('заяц ');
4: write('дракон ');
5: write('змея ');
6: write('лошадь');
7: write('овца');
8: write('обезьяна');
9: write('курица');
10: write('собака');
11: write('свинья');
end;
end.

```

3. Итог урока.

4. Домашнее задание: 2. Вычислить значение функции. $y = \begin{cases} x-12, & x > 0 \\ 5, & x = 0 \\ x^2, & x < 0 \end{cases}$.

Урок 17 Тема: Зачетная работа по теме «Условия»

Цель: Контроль знаний учащихся

Вариант 1

1. Нарисовать блок схему полного условного оператора.
2. Нарисовать блок-схему к задаче:
Определить, является ли введённое число отрицательным.
3. Какое значение будет иметь переменная d после выполнения операторов
 - 1) if $a > b$
 - 2) if $a > b$

```

then
  d := a
else
  d := b;
d := d * 10;

```

```

then
  d := b;
d := d * 10;

```

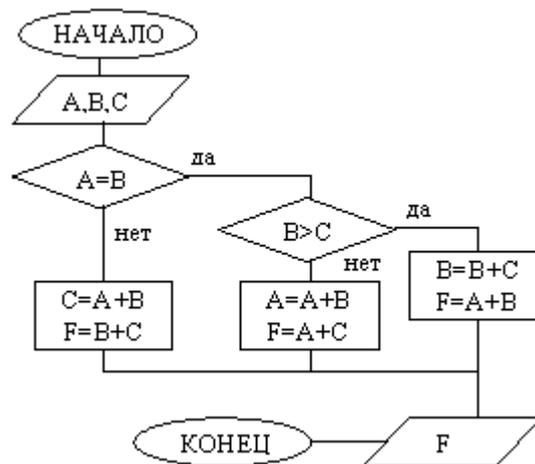
3) if $a > 2$ then $d := 2$ else if $b > 2$ then $d := 1$;
 $d := d * 10$;

при $a = 3, b = 1, d = 0$

- 1) _____
- 2) _____
- 3) _____

4. Перечислите основные логические выражения
5. Напишите общий вид оператора выбора:

6. Вычисленное по блок-схеме значение переменной F для входных данных 2,2,3 равно
 1) 5 2) 6 3) 7 4) 9 5) 8



7. Определить результат операции при значении $x = 12$
 if $x < 10$ then $x := 10$ else if $x < 15$ $x := 15$ else $x := 20$;
8. Определить результат при $m=2, n = -4$
 if $(m > n)$ then if $(n > 0)$ or $(m > 0)$ then Write(m) else Write(n);
9. Каково будет значение выражения:
 $(3 > 5)$ or $(8 < -11)$ or $(30 > 6)$
10. Написать программу для нахождения суммы всех вводимых с клавиатуры чисел, больших заданного числа d.

Урок 18 Тема: Подпрограммы

Цель: Познакомить со способом решения задач с помощью подпрограмм.

1. Анализ зачетной работы
2. Новый материал.

Подпрограммы используются:

1. Для экономии памяти, когда некоторая последовательность действий встречается неоднократно на разных участках программы. Каждая подпрограмма существует в ед. экземпляре, но обращаться к ней можно неоднократно из разных точек программы.
2. В сложных программах при модульном проектировании.

Подпрограммы делятся на процедуры и функции

Результатом исполнения функции является единственное значение простого, строкового или указательного типа. Поэтому обращение к функции можно использовать выраженное наряду с константой и переменные.

Процедура обычно возвращает несколько значений, подпрограмма состоит из заголовка и тела подпрограммы.

Тело подпрограммы состоит из раздела описаний и раздела операторов.

Структура: PROCEDURE<имя процедуры >;
FUNCTION<имя функции>(параметр: тип);
USES<модули>;
LABEL<метки>;
CONST<константы>;
TYPE<типы>;
VAR<переменные>;
BEGIN
Тело подпрограммы
END;

Вложенные подпрограммы. Принципы локализации имен (локальные и глобальные переменные).

Вложенными являются подпрограммы, вызываемые из других подпрограмм.

Если какая либо программа использует некоторые переменные подпрограммы,

То данные переменные должны быть описаны не в подпрограмме, а в самой программе.

Данные переменные называются глобальными, а переменные, объявленные в модуле или подпрограмме являются локальными, и значения этих временных нельзя использовать в главной программе.

Вызов подпрограмм (формальные и фактические параметры).

Подпрограммы могут быть вызваны как из тела основной программы, так и из тела другой подпрограммы. Для вызова необходимо написать имя процедуры или функции, а также, если надо, ряд переменных передаваемых в подпрограмму.

Пример:

```
PROCEDURE<имя>[(<формальные параметры>)];  
FUNCTION<имя>[(<формальные параметры>):<тип>
```

Сразу за заголовками может следовать одна стандартных директив: ASSEMBLER <тело подпрограммы на ассемблере>, EXTERNAL<объявление внешней подпрограммы>, FAR, FORWARD, INLINE<встроенные машинные инструменты>, INTERRUPT, NEAR.

```
PROCEDURE SB(a: real<формальный параметр>);
```

Фактические параметры – это глобальные переменные подставленные в формальные
ТИП ФАКТ=ТИП ФОРМ.

3. Итог урока.

4. Домашнее задание: Записи в тетради.

Урок 19 Тема: Процедуры в Паскале

Цель: Познакомить учащихся со способами записи процедур в Паскале

1. Письменный опрос

Проверочная работа по теме «Подпрограммы»

Вариант № 1

1. Что такое подпрограмма?
2. На какие группы подразделяются подпрограммы в Паскале?
3. Приведите пример программы с использованием подпрограммы.

2. Новый материал.

Процедура – это независимая именованная часть программы, которую после однократного описания можно многократно вызывать по имени

Procedure Имя(формальные параметры)

Описательная часть процедуры

Begin

Исполнительная часть процедуры

End;

Процедуры без параметров и с параметрами.

PROCEDURE<имя>

Begin

<текст процедуры>

end;

3. Итог урока.

4. Домашнее задание: Записи в тетради.

Урок 20 Тема: Функции в Паскале

Цель: Познакомить учащихся со способами записи функций в Паскале

1. Проверка домашнего задания

2. Новый материал.

Функция передает в программу результат своей работы единственное значение, имя функции может входит в состав выражения

Function Имя(формальные переменные):типрезультата;

Описательная часть функции

Begin

Исполнительная часть функции

Имяфункции := Результат

End;

Табулирование функции одной переменной

Program P; {составление таблицы значений f(x) на отрезке a,b}

var x,y,a,b,h:real;

Function F(x:real):real;

begin

F:=x*x*sin(x);

end;

begin

writeln ('введи a,b,h ');

read (a,b,h);

writeln('таблица');

x:=a;

while x<=b do

begin

y:=F(x);

writeln (x:6:2,y:6:2);

x:=x+h;

```
end;  
end
```

Для возведения числа в натуральную степень можно написать собственную функцию.

```
{Функция возведения числа X в натуральную степень N}  
Function Stepen(X : Real; N : Integer) : Real;  
Var I : Integer; St : Real;  
Begin  
    St := 1;  
    For I := 1 To N Do St := St * X;  
    Stepen := St;  
End;
```

Рекурсия – это такой способ организации вычислительного процесса при котором программа в ходе выполнения составляющих её операторов обращается сама к себе.

Для избежания переполнения стёка, следует размещать промежуточные результаты во вспомогательной переменной.

```
Begin
```

```
.....
```

```
F:=fac(n-1); - вспомогательная переменная
```

```
.....
```

```
End;
```

Рекурсивный вызов может быть косвенным, который разрешается опережающим описанием:

```
Procedure b(j: byte);
```

```
    Forward;
```

```
Procedure a (i: byte);
```

```
Begin
```

```
B(i);
```

```
End;
```

3. Контрольные вопросы

1. Что такое процедура, функция? Виды процедур и функций. Примеры.
2. В чём отличия процедур и функций?
3. На какие группы делятся подпрограммы?
4. Описание и вызов процедуры и функции?
5. Что такое рекурсия? Пример.

4. Итог урока.

5. Домашнее задание: Задачи по вариантам.

Варианты

1. Найти сумму, разность, произведение и частное 2-х натуральных чисел.
2. Найти НОД 4-х чисел.
3. В каком из 3-х чисел цифр больше?
4. Определить сумму цифр 3-х натуральных чисел
5. Определить количество цифр 3-х натуральных чисел

6. Перевод чисел из 10-ой СС в р-ичную ($2 \leq p \leq 9$)

7. У 3-х натуральных чисел найти наибольшую и наименьшую цифры в записи числа.

Урок 21 Тема: Зачетная работа по теме «Подпрограммы»

Цель: Контроль знаний учащихся

Вариант 1

1 – 12 по 1 баллу, 13 – 3 балла, 14 – 15 – 5 баллов

> 20 – оценка 5, от 16 до 20 – оценка 4

от 12 до 15 – оценка 3, < 12 – оценка 2

1. Что такое подпрограмма?
2. Что такое процедура?
3. Перечислите некоторые стандартные процедуры или функции
4. Приведите пример описания процедуры
5. Приведите пример вызова процедуры из задания 4
6. Что такое функция?
7. Приведите пример описания функции
8. Приведите пример вызова функции из задания 4
9. Что такое рекурсия?
10. В чём отличия процедур и функций?
11. Для чего в программе используются подпрограммы?
12. В чем здесь ошибка?

```
Funtion (p; Var x:real)
```

```
End.
```

13. Напишите функцию и процедуру сложения двух чисел.

14. Написать программу для нахождения НОД 4-х чисел.

15. Написать программу для Определения количества цифр 3-х натуральных чисел

Урок 22 Тема: Циклы с предусловием и постусловием

Цель: познакомить учащихся с форматом оператора цикла с предусловием и постусловием, учить составлять программы на использование оператора цикла с предусловием.

1. Анализ зачетной работы

2. Новый материал

Оператор цикла **while** – цикл с предусловием

Оператор цикла **while** имеет следующую форму:

```
while условие do  
    оператор
```

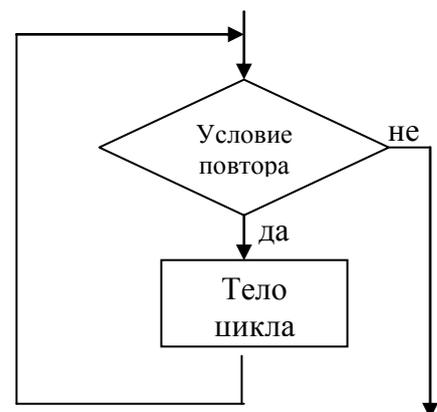
Условие представляет собой выражение логического типа, а оператор после **do** называется *телом цикла*. Перед каждой итерацией цикла условие вычисляется, и если оно истинно, то выполняется тело цикла, в противном случае происходит выход из цикла.

Если *условие* всегда оказывается истинным, то может произойти *зацикливание*:

```
while 2>1 do  
    write(1);
```

Чтобы прервать зациклившуюся программу, следует использовать комбинацию клавиш **Ctrl-F2**

Точка с запятой сразу после **do** в **Pascal ABC** считается синтаксической ошибкой.



Оператор цикла repeat

Оператор цикла **repeat** имеет следующую форму:

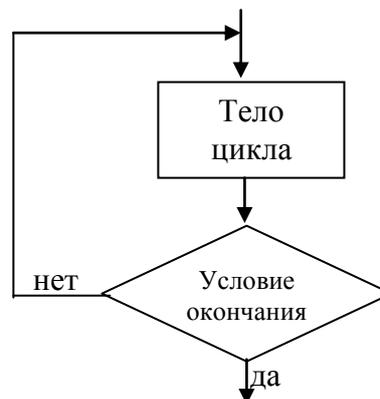
```
repeat  
  операторы  
until условие
```

В отличие от цикла **while**, условие вычисляется после очередной итерации цикла, и если оно истинно, то происходит выход из цикла. Таким образом, операторы, образующие тело цикла оператора **repeat**, выполняются по крайней мере один раз.

Если *условие* всегда оказывается ложным, то может произойти *заикливание*:

```
repeat  
  write(1);  
until 2=1;
```

Чтобы прервать заиклившуюся программу, следует использовать комбинацию клавиш **Ctrl-F2**.



3. Решение задач

Задача 1. Вычислить $1^2+3^2+\dots+25^2$.

```
program z2;  
uses crt;  
var i:integer; s: longint;  
begin  
  s:=0; i:=1;  
  while i<=25 do begin s:=s+sqr(i); i:=i+2; end;  
  writeln (s);  
end.
```

4. Итог урока.

5. Домашнее задание.

Задача 1. Дано число n.

а) напечатать те натуральные числа, квадрат которых не превышает n.

Урок 23 Тема: Цикл с параметром. Использование циклов в задачах

Цель: познакомить учащихся с форматом оператора цикла с параметром; объяснить особенности применения оператора

1. Проверка домашнего задания

2. Новый материал

- Цикл представляет собой последовательность операторов, которая выполняется неоднократно.
- Оператор цикла **for** – цикл со счетчиком (используется с заранее известным **числом повторений**)

Оператор цикла **for** имеет одну из двух форм:

for переменная:=начальное значение **to** конечное значение **do**

оператор;

или

for переменная:=начальное значение **downto** конечное значение **do**

оператор;

Текст от слова **for** до слова **do** включительно называется *заголовком цикла*, а оператор после **do** - *телом цикла*. Переменная после слова **for** называется *параметром цикла*. Для первой формы цикла с ключевым словом **to** параметр цикла меняется от начального значения до

конечного значения, увеличиваясь всякий раз на единицу, а для второй формы ключевым словом **downto** - уменьшаясь на единицу. Для каждого значения переменной-параметра выполняется тело цикла.

3. Решение задач.

Задача 1. Вычислить $1^2+2^2+\dots+12^2$.

```
program z1;
uses crt;
var i:integer; s: longint;
begin
s:=0;
for i:=1 to 100 do s:=s+sqr(i);
writeln (s);
end.
```

$S= 650$

Задача 2. Найти все делители числа n.

```
program z1;
uses crt;
var n, i: integer; k: real;
begin
read (n);
for i:=1 to n do
if n mod I = 0 then writeln (i);
end.
```

Задача 3. Напечатать таблицу квадратных корней из чисел от 1 до 20.

```
program z3;
uses crt;
var i: integer; b: real;
begin
for i:=1 to 20 do
begin
b:= sqrt(i);
writeln (i: 4, b:7:2);
end;
end.
```

4. Итог урока.

5. **Домашнее задание:** Напечатать таблицу перевода дюймов в см. 1 дюйм = 2.54 см

Урок 24 Тема: Решение задач с использованием циклов

Цель: отрабатывать навык решения задач с использованием операторов цикла, закреплять знания учащихся

1. Проверка дом. задания

2. Устная работа

Дана последовательность операторов. Сколько раз будет повторен цикл? Чему будут равны значения a, b, s .

a) $a=1; b=1;$
 $while a+b<8 do$
 $begin a=a+1; b=b+2;end;$
 $s=a+b;$

б) $a=1; b=1;$
 $while a <=3 do$
 $begin$
 $a=a+1;$


```

K:=0;
FOR M:=1 TO 10 DO
IF M MOD 3<>M MOD 5
THEN K:=K+1;
WRITE(K)
END.

```

Чему равно K после выполнения работы программы: 1; 9; 8; 6; 4.

9. Приведите фрагмент программы цикла с постусловием.

10. Нарисуйте блок-схему к задаче: вычисление суммы чисел от – 1 до n

11. Написать программу на языке Паскаль: вычислите сумму ряда если $a_n = \frac{3^n}{2n!}$

12. Написать программу на языке Паскаль: вычисление функции на отрезке [a,b] с шагом h, результат в виде таблицы $y = \text{tg}(x) + \cos\left(\frac{x}{3}\right)$

Урок 26 – 29 Тема: Графика в Паскале. Графические примитивы. Пользовательские процедуры в графике. Модуль GraphABC.

Ввиду большого количества названий процедур и функций графики целесообразно объединить 4 урока вместе.

Цель: Познакомить с возможностями рисования в Паскале, с помощью графических примитивов, процедур и функций.

1. Проверка домашнего задания
2. Новый материал

Графика в Паскале (для Турбо Паскаль)

Константы цветов			Процедура SetLineStyle(Style, P, T)		
Имя	Номер цвета	Цвет	Типы линий - S		
Black	0	Черный	Имя	Значение	Назначение
Blue	1	Темно-синий	SolidLn	0	Сплошная
Green	2	Темно-зеленый	DottedLn	1	Пунктирная
Cyan	3	Бирюзовый	CenterLn	2	Штрихпунктирная
Red	4	Красный	DashedLn	3	Штриховая
Magenta	5	Фиолетовый	Толщины линий - T		
Brown	6	Коричневый	NormWidth	1	Нормальная
LightGray	7	Светло-серый	TrinckWidth	3	Толстая
DarkGray	8	Темно-серый	Процедура SetFillStyle(Style, Color)		
LightBlue	9	Синий	Орнамент заполнения - Style		
LightGreen	10	Светло-зеленый	Имя	Значение	Назначение
LightCyan	11	Светло-бирюзовый	EmptyFill	0	Заполнение цветом фона
LightRed	12	Розовый	SolidFill	1	Однородное заполнение цветом
LightMagenta	13	Малиновый	LineFill	2	Заполнение ---
Yellow	14	Желтый	LtSlashFil	3	Заполнение ///
White	15	Белый	SlashFill	4	Заполнение /// толстыми

Фрагмент программы для подключения модулей Graph (графики), Crt (текста, звука, клавиатуры), инициализации графического режима и т.п.

```
Uses Graph, Crt;
Var Gr, Gm: Integer;
begin
  Gr := Detect;
  Gm:=0;
  InitGraph(Gr, Gm, 'C:\BP\BGI');
```

{все графические процедуры и функции}

```
  ReadKey;
  CloseGraph;
End.
```

BkSlashFill	5	Заполнение \\\ толстыми
LtBkSlashFill	6	Заполнение \\\
HatchFill	7	Заполнение клеткой
XhatchFill	8	Заполнение косой клеткой
InterLeaveFill	9	Заполнение частой клеткой
WideDotFill	10	Заполнение редкими точками
CloseDotFill	11	Заполнение частыми точками
Процедура SetTextStyle(Font, D, C) D – направление текста (горизонтальное или вертикальное), C – коэф увеличения		
Тип шрифта – Font		
Название	Значение	Файл
DefaultFont	0	Нет
TriplexFont	1	Tripp.chr
SmallFont	2	Litt.chr
SansSerifFont	3	Sans.chr
GothicFont	4	Goth.chr

Процедура	Описание	Пример в программе
MoveTo(X, Y);	Назначение координат графического курсора	MoveTo(100, 50);
SetBkColor(Color);	Задание цвета фона экрана	SetBkColor(Red); или SetBkColor(3);
SetColor(Color)	Задание текущего цвета	SetColor(White); или SetColor(1);
ClearDevice;	Очистка графического экрана (очистка рисунка)	ClearDevice;
PutPixel(X, Y, Color)	Выставление точки на экране	PutPixel(150, 100, Red); или PutPixel(150, 100, 4);
Line(X1, Y1, X2, Y2);	Линия от одной точки до другой	Line(10,20,300,400);
LineTo(X, Y);	Линия от текущей точки Используется вместе с MoveTo	MoveTo(100, 50); LineTo(300, 100);
Rectangle(X1, Y1, X2, Y2);	Прямоугольник	Rectangle(50, 100, 200, 300);
Circle(X, Y, Radius);	Окружность	Circle(100, 50, 30);
Arc(X, Y, A1, A2, Radius);	Дуга окружности	Arc(150, 170, 45, 135, 40);
Ellipse(X, Y, A1, A2, Rx, Ry);	Эллипсная дуга	Ellipse (50, 70, 0, 340, 60,20);
SetFillStyle(Style, Color);	Задание стандартных параметров заполнения фигур	SetBkColor(7); SetColor(3); Circle(320,240,100); MoveTo(320,240); SetFillStyle(11,4);
FloodFill(X, Y, Color); Color – цвет, которым рисовали	Заполнение области текущим орнаментом, чаще используется вместе с SetFillStyle	SetBkColor(7); SetColor(3); Circle(320,240,100); MoveTo(320,240);

		SetFillStyle(11,3); FloodFill(320,240, 3)
Bar(X1, Y1, X2, Y2);	Построение закрашенного прямоугольника	SetBkColor(4); SetColor(9); SetFillStyle(10,14); Bar(10,10,50,50);
Bar3D(X1, Y1, X2, Y2, h, Top);	Построение параллелепипеда	SetBkColor(3); SetColor(6); Bar3D(150,200,300,250,120,Торон);
FillEllipse(X, Y, Rx, Ry);	Обведенный линией и закрашенный эллипс	SetBkColor(3); SetColor(9); SetFillStyle(10,14); FillEllipse(120,130,150,170);
Sector(X, Y, A1, A2, Rx, Ry);	Обведенный линией и закрашенный эллипсный сектор	SetBkColor(2); SetColor(6); SetFillStyle(10,11); Sector(120,130,150,170,100,230);
PieSlice(X, Y, A1, A2, R);	Обведенный линией и закрашенный сектор круга	SetBkColor(1); SetColor(8); SetFillStyle(10,11); PieSlice(120,130,400,170,120);
OutTextXY(X, Y, 'Hi'); или OutText('Hi');	Вывод текста на графическом экране	OutTextXY(50,150,'Привет') OutText('Привет')
SetLineStyle(Style, P, T);	Задание стиля линии	SetColor(5); SetLineStyle(2,1,0); Line(100,100,100,400);
SetTextStyle(Font, D, C);	Задание стиля шрифта	SetTextStyle(1, HorizDir,3);
CloseGraph;	Возвращение в текстовой режим	CloseGraph;

Модуль GraphABC (дляPascalABC)

Модуль GraphABC содержит константы, типы, процедуры и функции для рисования в *графическом окне*. Они подразделяются на следующие группы: Графические примитивы

Действия с цветом

Действия с точками и прямоугольниками

Действия с пером

Действия с кистью

Цветовые константы и функции для работы с цветом

Модуль GraphABC содержит константы и функции для работы с цветами. Тип ColorType, описывающий цвет, определен следующим образом:

type ColorType=integer;

Стандартные цвета задаются символическими константами:

clBlack – черный	clAqua – бирюзовый
clPurple – фиолетовый	clOlive – оливковый
clWhite – белый	clFuchsia – сиреневый
clMaroon – темно-красный	clTeal – сине-зеленый
clRed – красный	clGray – темно-серый
clNavy – темно-синий	clLime – ярко-зеленый
clGreen – зеленый	clMoneyGreen – цвет зеленых денег
clBrown – коричневый	clLtGray – светло-серый

clBlue – синий	clDkGray – темно-серый
clSkyBlue – голубой	clMedGray – серый
clYellow – желтый	clSilver – серебряный
clCream – кремовый	

Для работы с цветами используются следующие функции.

function RGB(r,g,b: integer): ColorType;

Возвращает целое значение, являющееся кодом цвета, который содержит красную, зеленую и синюю составляющие с интенсивностями r, g и b соответственно (r, g и b – целые в диапазоне от 0 до 255, причем, 0 соответствует минимальной интенсивности, 255 – максимальной).

function GetRed(color: ColorType): integer;

Выделяет красную составляющую из цвета color (целое в диапазоне от 0 до 255);

function GetGreen(color: ColorType): integer;

Выделяет зеленую составляющую из цвета color (целое в диапазоне от 0 до 255);

function GetBlue(color: ColorType): integer;

Выделяет синюю составляющую из цвета color (целое в диапазоне от 0 до 255).

Действия с пером

function PenX: integer;

function PenY: integer;

Возвращают текущие координаты пера.

procedure SetPenColor(color: integer);

Устанавливает цвет пера, задаваемый параметром color.

function PenColor: integer;

Возвращает текущий цвет пера.

procedure SetPenWidth(w: integer);

Устанавливает ширину пера, равную w пикселям.

function PenWidth: integer;

Возвращает текущую ширину пера.

procedure SetPenStyle(ps: integer);

Устанавливает стиль пера, задаваемый параметром ps.

function PenStyle: integer;

Возвращает текущий стиль пера.

Стили пера задаются следующими именованными константами:

psSolid psDot

psClear psDashDot

psDash psDashDotDot

procedure SetPenMode(m: integer);

Устанавливает режим пера, задаваемый параметром m.

function PenMode: integer;

Возвращает текущий режим пера. Режим пера определяет, как цвет пера взаимодействует с цветом поверхности.

Режимы пера задаются следующими именованными константами:

pmCopy – обычный режим; при рисовании цвет поверхности заменяется цветом пера;

pmNot – режим инвертирования; при рисовании цвет поверхности инвертируется (становится негативным), а цвет пера при этом игнорируется.

Действия с кистью

procedure SetBrushColor(color: integer);

Устанавливает цвет кисти, задаваемый параметром color.

function BrushColor: integer;

Возвращает текущий цвет кисти.

procedure SetBrushPicture(fname: **string**);

Устанавливает в качестве образца для закраски кистью образец, хранящийся в файле fname, при этом текущий цвет кисти при закраске игнорируется.

procedure ClearBrushPicture;

Очищает рисунок-образец, выбранный для кисти.

procedure SetBrushStyle(bs: integer);

Устанавливает стиль кисти, задаваемый параметром bs.

function BrushStyle: integer;

Возвращает текущий стиль кисти.

Действия со шрифтом

procedure SetFontColor(color: integer);

Устанавливает цвет шрифта.

function FontColor: integer;

Возвращает текущий цвет шрифта.

procedure SetFontSize(sz: integer);

Устанавливает размер шрифта в пунктах.

function FontSize: integer;

Возвращает текущий размер шрифта в пунктах.

procedure SetFontName(name: **string**);

Устанавливает наименование шрифта.

function FontName: **string**;

Возвращает текущее наименование шрифта.

По умолчанию установлен шрифт, имеющий наименование MS Sans Serif.

Наиболее распространенные шрифты – это Times, Arial и Courier New.

Наименование шрифта можно набирать без учета регистра.

procedure SetFontStyle(fs: integer);

Устанавливает стиль шрифта.

function FontStyle: integer;

Возвращает текущий стиль шрифта.

Стили шрифта задаются следующими именованными константами:

fsNormal – обычный;

fsBold – жирный;

fsItalic – наклонный;

fsBoldItalic – жирный наклонный;

fsUnderline – подчеркнутый;

fsBoldUnderline – жирный подчеркнутый;

fsItalicUnderline – наклонный подчеркнутый;

fsBoldItalicUnderline – жирный наклонный подчеркнутый.

function TextWidth(s: **string**): integer;

Возвращает ширину строки s в пикселях при текущих настройках шрифта.

function TextHeight(s: **string**): integer;

Возвращает высоту строки s в пикселях при текущих настройках шрифта.

Действия с рисунками

Каждый рисунок имеет описатель (дескриптор), представляющий собой целое число. Это число возвращается функцией загрузки рисунка из файла LoadPicture, а также функцией создания рисунка CreatePicture, после чего передается в качестве первого параметра во все остальные процедуры и функции работы с рисунками.

function LoadPicture(fname: **string**): integer;

`n:=LoadPicture(fname)` – загружает рисунок из файла с именем `fname` в оперативную память и возвращает описатель рисунка в целую переменную `n`; если файл не найден, то возникает ошибка времени выполнения.

procedure `DrawPicture(n,x,y: integer);`

Выводит рисунок с описателем `n` в позицию `(x,y)` графического окна.

procedure `DestroyPicture(n: integer);`

Удаляет рисунок с описателем `n` из оперативной памяти, описатель рисунка при этом становится недействительным.

function `PictureWidth(n: integer): integer;`

Возвращает ширину рисунка с описателем `n`.

function `PictureHeight(n: integer): integer;`

Возвращает высоту рисунка с описателем `n`.

function `CreatePicture: integer;`

Создает рисунок и возвращает его описатель.

procedure `CopyToPicture(n: integer; r: Rect);`

Сохраняет изображение из прямоугольника `r` графического окна в рисунок с описателем `n`.

procedure `SavePicture(n: integer; fname: string);`

Сохраняет рисунок с описателем `n` в файл с именем `fname`.

procedure `SetPictureSize(n,w,h: integer);`

Устанавливает размер рисунка с описателем `n` равным `w` на `h` пикселей.

procedure `SetPictureTransparent(n: integer; b: boolean);`

Устанавливает (`b=True`) или отключает (`b=False`) режим прозрачности при рисовании рисунка с описателем `n`. Если `b=True` и рисунок имеет белый фон, то при его рисовании фон не отображается.

function `ImageIntersect(n1,n2: integer): boolean;`

Определяет, пересекаются ли изображения на рисунках с описателями `n1` и `n2`. Рисунки должны иметь одинаковый размер. Белый цвет на рисунке считается прозрачным.

Изображения на рисунках считаются пересекающимися, если имеется хотя бы один пиксел, который не прозрачен и имеет одни и те же координаты для обоих рисунков.

Действия с графическим окном

procedure `ClearWindow;`

Очищает графическое окно кистью белого цвета;

function `WindowWidth: integer;`

Возвращает ширину графического окна;

function `WindowHeight: integer;`

Возвращает высоту графического окна;

function `WindowLeft: integer;`

Возвращает отступ графического окна от левого края экрана;

function `WindowTop: integer;`

Возвращает отступ графического окна от верхнего края экрана;

function `WindowCaption: string;`

Возвращает заголовок графического окна;

procedure `SetWindowWidth(w: integer);`

Устанавливает ширину графического окна;

procedure `SetWindowHeight(h: integer);`

Устанавливает высоту графического окна;

procedure `SetWindowLeft(l: integer);`

Устанавливает отступ графического окна от левого края экрана;

procedure `SetWindowTop(t: integer);`

Устанавливает отступ графического окна от верхнего края экрана;

procedure `SetWindowSize(w,h: integer);`

Устанавливает ширину и высоту графического окна;

procedure SetWindowPos(l,t: integer);

Устанавливает отступ графического окна от левого и верхнего края экрана;

procedure SetWindowCaption(s: **string**);

Устанавливает заголовок графического окна;

procedure SaveWindow(fname: **string**);

Сохраняет содержимое графического окна в файл с именем fname;

procedure LoadWindow(fname: **string**);

Выводит в графическое окно рисунок из файла с именем fname;

procedure CloseWindow;

Закрывает графическое окно.

function ScreenWidth: integer;

Возвращает ширину экрана;

function ScreenHeight: integer;

Возвращает высоту экрана;

procedure CenterWindow;

Центрирует графическое окно по центру экрана.

procedure MaximizeWindow;

Максимизирует графическое окно на экране, оставляя видимой лишь полосу задач.

Все размеры устанавливаются и возвращаются в пикселах.

Функция LoadWindow может загружать лишь файлы формата bmp.

Функция SaveWindow сохраняет рисунок в файле формата bmp с 256 цветами.

Задание режимов вывода

procedure SetDrawingSurface(n: integer);

Устанавливает в качестве канвы для рисования рисунок с описателем n. В результате весь графический вывод осуществляется не на экран, а на рисунок; настройки кисти, пера и шрифта также осуществляются для рисунка.

procedure RestoreDrawingSurface;

Устанавливает в качестве канвы для рисования графическое окно.

procedure Redraw;

Осуществляет перерисовку окна вывода при заблокированном выводе в графическое окно.

procedure LockDrawing;

Блокирует вывод в графическое окно, осуществляя рисование только во внеэкранный буфере. Для перерисовки графического окна требуется вызвать процедуру Redraw. Если графический вывод перенаправлен в рисунок вызовом процедуры SetDrawingSurface, то не оказывает никакого воздействия на вывод.

procedure UnlockDrawing;

Снимает блокировку вывода в графическое окно.

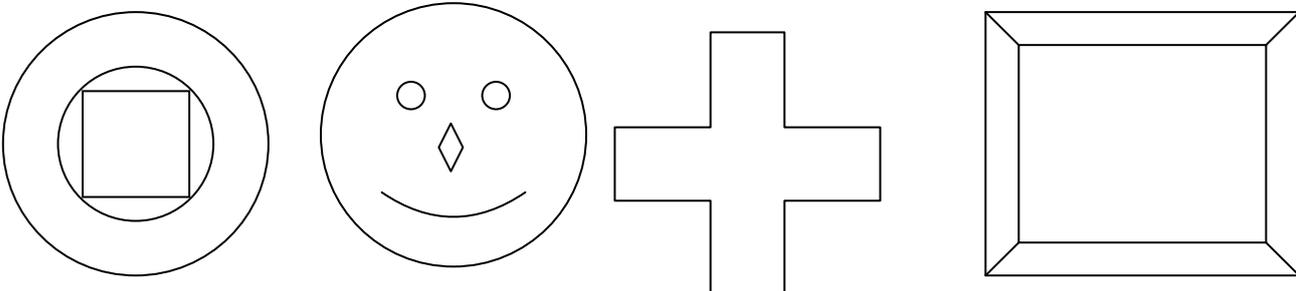
function DrawingIsLocked: boolean;

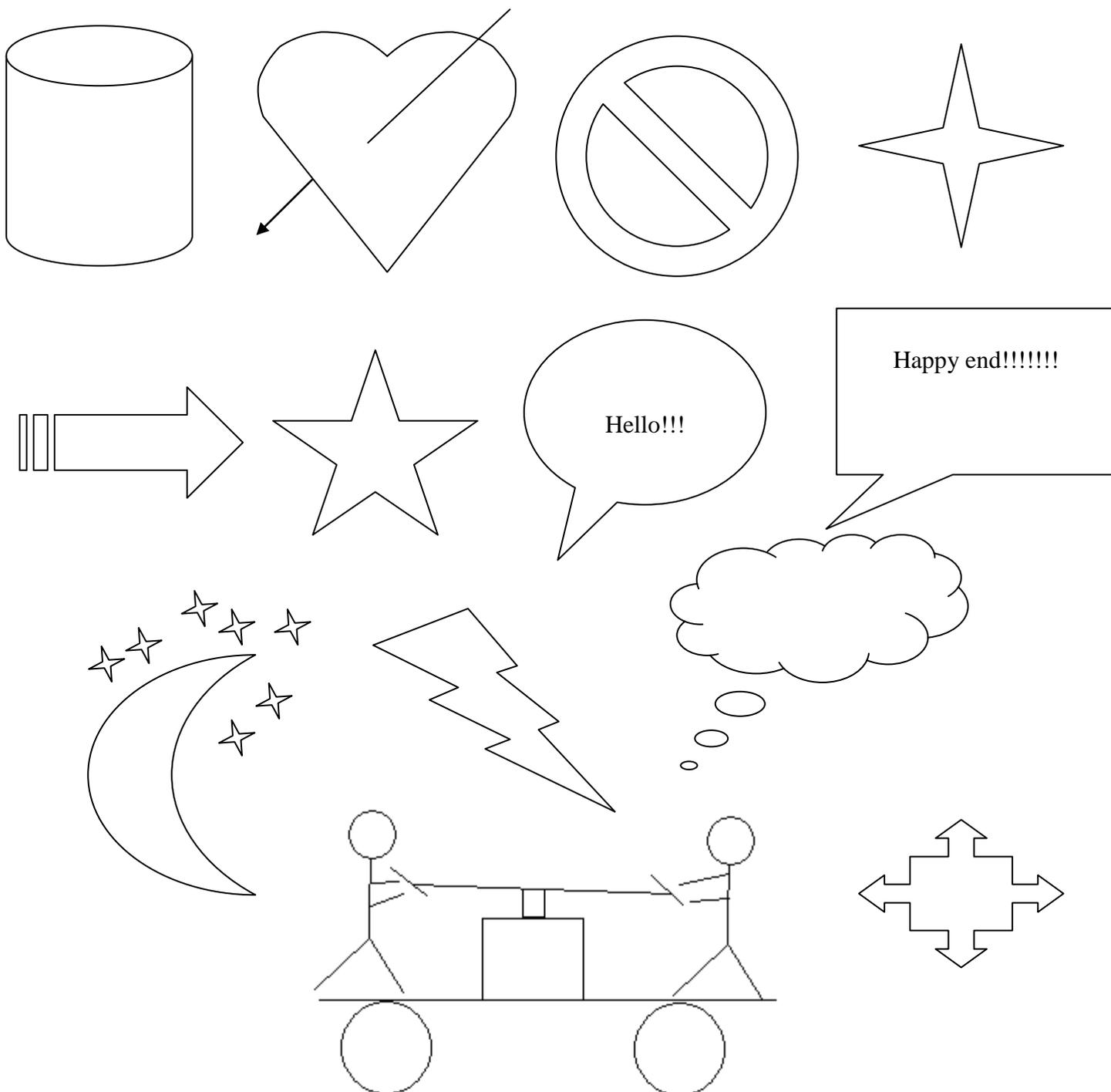
Возвращает True, если вывод в графическое окно заблокирован, и False в противном случае.

Блокировка вывода в графическое окно с последующим вызовом Redraw используется для простейшего создания анимации без мерцания.

3. Решение задач.

Нарисовать и разукрасить одну из представленных фигур (в правом нижнем углу подписать, кто выполнил данную работу)





4. Итоги уроков.

5. Домашнее задание: Собственный рисунок.

Урок 30 Тема: Зачетная работа по теме «Графика»

Цель: Контроль знаний учащихся

Контроль знаний учащихся осуществляется проверкой подробной записи стандартных процедур и функций графики

Урок 31 Тема: Повторение курса 10-го класса

Цель: Систематизация знаний учащихся за 10 класс

1. Анализ зачетной работы

2. Решение задач.

По выбору учащихся

Задачи по теме линейные алгоритмы

1. Вывести на экран надпись: «Привет, меня зовут Вася!»
2. Вывести на экран надпись: «Здравствуйтесь и до свидания!»
3. Вывести на экран надпись:
«Здравствуйтесь - Юзер!»
«До свидания - Юзер!»
4. Вывести на экран надпись:
-----Начало-----
| Здравствуйтесь - Юзер! |
| До свидания - Юзер! |
-----конец-----
5. Вывести на экран надпись:

* ++++++ *
* ++++++ *

6. Вычислить сумму двух целых чисел x и y .
7. Вычислить разность двух целых чисел x и y .
8. Вычислить разность двух чисел x и y .
9. Вычислить частное двух чисел a и b , где b не равно 0.
10. Вычислить $b + \sqrt{b^2 + 4ac} - a^3c$
11. Вычислить $\frac{a}{c} \cdot \frac{b}{d} - \frac{ab - c}{cd}$
12. Вычислить $|x^2 - x^3| - \frac{7x}{x^3 - 15x}$

Задачи по теме условный оператор

1. Определить, есть ли среди 3 введённых чисел хотя бы одно отрицательное число.
2. Определить, есть ли среди 3 введённых чисел отрицательные числа.
3. Определить, есть ли среди 4 введённых чисел число 0.
4. Определить, есть ли среди 5 введённых чисел число 1.
5. Подсчитать количество положительных чисел среди чисел m, n, p .
6. Определить, делителем каких чисел x, y, z является число a .
7. Написать программу нахождения большего и меньшего из 3 чисел.
8. Треугольник задан длинами сторон a, b, c . Определить является ли он прямоугольным. Если нет вычислить синусы его углов.
9. Написать программу, в которой находятся корни квадратного уравнения $ax^2 + bx + c = 0$.
10. Даны 3 положительных числа a, b, c проверить будут ли они сторонами треугольника.
11. Вычислить значение функции
$$y = \begin{cases} x^2 - 3x + 9, & \text{если } x \leq 3 \\ \frac{1}{x^3 + 6}, & \text{в противном случае} \end{cases}$$
12. Вычислить значение функции
$$y = \begin{cases} -3x + 9 & \text{если } x \leq 7 \\ \end{cases}$$

$\frac{1}{x-7}$, в противном случае

13. Вычислить значение функции

$$y = \begin{cases} 9, & \text{если } x \leq -3 \\ \frac{1}{x^2 + 1}, & \text{в противном случае} \end{cases}$$

Задачи на оператор выбора

1. Составить программу которая по номеру дня недели (целому числу от 1 до 7) выдаст название дня недели.
2. Для каждой введенной цифры (0 – 9) вывести соответствующее название на английском языке (0 – zero, 1 – one, 2 – two, ...)
3. Составить программу, позволяющую получить словесное описание школьных отметок (1 – плохо, 2 – неудовлетворительно, 3 – удовлетворительно, 4 – хорошо, 5 – отлично)
4. * Пусть элементами прямоугольного равнобедренного треугольника является:
 - 1) катет a;
 - 2) гипотенуза b;
 - 3) высота, опущенная из вершины прямого угла на гипотенузу h;
 - 4) площадь S.

Составить программу, которая по заданному номеру и значению соответствующего элемента вычисляла бы значение всех остальных элементов треугольника.

5. Написать программу, которая бы по введённому номеру времени года (1 – зима, 2 – весна, 3 – лето, 4 – осень) выдавала соответствующее этому времени года месяца, количество дней в каждом месяце.

Задачи на циклы

1. Найти сумму всех чисел натуральных чисел до n, которые делятся на k.
2. Вывести на экран 10 надписей «Привет всем!»
3. Вычислить произведение всех целых чисел от 2 до 10.
4. Вычислить произведение всех целых чисел от 3 до n.
5. Вычислить произведение всех целых чисел от – n до n.
6. Вычислить сумму всех натуральных чисел до 15.
7. Вычислить сумму всех натуральных чисел от 19 до 43.
8. Вычислить сумму всех натуральных чисел от k до p.
9. Вывести на экран все натуральные числа от 1 до n, в строку.
10. Вывести на экран все натуральные числа от n до 5.
11. Вывести на экран все целые числа от n до – 10, в столбик.
12. Вывести на экран все натуральные числа от 2 до 50, кратные 2 (чётные).
13. Вывести на экран все нечётные натуральные числа от 1 до 10.
14. Вывести на экран все двузначные чётные натуральные числа.
15. * Вычислить сумму ряда $S = 1! + 2! + 3! + \dots + n!$
16. * Вычислить сумму ряда $S = \frac{2}{1} + \frac{3}{2} + \frac{4}{3} + \dots + \frac{n+1}{n}$
17. Вычислить значение функции на отрезке [a, b] $a < b$ с шагом h ($h \neq 1$). Результат представить в виде 2 столбиков: 1 столбец – значение аргумента, 2 столбец – значение функции.

a) $y = x - \sin x$;

б) $y = x \odot \cos\left(\frac{1}{x}\right) + 2$;

$$в) y = \sqrt{x} \circ \cos x.$$

3. Итог урока.

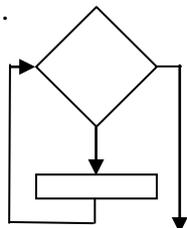
4. Домашнее задание: Записи в тетради, задачи по вариантам.

Урок 32 Тема: Зачет за курс 10-го класса

Цель: Контроль знаний учащихся за 10 класс

Вариант № 1

1.



Блок–схема изображает:

- 1) команду ветвления
- 2) команду цикла
- 3) простую команду
- 4) команду графики

2. Результатом выполнения программы на языке Паскаль при $a = 15$ будет:

Program rem;

Var a: integer;

Begin

Write('введите a');

a:= a * 10;

Write(a);

End.

3. Выберите команды цикла с параметром

- 1) if then else
- 2) for to do
- 3) repeat until
- 4) x:=x + 1

4. Для закрытия графического режима в Паскале существует команда

- 1) Close Graf
- 2) Close Graph
- 3) Close
- 4) CloseGraph

5. Команда для рисования окружности в Паскале выглядит следующим образом

- 1) Circle(R,x,y)
- 2) Circle(x1,y1,x2,y2,R)
- 3) Circle(x2,y2,R)
- 4) Circle(x,y,A1,A2,R)

6. Для создания цвета рисования используется команда:

- 1) Set Bk Color
- 2) SetColor
- 3) Set Color
- 4) SetBkColor

7. Команда Delay относится к модулю

- 1) Grah
- 2) Dos
- 3) Grt
- 4) К перечисленным не относится

8. Команда ReadKey обозначает

- 1) нажатие клавиши пробел
- 2) нажатие клавиши энтер
- 3) нажатие любой клавиши
- 4) нажатие клавиши Esc

9. после выполнения команд

- 1) три квадрата
- 2) три окружности
- 3) три прямоугольника
- 4) три эллипса

Rectangle(100,100,200,200);

Rectangle(120,120,180,180);

Rectangle(140,140,160,160);

на экране нарисуеться:

10. Результатом выполнения программы будет

Program rem;

Var M, x,y,z: integer;

Begin

X:=3;

Y:= - 2;

Z:= 1;

if x>=z then M:=x else M:=z;

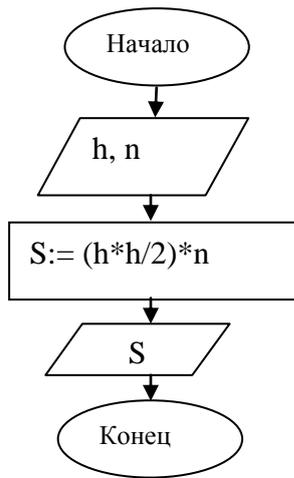
if y>=z then M:=y else M:=z;

Write(M);

End.

- 1) 3
- 2) - 2
- 3) 1
- 4) будет ошибка

11. Результатом выполнения блок схемы при h = 8, n = 5 будет



- 1) 160
- 2) 140
- 3) 6,4
- 4) ответа не будет

12. Определите правильно записанную формулу на языке Паскаль $\frac{x+y}{x+1} - \frac{xy-12}{34+x}$

- 1) $(x+y)/(x+1) - (x*y - 12)/(34+x)$
- 2) $x + y/x + 1 - x*y - 12/34 + x$
- 3) $(x+y)/x/1 - (x*y - 12)/34/x$
- 4) $(x+y)*(x+1) - (xy - 12)/(34+x)$

13. Команда A mod B обозначает:

- 1) остаток от целого деления B на A
- 2) остаток от целого деления A на B
- 3) целое деление B на A
- 4) целое деление A на B

14. Определите значение целочисленной переменной S после выполнения алгоритма

S:=0; j:=3;

While

Begin

S:=

j:=j + 2;

i:=i + 1;

End;

15. Определить конечное значение переменной X в результате выполнения алгоритма:

X:= 2;

X:=X*X;

X:=X*X*X;

- 1) 8
- 2) 16
- 3) 64
- 4) правильного ответа нет

16. В системе Турбо Паскаль выберите команду для запуска программы:

- 1) Run / Run
- 2) File / Save
- 3) Edit / Paste
- 4) Window / Color

17. Программирование – это раздел информатики, занимающийся

- 1) вопросами разработки программного обеспечения и компьютерных деталей

- 2) вопросами разработки программ управления компьютером
- 3) вопросами разработки программ компьютерных игр
- 4) вопросами разработки программ операционной системы Windows XP
- 18. Результат решения задачи на языке Паскаль выносится на экран с помощью команды
 - 1) Readln 3) ReadKey
 - 2) Writeln 4) Vivod
- 19. Описание всех модулей в программе осуществляется в разделе
 - 1) Uses 3) Type
 - 2) Program 4) InitGraph
- 20. Выражение $\sqrt{a \cdot b}$ на языке Паскаль записывается так:
 - 1) $\sqrt{a * b}$ 3) SQRT(a*b)
 - 2) $(a*b)^2$ 4) SQR(a*b)
- 21. Команды While do являются командами
 - 1) ветвления 3) цикла
 - 2) процедур 4) графики

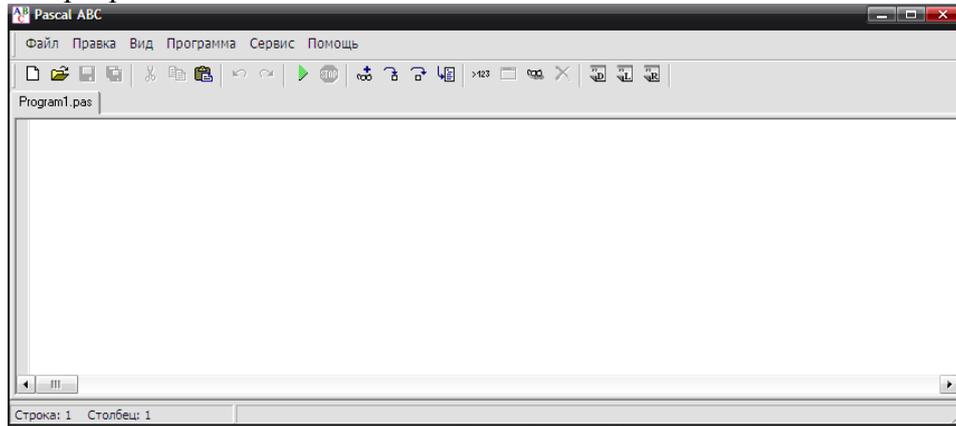
Практические занятия

Урок 1 Тема: Знакомство с системой Паскаль, операторы ввода, вывода, присваивания

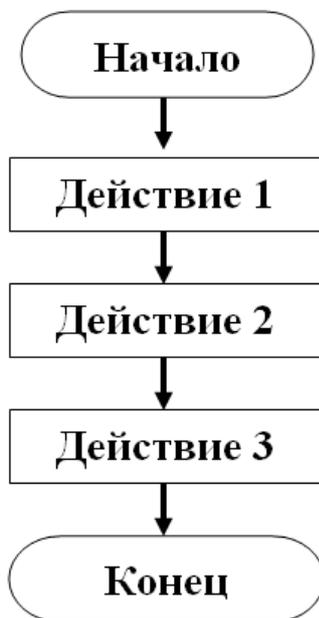
Цель: Познакомить с интерфейсом программы PascalABC, познакомить с линейными операторами Паскаля

1. Новый материал

Интерфейс программы PascalABC



Линейный алгоритм – это алгоритм, действия в котором выполняются однократно в заданном порядке



2. Решение задач.

Написать свое имя

Program Name;

Uses crt;

Begin

Write(' Привет ');

End.

Дописать программу так, чтобы появилось два имени в разных строчках

Найти сумму 2 чисел

Program Summa;

Uses crt;

Var a, b, s: integer;

Begin

Read(a, b);

S:=a+b;

Writeln(s);

End.

Дописать программу так, чтобы появилась надпись **Результат сложения =**

3. Итог урока.

4. Домашнее задание: По вариантам

Составить линейный алгоритм на языке Паскаль, для вычисления по формуле:

1) $y = x + 4a - 7$

2) $x = 4y + 2c - 9$

3) $a = 4 - 2y - t$

4) $s = v t$

5) $d = 2 R$

6) $h = c + x - 5$

7) $t = 6x - 5y$

8) $f = 1 - 7x$

9) $k = 10f + 11 - 3d$

10) $L = 2A - 4B + 3$

В программе предусмотреть ввод переменных и результат.

Урок 2 Тема: Операторы ввода, вывода, присваивания, приемы ввода и вывода

Цель: Научиться применять линейные операторы для решения задач

1. Проверка домашнего задания

2. Новый материал

Write – вывод информации на экран без перевода курсора на следующую строку.

Writeln – вывод информации на экран с переводом курсора на следующую строку

Примеры:

Write('привет')

Writeln('Привет')

операторы ввода информации с клавиатуры,

Необходимо помнить: они требуют обязательного ввода некоторой информации, а затем нажатия клавиши Enter.

Примеры: Readln(S1); Read(S1);

Оператор присваивания

Sum:= S1+ S2 + b;

X:=x;

Y:=a+3;

B:=2

3. Решение задач.

1. Вычислить площадь прямоугольного треугольника по его катетам

2. Заданы координаты трёх вершин треугольника (x_1, y_1) , (x_2, y_2) , (x_3, y_3) . Найти его периметр и площадь. $S = p(p - a)(p - b)(p - c)$, где p – полупериметр, длина стороны

$$a = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

3. Даны два числа. Найти среднее арифметическое кубов этих чисел.

4. Дано расстояние в метрах. Найти число полных километров в нем.

5. Составить программу, которая будет запрашивать у пользователя год его рождения и текущий год и выводит возраст пользователя. Пример работы программы:

В каком году ты родился?

1990

Какой сейчас год?

2010

Тебе в этом году 20 лет

6. Составить программу, которая запрашивает у пользователя два целых числа и выводит квадрат суммы $(a + b)^2$ и сумму квадратов $a^2 + b^2$ этих чисел. Пример работы программы:

Введите два числа

a=3

b=2

Квадрат суммы 3 и 2 равен 25

Сумма квадратов 3 и 2 равна 13

4. Итог урока.

5. Домашнее задание: Задачи по вариантам.

Урок 3 Тема: Вещественный и целый типы при решении задач

Цель: научиться использовать вещественный и целый типы при решении числовых задач

1. Проверка домашнего задания

2. Устный опрос

Определить типы переменных А и В

Program Summa;

Var A,B: ;

Begin

Write('Введите первое слагаемое: A= ');

Readln(A);

Write('Введите второе слагаемое: B= ');

Readln(B);

Writeln;

Writeln('A+B= ',A+B);

Writeln('Всем пока!')

End.

Поясните каждую строчку программы:

Program Naiti_ot;

Var x,y:integer;

ot:real;

begin

writeln('Введите x и y:');

readln(x,y);

ot:=(x*x+y*y)/x-(x-y*y)/y;

writeln('Ответ= ',ot:10:6);

readln;

End.

3. Решение задач.

1. Напишите программу, по которой вычисляется формула: $\frac{b-a}{2-b^2} \cdot \frac{b}{a}$, определить типы переменных.

2. Вычислить $b + \sqrt{b^2 + 4ac} - a^3c$, определить типы переменных.

3. Вычислить $|x^2 - x^3| - \frac{7x}{x^3 - 15x}$, определить типы переменных.

4. Итог урока.

5. Домашнее задание: Типы данных.

Урок 4 – 8 Тема: Решение задач с использованием простых операторов

Цель: научиться решать задачи с линейными операторами

1. Проверка домашнего задания

2. Решение задач.

1. Вычислить периметр и площадь прямоугольного треугольника по заданным длинам двух катетов a и b .
2. Заданы координаты трёх вершин треугольника (x_1, y_1) , (x_2, y_2) , (x_3, y_3) . Найти его периметр и площадь.
3. Вычислить длину окружности и площадь круга одного и того же заданного радиуса R .
4. Найти произведение цифр заданного четырёхзначного числа.
5. Даны два числа. Найти среднее арифметическое кубов этих чисел и среднее геометрическое модулей этих чисел.
6. Вычислить расстояние между двумя точками с данными координатами (x_1, y_1) и (x_2, y_2) .
7. Даны два действительных числа x и y . Вычислить их сумму, разность, произведение и частное.
8. Дана длина ребра куба. Найти площадь грани, площадь полной поверхности и объём этого куба.
9. Дана сторона равностороннего треугольника. Найти площадь этого треугольника, его высоты, радиусы вписанной и описанной окружностей.
10. Известна длина окружности. Найти площадь круга, ограниченного этой окружностью.
11. Найти площадь кольца, внутренний радиус которого равен r , а внешний – заданному числу R ($R > r$).
12. Треугольник задан величинами своих углов и радиусом описанной окружности. Найти стороны треугольника.
13. Найти площадь равнобедренной трапеции с основанием a и b и углом α при большем основании a .
14. Вычислить корни квадратного уравнения $ax^2 + bx + c = 0$, заданного коэффициентами a , b и c (предполагается, что a не равно 0 и что дискриминант уравнения неотрицателен).
15. Дано действительное число x . Не пользуясь никакими другими арифметическими операциями, кроме умножения, сложения и вычитания, вычислить за минимальное число операций $2x^4 - 3x^3 + 4x^2 - 5x + 6$.
16. Дано x . Получить значения $-2x + 3x^2 - 4x^3$ и $1 + 2x + 3x^2 + 4x^3$. Позаботиться об экономии операций.
17. Найти площадь треугольника, две стороны которого равны a и b , а угол между этими сторонами равен g .
18. Дано a . Не используя никаких функций и никаких операций, кроме умножения, получить a^8 за три операции; a^{10} и a^{16} за четыре операции.
19. Найти сумму членов арифметической прогрессии, если известны ее первый член, знаменатель и число членов прогрессии.
20. Найти все углы треугольника со сторонами a, b, c . Предусмотреть в программе перевод радианной меры угла в градусы, минуты и секунды.
21. Три сопротивления R_1, R_2, R_3 соединены параллельно. Найдите сопротивление соединения.
22. Составить программу для вычисления пути, пройденного лодкой, если ее скорость в стоячей воде v км/ч, скорость течения реки и км/ч, время движения по озеру t_1 ч, a против течения реки — t_2 ч.
23. Текущее показание электронных часов: m часов ($0 < m < 23$), n мин ($0 < n < 59$), k сек ($0 < k < 59$). Какое время будут показывать часы через p ч q мин r с?

24. Полторы кошки за полтора часа съедают полторы мышки. Сколько мышек съедят X кошек за Y часов?
25. Составить программу вычисления объема цилиндра и конуса, которые имеют одинаковую высоту H и одинаковый радиус основания R .
26. Ввести любой символ и определить его порядковый номер, а также указать предыдущий и последующий символы.
27. Дана величина A , выражающая объем информации в байтах. Перевести A в более крупные единицы измерения информации.

3. Итог урока.

4. Домашнее задание: Задачи по вариантам, записи в тетради.

Урок 9 Тема: Самостоятельная работа по теме «Линейные программы»

Цель: Контроль знаний учащихся

Вариант 1

1. Указать значения величин S и k после выполнения следующих операторов присваивания:
 - $s:=14$
 - $k:=-3$
 - $d:=s+1$
 - $s:=d$
 - $k:=2*S$
2. В каком из примеров допущены ошибки и какие именно?
 Var A, B: real;
 - a) $A:=5-B$
 - b) $A+B:=5$
 - c) $\text{Sqrt}(A):=B$
 - d) $A:=B$
 - e) $5:=A$
 - f) $A:=5$
3. Записать выражение по правилам языка

Паскаль
$$\frac{\sin \sqrt{x+y^2}}{ab-|y|}$$

4. Вычислите площадь треугольника по известным длинам его сторон. Воспользуйтесь формулой Герона:

$$S = \sqrt{p(p-a)(p-b)(p-c)},$$
 где

$$p = \frac{a+b+c}{2}.$$
5. Найти высоту равностороннего треугольника, если сторона равна a .

Вариант 2

1. Указать значения величин S и k после выполнения следующих операторов присваивания:
 - $s:=0$
 - $k:=30$
 - $d:=k-5$
 - $k:=2*d$
 - $s:=k-100$
2. В каком из примеров допущены ошибки и какие именно?
 Var A, B: integer;
 - a) $A:=B+7;$
 - b) $A-B:=4$
 - c) $\text{Abs}(A):=5-B$
 - d) $B:=A$
 - e) $A:=-7$
 - f) $A:=\text{Sqrt}(B)$
3. Записать выражение по правилам языка Паскаль

языка Паскаль
$$\frac{\cos|x|-x^2}{\sqrt{x+y}}$$

4. Определите расстояние, пройденное телом за время t , если известны его начальная скорость и постоянное ускорение. Формула:

$$S = Vt + \frac{at^2}{2}.$$
5. Найти боковую сторону равнобедренного треугольника, если основание a , высота к основанию h .

Урок 10 – 11 Тема: Решение задач по формулам

Цель: научиться составлять формулы на языке Паскаль, научиться решать задачи с формулами

1. Анализ самостоятельной работы
2. Решение задач.

$$\frac{3 + e^{y-1}}{1 + x^2 |y - \operatorname{tg} x|}$$

$$x - \frac{x^3}{3} + \frac{x^5}{5}$$

$$(1 - \operatorname{tg} x)^{\operatorname{ctg} x} + \cos(x - y)$$

$$\frac{\ln |\cos x|}{\ln(1 + x^2)}$$

$$e^{x-x-2+(1+x)^x} \\ 2\operatorname{ctg}(3x) - \frac{\operatorname{Ln} \cos x}{\operatorname{Ln}(1 + x^2)}$$

$$\ln \left| \left(y - \sqrt{|x|} \right) \left(x - \frac{y}{x + \frac{x^2}{4}} \right) \right|$$

$$\left(\frac{x+1}{x-1} \right)^x + 18xy^2$$

$$\left(1 + \frac{1}{x^2} \right)^x - 12x^2y$$

$$2^{-x} - \cos x + \sin(2xy)$$

$$2\operatorname{ctg}(3x) - \frac{1}{12x^2 + 7x - 5}$$

$$|x^2 - x^3| - \frac{7x}{x^3 - 15x}$$

$$x \cdot \ln x + \frac{y}{\cos x - \frac{x}{3}}$$

$$\sin \sqrt{x+1} - \sin \sqrt{x-1}$$

$$e^x - \frac{Y^2 + 12XY - 3X^2}{18y - 1}$$

$$\frac{1 + \sin^2(x+y)}{2 + \left| x - \frac{2x}{1 + x^2 y^2} \right|} + x$$

$$x - 10^{\sin x} + \cos(x-y)$$

$$x - 10 \sin x + |x^4 - x^5|$$

$$3^x - 4x + (y - \sqrt{|x|})$$

3. Итог урока.
4. Домашнее задание: формулы по вариантам.

Урок 12 Тема: Решение задач на логические выражения

Цель: Научится составлять логические выражения

1. Новый материал

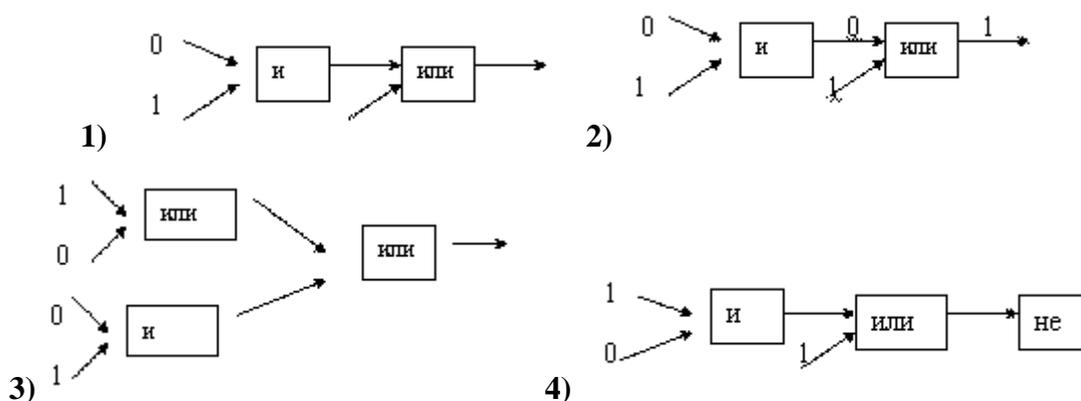
Взять из теории

Таблица истинности

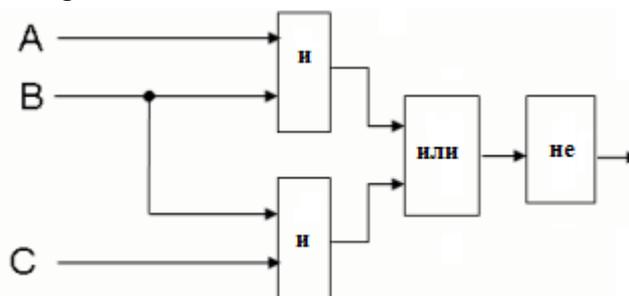
Название— Аргументы x y		И (AND)	НЕ (NOT)	ИЛИ (OR)
0	0	0	1	0
0	1	0	1	1
1	0	0	0	1
1	1	1	0	1

2. Решение задач.

1. Составить логические выражения по логическим схемам:



2. Составить логические выражения по логическим схемам:



3. Постройте логические схемы, соответствующие логическим выражениям и найдите значения логических выражений:

$$F = A \text{ или } B \text{ и не } C, \text{ если } A=1, B=1, C=1.$$

$$F = \text{не}(A \text{ или } B \text{ и } C), \text{ если } A=0, B=1, C=1.$$

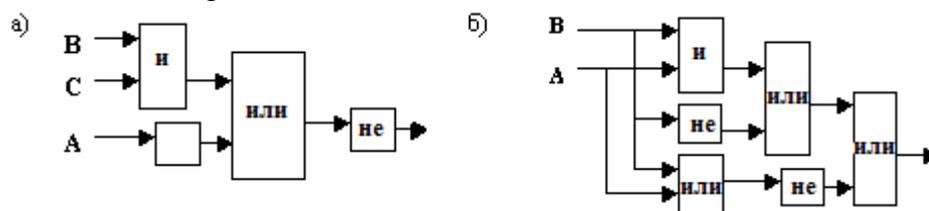
4. Построить логические схемы по формулам:

$$F = A \text{ или } B \text{ и } C, \text{ если } A=1, B=0, C=1;$$

$$F = (A \text{ или } B) \text{ и } (C \text{ или } B), \text{ если } A=0, B=1, C=0;$$

$$F = (A \text{ и } B \text{ и } C), \text{ если } A=0, B=0, C=1.$$

5. Составить логические выражения по схемам:



3. Итог урока.

4. Домашнее задание: Записи в тетради, задачи по вариантам.

Варианты

Составить логические выражения

- 1) НЕ X И НЕ Y И Z 2) НЕ X ИЛИ НЕ Y ИЛИ Z
- 3) X ИЛИ Y ИЛИ НЕ Z 4) X ИЛИ Y ИЛИ Z
- 5) A ИЛИ B 6) A И B 7) НЕ A ИЛИ НЕ B 8) НЕ A И B
- 9) $X > 1$ И $((X < 5) \rightarrow (X < 3))$
- 10) (K И L И M) ИЛИ (НЕ L И НЕ M И N)

Урок 13 Тема: Самостоятельная работа по теме «Формулы»

Цель: Контроль знаний учащихся

Вариант 1

Составить программы на Паскале для вычисления по формулам:

$$1. \frac{a}{c} \cdot \frac{b}{d} - \frac{ab - c}{cd}$$

$$2. \frac{\sin x + \cos y}{\cos x - \sin y} \cdot \operatorname{tg} xy$$

$$3. 2 \operatorname{ctg}(3x) - \frac{1}{12x^2 + 7x - 5}$$

$$4. x - 10 \sin x + |x^4 - x^5|$$

$$5. 2 + \left| x - \frac{2x}{1 + x^2 y^2} \right| + x$$

Вариант 2

Составить программы на Паскале для вычисления по формулам:

$$1. \frac{x^2 - 7x + 10}{x^2 - 8x + 12}$$

$$2. \frac{\cos x}{\pi - 2x} + 16x \cdot \cos(xy) - 2$$

$$3. \frac{3 + e^{y-1}}{1 + x^2 |y - \operatorname{tg} x|}$$

$$4. \sin \sqrt{x+1} - \sin \sqrt{x-1}$$

$$5. (1 - \operatorname{tg} x)^{\operatorname{ctg} x} + \cos(x - y)$$

Урок 14 Тема: Решение задач на повторение

Цель: закрепить навыки решения задач с линейными операторами.

1. Анализ самостоятельной работы
2. Решение задач.

Использовать задачи на линейные операторы, не решенные до этого

3. Итог урока.
4. Домашнее задание: нет.

Урок 15 – 16 Тема: Решение задач с использованием условного оператора

Цель: научиться решать задачи с использованием условного оператора

1. Новый материал

If <условие> then оператор 1 else оператор 2;

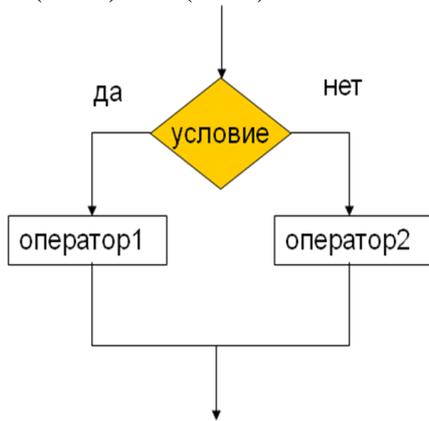
Выполнение условного оператора начинается с вычисления условия.

Условие - это логическое выражение, в зависимости от которого выбирается одна из двух альтернативных ветвей алгоритма.
 Если оно *истинно* (true), то выполняется *оператор стоящий после* служебного слова then.
 Если условие *ложно* (false), то выполняется *оператор, стоящий после* служебного слова else.

Примеры

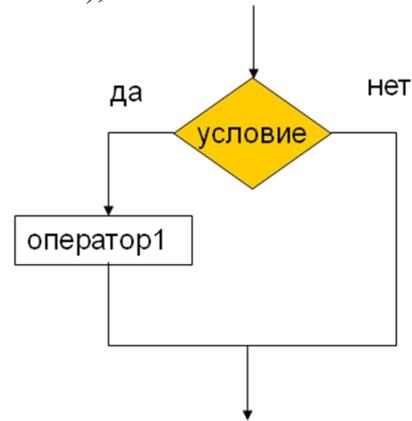
```
If A>B then Writeln ('А больше В')
           else Writeln ('А меньше или равно В');
```

```
If (a>=0) and (a<10) then Writeln ('однозначное число');
```



Полная форма

```
If <условие> then оператор 1 else оператор 2;
```



Сокращенная форма

```
If <условие> then оператор1;
```

2. Решение задач.

Проверить выполнение программы, определить результат работы, найти ошибки (из файла)

```
Program Divon;
Var A:integer;
Begin
  Writeln('Введите целое число ');
  Readln(A);
  If (A mod 5<>0) then
    begin
      Writeln('No');
      Writeln('-----')
    end
  else
    begin
      Writeln('Yes');
      Writeln('-----')
    end
End.
```

```
Program Divizion;
Var A:integer;
Begin
  Writeln('Введите целое число ');
  Readln(A);
  If (A mod 3=0) then
    begin
      Writeln('да');
      Writeln('-----')
    end
  else
    begin
      Writeln('нет');
      Writeln('-----')
    end
End.
```

1. Если число, введенное с клавиатуры, больше 100, то вывести слово «Удача» в противном случае «Неудача»
2. Если число, введенное с клавиатуры, равно 5, то вывести слова «Получай 5»
3. Определить, делителем каких чисел x, y, z является число a.
4. Определить, есть ли среди 5 введенных чисел число 1
5. Из файла. Исправить программу, так что бы на экране появлялся ответ: **Первое больше** или **Второе больше**

```
Program Big1;
Var A, B, max : real;
```

Begin

```
Write ('Введите первое число'); Readln (A);  
Write ('Введите второе число'); Readln (B);  
If A>B then max := A  
      else max := B;  
Writeln ('Большее число ',max);
```

End.

Пример выполнения (пусть и с ошибкой)

```
Program Big1;
```

```
Var A, B : real;
```

Begin

```
Write ('Введите первое число'); Readln (A);  
Write ('Введите второе число'); Readln (B);  
If A>B then Writeln ('Первое больше');  
      else Writeln ('Второе больше');
```

End.

По вариантам

1. Даны 2 числа. Если сумма этих чисел равна 0, вывести слово «да», в противном случае вывести «нет».
2. Дано x . Удовлетворяет ли условию $|x| \geq 8$ число x .
3. Если число, введенное с клавиатуры меньше 5, то вывести «Меньше 5».
4. Даны 2 числа. Являются ли они равными.
5. Дано число. Является ли оно положительным.
6. Даны 3 числа. Если произведение этих чисел равно 0, вывести «Один равен 0».
7. Дано целое число. Если оно равно 7, то вывести «Воскресенье».
8. Дано натуральное число. Если оно меньше или равно 12, то вывести «Это номер месяца», иначе вывести «Это не номер месяца»
9. Даны 2 натуральных числа. Если их сумма меньше 10, то вывести «Это цифра», иначе вывести «Это не цифра».
10. Дано число. Определить какое оно: отрицательное, положительное или ноль.

3. Итог урока.

4. Домашнее задание: Записи в тетради, задачи по вариантам.

Урок 17 Тема: Самостоятельная работа по теме «Условный оператор»

Цель: Контроль знаний учащихся

I вариант

1. Запишите операторы условного перехода.
2. Какое значение примет переменная a после выполнения фрагмента программы?

```
a:=4;
```

```
If a<0 then a:=a+1 else a:=a-1;
```

```
Write(a);
```

3. Известно значение x . Вычислить y , если

$$y = \begin{cases} x + 1, & \text{если } x > 0 \\ x - 1, & \text{если } x \leq 0. \end{cases}$$

4. Даны координаты точки. Выяснить лежит ли эта точка во второй координатной четверти.

5. Дано двузначное число n . Определить больше ли числа 9 сумма его цифр?

II вариант

1. Запишите оператор выбора.
2. Какое значение примет переменная c после выполнения фрагмента программы?

```
c:=7;
```

```
If c > 0 then c:=c+3 else c:=c-2;
```

```
Write(c);
```

3. Известно значение x . Вычислить y , если

$$y = \begin{cases} 1 - x, & \text{если } x < 0 \\ 1 + x, & \text{если } x \geq 0. \end{cases}$$

4. Даны координаты точки. Выяснить лежит ли эта точка в третьей координатной четверти.

5. Дано двузначное число n . Определить кратна ли трем сумма его цифр?

Урок 18 – 19 Тема: Решение задач с использованием оператора выбора

Цель: научиться решать задачи с использованием оператора выбора

1. Анализ самостоятельной работы
2. Новый материал

Оператор выбора case

Оператор case позволяет сделать выбор между несколькими вариантами:

```
case <переключатель> of
    <список_констант> : <один_оператор>;
    [<список_констант> : <один_оператор>;]
    [<список_констант> : <один_оператор>;]
    [else <один_оператор>;]
end;
```

Замечание: Обратите внимание, что после else двоеточие не ставится.

Существуют дополнительные правила, относящиеся к структуре этого оператора:

1. Переключатель должен относиться только к порядковому типу данных, но не к типу longint.
2. Переключатель может быть переменной или выражением.
3. Список констант может задаваться как явным перечислением, так и интервалом или их объединением.
4. Повторение констант не допускается.
5. Тип переключателя и типы всех констант должны быть совместимыми¹⁾.

Пример оператора выбора:

```
case symbol(* :char *) of
    'a'..'z', 'A'..'Z' : writeln("Это латинская буква");
    'a'..'я', 'A'..'Я' : writeln("Это русская буква");
    '0'..'9' : writeln("Это цифра");
    '',#10,#13,#26 : writeln("Это пробельный символ");
    else : writeln("Это служебный символ");
end;
```

Выполнение оператора case происходит следующим образом:

1. вычисляется значение переключателя;
2. полученный результат проверяется на принадлежность к тому или иному списку констант;
3. если такой список найден, то дальнейшие проверки уже не производятся, а выполняется оператор, соответствующий выбранной ветви, после чего управление передается оператору, следующему за ключевым словом end, которое закрывает всю конструкцию case.
4. если подходящего списка констант нет, то выполняется оператор, стоящий за ключевым словом else. Если else-ветви нет, то не выполняется ничего.

3. Решение задач.

1. Написать алгоритм, позволяющий получить словесное наименование школьных оценок.
2. Написать алгоритм, классифицирующий треугольники (остроугольные, прямоугольные, тупоугольные), если даны углы.
3. Написать алгоритм, который по номеру дня недели - целому числу от 1 до 7 выдавать в качестве результата количество уроков в классе в соответствующий день.
4. Написать алгоритм нахождения числа дней в месяце, если даны: Номер месяца n - целое число a , равное 1 для високосного года и равное 0 в противном случае.
5. По номеру дня недели вывести его название.
6. В зависимости от того введена ли открытая скобка или закрытая, напечатать "открытая круглая скобка" или "закрытая фигурная скобка". (Учитывать круглые, квадратные, фигурные скобки).
7. В зависимости от введённого символа L, S, V программа должна вычислять длину окружности; площадь круга; объём цилиндра.
8. По введённому числу от 0 до 15 вывести название цвета, соответствующего этому коду.
9. Определить, является ли введенная буква русского алфавита гласной.
10. Написать программу, которая бы предлагала меню для вывода графических объектов, и рисовала бы соответствующий выбор.
11. Придумайте шуточный тест с выдачей шуточных результатов.
12. Напишите программу, которая по введённому числу из промежутка 0..24, определяет время суток.

4. Итог урока.

5. Домашнее задание: По вариантам.

Урок 20 Тема: Самостоятельная работа по теме «Оператор выбора»

Цель: Контроль знаний учащихся

1. Составить программу, которая по введённому возрасту сообщает пользователю, к какой возрастной группе он относится:

до 13 лет – детство

от 14 до 24 – молодость

от 25 до 59 – зрелось

от 60 до 150 – старость

2. Работа светофора для пешеходов запрограммирована следующим образом: в начале каждого часа в течении трех минут горит зеленый сигнал, затем в течении одной минуты – желтый, затем в течении двух минут – красный. С начала работы светофора прошло t мин. Какой свет горит?

Урок 21 – 22 Тема: Решение задач с использованием процедур и функций

Цель: научиться составлять программы с процедурами и функциями

1. Анализ самостоятельной работы

2. Новый материал

Описание процедур

Procedure Имя(формальные параметры)

Описательная часть процедуры

Begin

Исполнительная часть процедуры

End;

Описание функций

Function Имя(формальные переменные):типрезультата;

Описательная часть функции

Begin

Исполнительная часть функции

Имяфункции := Результат

End;

3. Решение задач.

1. Даны три целых числа. Определить, у которого из них больше сумма цифр. Подсчет суммы цифр организовать через подпрограмму.
2. По координатам вершин треугольника вычислить периметр этого треугольника. Для вычисления длины отрезка использовать подпрограмму.
3. Составить программу нахождения НОД четырёх натуральных чисел.
4. Составить программу нахождения НОК трёх натуральных чисел.
5. Написать программу нахождения суммы большего и меньшего из 3 чисел.
6. Треугольник задан координатами своих вершин. Составить программу вычисления его площади.
7. Вычислить площадь правильного шестиугольника со стороной a , используя подпрограмму вычисления площади треугольника.
8. Даны две дроби $\frac{A}{B}$ и $\frac{C}{D}$ (A, B, C, D – натуральные числа). Составить программу деления одной дроби на другую. Ответ должен быть несократимой дробью.
9. Даны две дроби $\frac{A}{B}$ и $\frac{C}{D}$ (A, B, C, D – натуральные числа). Составить программу умножения одной дроби на другую. Ответ должен быть несократимой дробью.
10. Даны две дроби $\frac{A}{B}$ и $\frac{C}{D}$ (A, B, C, D – натуральные числа). Составить программу вычитания первой дроби из второй. Ответ должен быть несократимой дробью.
11. Даны две дроби $\frac{A}{B}$ и $\frac{C}{D}$ (A, B, C, D – натуральные числа). Составить программу сложения этих дробей. Ответ должен быть несократимой дробью.
12. Написать программу, которая выводит на экран все четырёхзначные числа вида $abcd$, для которых выполняется:
 - а) a, b, c, d – разные цифры;
 - б) $ab - cd = a + b + c + d$.
13. Имеется часть катушки с автобусными билетами. Номер билета шестизначный. Составить программу, определяющую количество счастливых билетов на катушке, если меньший номер билета – N , – больший M (билет является счастливым, если сумма первых трёх его цифр равна сумме последних трёх).
14. Дано натуральное число n . Выяснить, можно ли его представить в виде произведения трёх последовательных чисел.
15. Написать программу, определяющую сумму трёхзначных чисел, содержащих только нечётные цифры. Определить также, сколько четных цифр в найденной сумме.

4. Итог урока.

5. Домашнее задание: Записи в тетради, задачи по вариантам.

Урок 23 Тема: Самостоятельная работа по теме «Процедуры и функции»

Цель: Контроль знаний учащихся

1. Найти сторону квадрата
2. Найти площадь кольца
3. Найти площадь замкнутой фигуры

Сторона квадрата

```
Program ST_Kvadrata;
Var x,n1,n2:real;
Procedure Pl(S:real; Var x:real);
Begin
  x:=Sqrt(S);
  Writeln('Сторона = ',x:4:5);
End;
Begin
  Writeln ('Введите 1 число');
  Readln (n1);
  Writeln ('Введите 2 число');
  Readln (n2);
  PL(n1,x);
  Pl(n2,x);
  Readln;
End.
```

Площадь кольца

```
Program Zадание_2;
Var r1,r2, S, s1, s2:real;
Procedure S_Okr(r:real; S:real);
Begin
  S:=3.14*r*r;
End;
begin
  readln(r1);
  readln(r2);
  S_Okr(r1,s1);
  S_Okr(r2,s2);
  S:=s1 - s2;
  Writeln('S = ',S:4:2);
  Readln;
end.
```

Площадь замкнутой фигуры

```
Program Zадание_1;
Var A1, B1,A2,B2, R, P1, P2:real;
Procedure S_Pr(a,b:real; P:real);
Begin
  P:=a*b;
End;
begin
  readln(A1,B1);
```

```

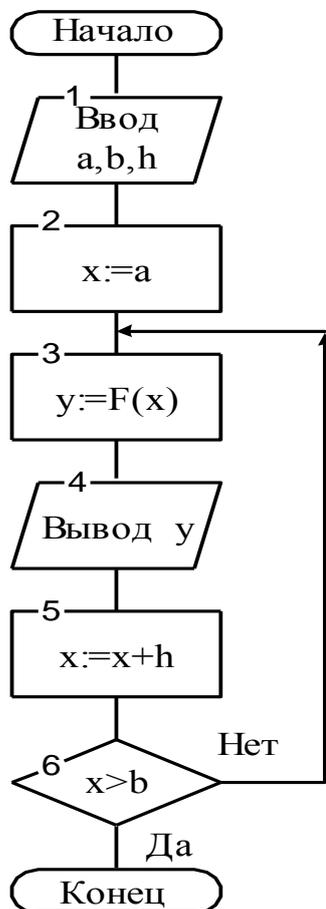
readln(A2,B2);
S_Pr(A1,B1,P1);
S_Pr(A2,B2,P2);
R:=P1 - P2;
Writeln('R = ',R:4:2);
Readln;
end.

```

Урок 24 Тема: Циклы: знакомство (цикл с постусловием)

Цель: Познакомится с использованием оператора цикла с постусловием

1. Анализ самостоятельной работы
2. Новый материал



В этой программе оператор цикла используется для многократного выполнения группы операторов, расположенных между словами *repeat*, *until*. Каждый раз в цикле вычисляется значение y , выводятся x и y , задается новое значение x и проверяется, не выходит ли x за пределы интервала. В результате работы этой программы будут напечатаны в два столбика значения x и y .

В вышеприведенном примере как раз используется оператор цикла с постусловием. Синтаксис этого оператора следующий:

repeat <операторы> *until* <условие>

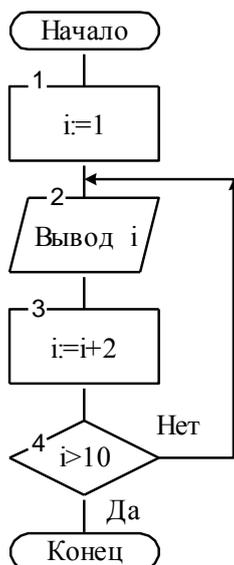
Здесь *repeat*, *until* – ключевые слова (перев. с *англ.* повторять и до тех пор пока);

<операторы> – любые операторы Паскаля (их называют телом цикла);

<условие> – логическое выражение типа сравнения, используемое для выхода из цикла. Оператор работает следующим образом: сначала выполняются операторы, расположенные в теле цикла, затем вычисляется условие, и если получается истинное значение, то осуществляется выход из цикла. Если значение выражения ложно, то выполнение операторов тела цикла повторяется, а затем снова проверяется условие.

И так, операторы тела цикла выполняются хотя бы раз, а потом все зависит от условия выхода из цикла. Очевидно, один из операторов тела цикла должен влиять на значение условного выражения, поскольку иначе цикл будет повторяться бесконечно.

Проиллюстрируем использование оператора цикла с постусловием на примере, в котором выводятся нечетные числа, меньшие 10. Схема алгоритма приведена, а программа в примере – *pr10*.



```

program pr10;
  var i:integer;
  begin
    i:=1;
    repeat
      writeln(i);
      i:=i+2;
    until i>10;
  end.
  
```

В результате работы этой программы будут напечатаны в столбик все нечетные числа от 1 до 9.

3. Решение задач.

Вычислить сумму ряда $s = 1 + 1/2 + 1/3 + 1/4 + \dots$

4. Итог урока.

5. Домашнее задание: Записи в тетради.

Урок 25 – 26 Тема: Циклы: решение задач

Цель: Научится решать задачи с использованием циклов

1. Новый материал

В **циклах со счетчиком** выполнение тела цикла должно повторяться заранее определенное число раз. Хотя такой цикл можно организовать с помощью оператора цикла с постусловием или предусловием (они универсальны), в языке Паскаль имеется специальная конструкция для организации циклов со счетчиком. Синтаксис оператора цикла со счетчиком следующий:

```
for <n.ц.> := <н.з> to <к.з.> do <оператор>;
```

Здесь *for*, *to*, *do* – ключевые слова (перев. с *англ.* для, к, выполнить соответственно);

<n.ц.> – переменная цикла, которая может быть только порядкового типа;

<н.ц.> – начальное значение, выражение такого же порядкового типа, как и переменная цикла;

<к.з.> – конечное значение, выражение такого же порядкового типа, как и переменная цикла;

<оператор> – любой оператор Турбо-Паскаля, в том числе и составной.

Оператор работает следующим образом: сначала вычисляется выражение, соответствующее начальному значению, и присваивается переменной цикла, потом проверяется условие <n.ц.> <= <к.з.> и, если получается истинное значение, то выполняется оператор, являющийся телом цикла, а затем переменная цикла увеличивается на 1 и снова проверяется условие. Если значение выражения ложно, то осуществляется выход из цикла.

Если начальное значение переменной цикла больше конечного значения, то операторы тела цикла не выполняются. Можно сказать, что оператор цикла со счетчиком – это оператор цикла с предусловием. Таким образом, следующий оператор не приведет к выполнению каких-либо операций:

```
for i:=1 to 0 do writeln(i);
```

Следующий оператор распечатает целые числа от 1 до 10:

for i:=1 to 10 do writeln(i);

Если нужно выполнить несколько операторов в теле цикла, то пользуются составным оператором:

```
for i:=5 to 10 do
begin
k:=i*i;
writeln('k=',k);
end;
```

Оператор цикла со счетчиком имеет вторую форму представления, которая позволяет изменять переменную цикла с шагом минус один. В этом случае синтаксис оператора следующий:

```
for <п.ц.>:=<н.з.> downto <к.з.> do <оператор>;
```

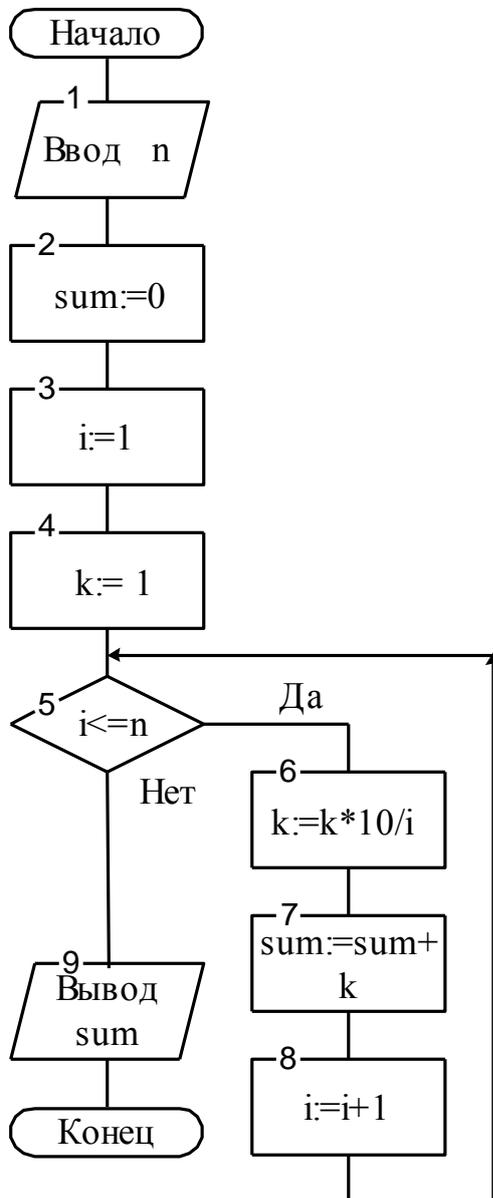
Следующий оператор распечатает целые числа от -10 до 10:

```
for i:= 10 downto -10 do writeln(i);
```

Пример полной программы рассмотрим на решении задачи нахождения суммы n первых членов ряда:

$$s = \frac{10}{1} + \frac{10^2}{1*2} + \frac{10^3}{1*2*3} + \dots$$

Схема алгоритма приведена на рис.13, а программа представлена в примере *pr13*.



```
program pr13;
var
i,n:integer;
sum,{сумма ряда}
k:real,{очередной
член ряда}
begin
sum:=0;
i:=1;
writeln('Введите число
членов ряда:');
read(n);
k:=1;
for i:=1 to n do
begin
k:=k*10/i;
sum:=sum+k;
end;
writeln('Сумма ряда
равна ',sum);
end.
```

В отличие от оператора цикла с постусловием оператор **цикла с предусловием** вычисляет и проверяет условие до выполнения операторов, составляющих тело цикла. В остальном эти операторы похожи. Синтаксис этого оператора следующий:

```
while <условие> do <оператор>;
```

Здесь *while*, *do* – ключевое слово (перев. с *англ.* пока и делать);

<оператор> – любой оператор Турбо-Паскаля, в том числе и составной (этот оператор называют телом цикла);

<условие> – логическое выражение типа сравнения, используемое для выхода из цикла.

Оператор работает следующим образом: сначала вычисляется условие, и если получается истинное значение, то выполняется оператор, являющийся телом цикла, а затем снова проверяется условие. Если значение условного выражения ложно, то осуществляется выход из цикла. Таким образом, если условие было ложно при первом входе в цикл, то операторы тела цикла не выполняются ни разу. Очевидно, один из операторов тела цикла должен влиять на значение условного выражения, поскольку иначе цикл будет повторяться бесконечно. Например, следующий фрагмент программы будет печатать радостное сообщение бесконечное число раз, так как отсутствуют в цикле конструкции, изменяющие величину *i*:

```
i:=1;  
while i<5 do  
  writeln('Доброе утро!');
```

Чтобы получить работающий фрагмент программы, добавим в тело цикла оператор, увеличивающий значение *i*:

```
i:=1;  
while i<5 do  
  begin  
    writeln('Доброе утро!');  
    i:=i+1;  
  end;
```

end;

Проиллюстрируем использование оператора цикла с предусловием на примере *pr11* нахождения среднего арифметического последовательности чисел. Последовательность чисел вводится с клавиатуры и завершается стоп-кодом. Использование стоп-кода в данном случае состоит в том, что какое-то числовое значение заведомо исключается из входной последовательности и используется как стоп-код. Если мы заранее знаем, что *-1* у нас никогда не появится в последовательности, то можно *-1* использовать для указания ее конца.

```
program pr11;  
  const    stopcod=-1;  
  var c: integer; {количество вводимых чисел}  
      sum, {сумма вводимых чисел}  
      number: real; {вводимое число}  
  begin  
    sum:=0;  
    c:=0;  
    writeln('Введите первое число последовательности:');  
    read(number);  
    while number<>stopcod do  
      begin  
        sum:=sum+number;  
        c:=c+1;  
        writeln('Введите следующее число  
                последовательности:');
```

```

    read(number);
end;
if c=0 then writeln ('Среднее значение равно нулю')
    else writeln ('Среднее значение равно ',sum/c);
end.

```

Необходимо отметить, что перед первым входением в цикл *while* значение *number* должно быть прочитано, иначе *number* не будет определено при первой проверке условия *while*. Если первым вводимым числом оказался стоп-код, то тело цикла не выполнится ни разу, счетчик *c* останется равным нулю, и чтобы не произошло деление на нуль используется условный оператор.

Кроме того, в этой программе впервые появился раздел констант. Константа *stopcod*, в отличие от переменной, не может получить новое значение в программе, и тип константы определяется её видом.

Следующий пример иллюстрирует работу оператора цикла с предусловием на задаче, вычисляющей с точностью *e* сумму ряда:

$$s=1+1/2+1/3+1/4+ \dots$$

Вычислить сумму ряда с точностью *e* – это значит завершить суммирование членов ряда тогда, когда очередной член ряда окажется меньше *e* по абсолютной величине.

Программа – в примере *pr12*.

```

program pr12;
var
    i:integer;
    sum,{сумма ряда}
    e,{точность}
    k:real;{очередной член ряда}
begin
    sum:=0;
    i:=1;
    writeln('Введите точность:');
    read(e);
    k:=1/i;
    while k>e do {пока очередной член ряда больше точности}
        begin
            sum:=sum+k;
            i:=i+1;
            k:=1/i;
        end;
    writeln ('Сумма ряда равна ', sum);
end.

```

2. Решение задач.

1. 2. Задачи. Вывести на экран целые числа в следующем виде:

```

7           7 6 5 4 3 2
7 6         6 5 4 3 2
7 6 5       5 4 3 2
7 6 5 4     4 3 2
7 6 5 4 3   3 2
7 6 5 4 3 2 2
7 6 5 4 3 2 1

```

- Задача. Дано натуральное число *n*. Найти все тройки натуральных чисел *x*, *y*, *z*, если они существуют, такие, что $x^2 + y^2 + z^2 = n$
- Задача. Даны натуральные числа *a*, *b* ($a < b$). Получить все простые числа *p*, удовлетворяющие неравенству: $a \leq p \leq b$
- Дано натуральное число *n*. Найти сумму первой и последней цифры этого числа.

6. Дано натуральное число n . Переставить местами первую и последнюю цифры этого числа.
7. Дано натуральное число n . Проверить, есть ли в записи числа три одинаковые цифры. ($n < 10000$).
8. Дано натуральное число $n < 100$. Дописать к нему цифру k в начало и конец.
9. Найти сумму делителей каждого из целых чисел от 50 до 70.
10. Найти все натуральные числа a, b, c из интервала от 1 до 40, для которых выполняется равенство: $a^2 + b^2 = c^2$.
11. Вычислить сумму: $s = \frac{1}{5} + \frac{3}{8} + \frac{5}{11} + \dots + \frac{26}{40}$.

12. . Найти все целочисленные решения системы
$$\begin{cases} |x| \leq 10 \\ |y| \leq 15 \\ x^2 + y^2 = 25. \end{cases}$$

9.

```
for i:=50 to 70 do Begin
  S:=0;
  for a:=1 to i do
    if i mod a =0 then S:=S+a;
  Writeln('Число ',i,' Сумма делителей = ',S);
end;
```

10.

```
for a:=1 to 40 do
  for b:=1 to 40 do
    for c:=1 to 40 do
      if a*a+b*b=c*c Then Writeln(a:3,b:3,c:3);
```

11.

```
S:=0;i:=1;z:=5;
While i<=26 do Begin
  S:=s+i/z;
  i:=i+2;
  z:=z+3;
end;
Writeln('S = ',S);
```

12.

```
for x:=-10 to 10 do
  for y:=-15 to 15 do
    if x*x+y*y = 25 then Writeln(x:3,y:3);
```

3. Итог урока.

4. Домашнее задание: По вариантам.

Варианты

1. Дано натуральное число N . Отбросить у этого числа первую и последнюю цифры. Например, было число 45268, станет 526.
2. Дано натуральное число N . Увеличить первую и последнюю цифру этого числа в 10 раз. Например, было число 234, станет 20340.
3. Дано натуральное число N . Проверить есть ли в записи числа цифры m и p .
4. Дано натуральное число N . Дописать к нему цифру m два раза в начало. Например, было число 234, цифра $m = 5$, станет 55234.

Урок 27 Тема: Самостоятельная работа по теме «Циклы»

Цель: Контроль знаний учащихся

1 вариант

1. Может ли тело цикла с параметром не выполниться ни разу? Когда?
2. Как оформляется цикл с предусловием. В каких случаях используется данный цикл.
3. Сколько раз повторится цикл, какие значения примут переменные, после выполнения программы
 $x:=2;$
While $x \leq 5$ do
begin
Writeln(x);
 $x:=x+3;$
end;
writeln(x);

4. Найти все целочисленные решения

$$\text{системы } \begin{cases} |x| \leq 5 \\ |y| \leq 9 \\ x^2 + y = 10. \end{cases}$$

5. Найти сумму $S = \frac{1}{3} + \frac{3}{5} + \frac{5}{7} + \dots + \frac{11}{13}$.

6. Найти все двузначные числа, сумма квадратов цифр которых делится на 15.

2 вариант

1. Что такое цикл?
2. Как оформляется цикл с параметром. В каких случаях используется данный цикл.
3. Сколько раз повторится цикл, какие значения примут переменные, после выполнения программы
 $a:=-2;$
While $a \leq 3$ do
begin
Writeln(x);
 $a:=a+2;$
end;
writeln(a);

4. . Найти все целочисленные решения

$$\text{системы } \begin{cases} |x| \leq 7 \\ |y| \leq 5 \\ x + y^3 = 30. \end{cases}$$

5. Найти сумму $S = \frac{2}{3} + \frac{5}{6} + \frac{8}{9} + \dots + \frac{20}{21}$.

6. Найти все трехзначные числа, сумма цифр которых нечетна.

Урок 28 Тема: Графика в Паскале. Графические примитивы, работы учащихся

Цель: Научится рисовать в Паскале

1. Анализ самостоятельной работы

2. Новый материал

В языке turbo Pascal имеется значительное количество графических процедур и функций. Нам понадобятся лишь некоторые из них. Для того, чтобы компилятор "узнавал" их названия, мы должны после заголовка программы разместить строчку следующего вида:

```
uses Graph;
```

(что в переводе на русский означает "используется графика").

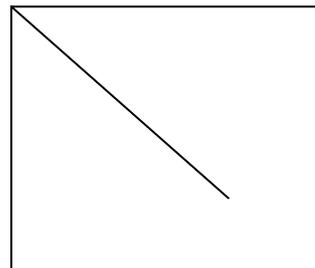
До сих пор во время нашей работы за компьютером экран всегда находился в текстовом режиме, поэтому на экране можно было видеть только лишь символы (правда, включая такие изыски, как так называемые "символы псевдографики").

Для рисования прямых, окружностей и пр. необходимо перевести экран в графический режим. Для включения графического режима используется процедура InitGraph.

Простейшая программа может иметь вид:

Пример 1.

```
PROGRAM Primer_1;
  uses Graph;
  var Gd,Gm: Integer;
BEGIN
  Gd:=VGA; { Графический адаптер - VGA      }
  Gm:=VGAHi; { Графический режим VGAHi (640x480) }
  InitGraph (Gd,Gm,..\bgi'); { Включить графический режим }
  If GraphResult=grOk
  then { Если режим включился успешно }
    { Нарисовать отрезок прямой }
    Line (0,0,639,479); ReadLn
END.
```



Мы видим, что у процедуры `InitGraph` три параметра. В качестве первых двух фактических параметров должны стоять имена целых (`integer`) переменных. Не будем вдаваться в подробности, почему это так; вместо этого выясним их предназначение.

Первый параметр является кодом графического адаптера (т.е. электронной схемы, управляющей выводом информации на экран). (Дело в том, что на IBM-совместимых компьютерах применяется ряд стандартных графических адаптеров, носящих названия CGA, EGA, VGA.)

По нашей программе можно догадаться, что в используемых нами компьютерах используется адаптер VGA, и компилятор сам "узнает" слово VGA и заменит его на нужное целое число (на самом деле равное 9, но мы этого можем и не запоминать).

Каждый графический адаптер позволяет использовать несколько графических режимов, отличающихся количеством цветов и разрешающей способностью (в дальнейшем мы узнаем, что это такое). И второй из параметров как раз предназначен для того, чтобы указать, какой из графических режимов следует включить. Пока что мы ограничимся лишь одним графическим режимом VGAHi.

Третий параметр является строкой, содержащей путь к файлу, который называется EGA\VGA.BGI. В этом файле содержится драйвер (такая специальная программа), необходимый для работы с адаптерами EGA и VGA. И, как легко увидеть из нашего примера, файл этот находится в подкаталоге TPBGI.

Все вышеизложенное необходимо знать каждому грамотному пользователю IBM-совместимых компьютеров. Однако в нашей лабораторной работе достаточно использовать конструкцию, использованную в первом примере, для включения графического режима. (И не страшно, если в ней не все понятно.)

Для того, чтобы мы могли что-либо нарисовать на экране, нам нужно уметь задавать положение на экране того, что мы рисуем. Для этого с экраном связывается система координат следующего вида:

```
(0,0)                X
+----->
|
|
|
Y|
```

Каждая точка на экране на самом деле представляет собой очень маленький прямоугольник (и поскольку это не совсем точка, то иногда используют специальный термин - "пиксел"). Количество точек (пикселов), умещающихся на экране по вертикали и горизонтали, называют разрешающей способностью. Разрешающая способность экрана в

режиме VGAHi - 640x480. Это означает, что по горизонтали на экране умещается 640 точек, а по вертикали - 480.

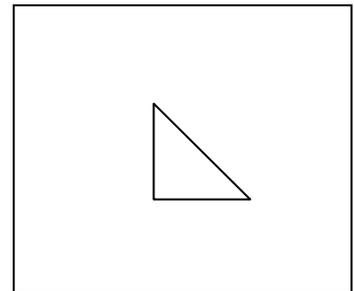
Точка в левом верхнем углу экрана имеет координаты (0,0). Координата X любой точки экрана лежит в пределах от 0 до 639, а координата Y - в пределах от 0 до 479.

Как Вы уже догадались, процедура Line (x1,y1,x2,y2) рисует на экране прямую, соединяющую точки (x1,y1) и (x2,y2).

Пример 2.

Изобразить на экране прямоугольный треугольник с вершинами (320, 10), (120,210), (520,210).

```
PROGRAM Primer_2;
uses Graph;
var Gd,Gm: Integer;
BEGIN
Gd:=VGA; Gm:=VGAHi; InitGraph (Gd,Gm,'.\bgi');
If GraphResult=grOk
then begin
Line (120,210,520,210); { Горизонтальный отрезок }
Line (120,210,320,10); { Левый катет }
Line (320,10,520,210); { Правый катет }
ReadLn
end;
END.
```



Довольно обидно работать на цветном мониторе только с черно-белыми

изображениями. Для задания цвета рисования прямых, окружностей, точек и пр. используется процедура SetColor. В качестве единственного ее параметра нужно указать целое число, являющееся кодом цвета. Цвета кодируются в соответствии со следующей

лькдовой таблицей: Black=0-черныйDarkGray=8-темно-серый
Blue=1-синийLightBlue=9-голубо
Green=2-зеленыйLightGreen=10- светло-зеленый
Cyan=3-цвет морской волныLightCyan=11-светло-циановый
Red=4-красныйLightRed=12-розовый
Magenta=5-сиреневыйLightMagenta=13-светлосиреневый
Brown=6-коричневыйYellow=14- желтый
LightGray=7-светло-серыйWhite=15-белый

Если Вы хорошо знаете английский язык, то Вам будет удобнее использовать не числа, а соответствующие им идентификаторы; если же Вы английский знаете плохо, то все равно полезнее запомнить английские названия цветов, чем запоминать числа, кодирующие цвета.

Пример 3.

Изобразить тот же треугольник, что и в предыдущем примере, но сделать стороны треугольника разноцветным.

```
PROGRAM Primer_3;
uses Graph;
var Gd,Gm: Integer;
```

```

BEGIN
  Gd:=VGA; Gm:=VGAHi; InitGraph (Gd,Gm,'..\bgi');
  If GraphResult=grOk
  then begin
    SetColor (LightMagenta); { Цвет - светло-сиреневый }
    Line (120,210,520,210); { Горизонтальный отрезок }
    SetColor (LightCyan); { Цвет - светло-циановый }
    Line (120,210,320,10); { Левый катет }
    Set Color (Green); { Цвет - зеленый }
    Line (320,10,520,210); { Правый катет }
    ReadLn
  end
END.

```

Пример 4

Разноцветные лучи.

```

PROGRAM Primer_4;
uses Graph;
const CenterX=320;
      CenterY=240;
      Radius=200;
var Gd,Gm: Integer;
    i : Integer;
    dx,dy: Integer;
BEGIN
  Gd:=VGA; Gm:=VGAHi; InitGraph (Gd,Gm,'..\bgi');
  If GraphResult=grOk
  then begin
    For i:=0 to 160 do
    begin
      dx:=Round (Radius*sin(i*pi/80));
      dy:=Round (Radius*cos(i*pi/80));
      SetColor (i MOD 16);
      Line(CenterX,CenterY,CenterX+dx,CenterY+dy)
    end;
    ReadLn
  end
END.

```

Пример 5

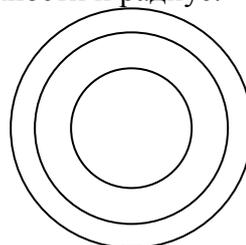
Концентрические окружности.

Для рисования окружностей используется процедура Circle с тремя целочисленными параметрами, задающими координаты центра окружности и радиус.

```

PROGRAM Primer_5;
uses Graph;
const CenterX=320;
      CenterY=240;
var Gd,Gm: Integer;
    i : Integer;

```



```

BEGIN
  Gd:=VGA; Gm:=VGAHi; InitGraph (Gd,Gm,'..\bgi');
  If GraphResult=grOk
  then begin
    For i:=0 to 23 do
      Circle (CenterX,CenterY,i*10);
    ReadLn
  end
END.

```

Пример 6

Разноцветные концентрические окружности.

Для закраски замкнутой области используется процедура FloodFill, три целочисленных параметра которой задают начальную точку закраски и код цвета ограничивающей область линии.

Цвет, которым будет производиться закраска, ничего общего не имеет с цветом, задаваемым процедурой SetColor. Цвет закраски задается вторым параметром процедуры SetFillStyle. Первый параметр этой процедуры (задающий узор для закраски) на первых порах будем задавать равным единице (что соответствует сплошной закраске).

```

PROGRAM Primer_6;
uses Graph;
const CX=320;
      CY=240;
var Gd,Gm: Integer;
    i : Integer;
BEGIN
  Gd:=VGA; Gm:=VGAHi; InitGraph (Gd,Gm,'..\bgi');
  If GraphResult=grOk
  then begin
    For i:=0 to 23 do
      Circle (CX,CY,i*10);
    For i:=0 to 23 do
      begin
        SetFillStyle (1,i MOD 16);
        { Закрашивать до границы белого цвета }
        FloodFill (CX,CY+i*10-5,White)
      end;
    ReadLn
  end
END.

```

3. Итог урока.

4. Домашнее задание: Подготовить свой рисунок.

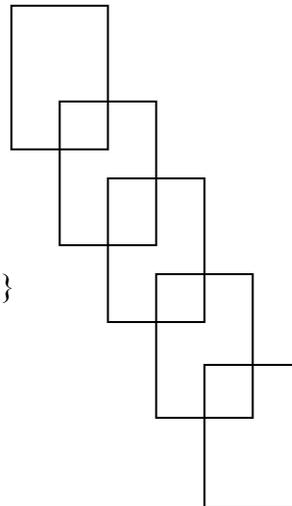
Урок 29 Тема: Использование процедур при рисовании

Цель: Научится рисовать используя пользовательские процедуры

1. Проверка домашнего задания
2. Новый материал

Пример 7.

```
PROGRAM Primer_7;
uses Graph;
var grDriver: Integer;
    grMode : Integer;
    i,x,y : Integer;
{ ----- }
PROCEDURE Rect (x,y,x1,y1: Integer);
{ Рисует прямоугольник, у которого левый нижний угол }
{ имеет координаты (x,y), а правый верхний - (x1,y1) }
BEGIN
  Line (x,y,x,y1); { Левая сторона }
  Line (x1,y,x1,y1); { Правая сторона }
  Line (x,y1,x1,y1); { Верхняя сторона }
  Line (x,y,x1,y) { Нижняя сторона }
END;
{ --- }
BEGIN
  GrDriver:=VGA; GrMode:=VGAHi;
  InitGraph (grDriver,grMode, '..\bgi');
  If GraphResult=grOk
  then begin
    For i:=1 to 15 do
    begin
      x:=i*30; y:=i*25; SetColor (i);
      Rect (x,y,x+50,y+60)
    end;
    ReadLn
  end
END.
```



Рисование прямоугольников - часто встречающаяся проблема, и поэтому неудивительно, что существует стандартная процедура, работающая так же, как и созданная нами ниже процедура Rect. Она называется Rectangle.

Пример 8.

Для рисования "заполненных" прямоугольников используется процедура Bar. Так же, как и для процедуры Rectangle, мы должны указать четыре числа - координаты двух противоположных углов прямоугольника. (Для процедуры Bar цвет задается не с помощью SetColor, а с помощью SetFillStyle!).

```
PROGRAM Primer_8;
uses Graph;
const Step=35;
var grDriver: Integer;
```

```

    grMode : Integer;
    i,x,y : Integer;
{ ----- }
PROCEDURE Square (x,y: Integer);
{ Рисует цветастый квадрат, центр которого }
{ имеет координаты (x,y) }
    var i,d: Integer;
BEGIN
    For i:=15 downto 0 do
        begin
            SetFillStyle (SolidFill,i); d:=i*3+2;
            Bar (x-d,y-d,x+d,y+d)
        end
    END;
{ --- }
BEGIN
    GrDriver:=VGA; grMode:=VGAHi;
    InitGraph (grDriver,grMode,'.\bgi');
    If GraphResult=grOk
        then begin
            For i:=0 to 10 do
                { На "побочной" диагонали - 11 точек }
                begin
                    x:=50+i*Step; y:=50+(10-i)*Step;
                    SetColor(i); Square(x,y)
                end;
            ReadLn
        end
    END.

```

При рисовании сложных изображений, содержащих много отрезков, возникает довольно противная проблема - вычислять координаты всех точек. Если использовать процедуру LineRel, то достаточно указывать смещения по обеим координатам относительно текущей точки.

Для относительного перемещения без рисования используется процедура MoveRel Для задания начальных значений координат текущей точки используется процедура MoveTo

Пример 9.

Квадратная спираль.

```

PROGRAM Primer_9;
uses Graph;
const CenterX=320;
      CenterY=240;
      d=12;
var grDriver: Integer;
    grMode : Integer;
    i,L : Integer;
{ ----- }
PROCEDURE Vitok (L,dL: Integer);
{ Начиная от текущей точки, рисует виток спирали }
{ из четырех отрезков увеличивающейся длины }
{ L - длина первого отрезка }
{ dL - приращение длины для каждого из следующих отрезков }

```

```

BEGIN
  LineRel (L,0);      { Сдвинуться вправо }
  LineRel (0,-(L+dL)); { Сдвинуться вверх }
  LineRel (-(L+2*dL),0); { Сдвинуться влево }
  LineRel (0,L+3*dL); { Сдвинуться вниз }
END;
{ --- }
BEGIN
  grDriver:=VGA; grMode:=VGAHi;
  InitGraph (grDriver,grMode,'.\bgi');
  If GraphResult=grOk
  then begin
    { Сделать текущей точкой центр экрана }
    MoveTo (CenterX,CenterY);
    L:=1; { Начальная длина отрезка }

    For i:=1 to 10 do { 10 витков спирали }
      begin Vitok (L,d); L:=L+4*d end;
    ReadLn
  end
END.

```

Пример 10. Небольшой городок.

```

PROGRAM Domiki;
  uses Graph;
  var grDriver: Integer;
      grMode : Integer;
      i,j : Integer;
{ ----- }
PROCEDURE Domik (x,y: Integer);
{ Рисует домик, у которого левый нижний угол }
{ имеет координаты (x,y) }
  const dx=60; { Ширина фасада }
        dy=40; { Высота фасада }
        dx2=dx DIV 2;
        dy2=dy DIV 2;
        wx=16; { Ширина окна }
        wy=22; { Высота окна }
        wx2=wx DIV 2;
        wy2=wy DIV 2;
BEGIN
  Rectangle (x,y,x+dx,y-dy); MoveTo (x,y-dy);
  Linerel (dx2,-dx2); { Левый скат крыши }
  Linerel (dx2,dx2); { Левый скат крыши }
  Rectangle (x+dx2-wx2,y-dy2-wy2,x+dx2+wx2,y-dy2+wy2); { Окно }
  MoveTo (x+dx2,y-dy2); { Центр фасада (и окна) }
  LineRel (0,wy2); { Вертикальная часть рамы окна }
  MoveTo (x+dx2-wx2,y-dy2); { Центр левой части рамы окна }
  LineRel (wx,0); { Горизонтальная часть рамы окна }
  SetFillStyle (SolidFill,Red);
  FloodFill (x+1,y-1,White);
  SetFillStyle (SolidFill,Blue);

```

```

    FloodFill (x+dx2,y-dy-1,White)
END;
{ --- }
BEGIN
  grDriver:=VGA; grMode:=VGAHi;
  InitGraph (grDriver,grMode,'.\bgi');
  If GraphResult=grOk
  then begin
    For i:=1 to 6 do
      For j:=1 to 5 do
        Domik (i*80,j*80);
      ReadLn
    end
  end
END.

```

Если Вас интересуют другие графические процедуры или функции, то Вам следует обратиться к системе "помощи" (Help). Для этого в меню Help выберите пункт Standard units. Среди всего прочего Вы увидите список названий стандартных модулей (неважно, что вы не знаете, что это такое). Если установить курсор на

слово Graph (этого можно достигнуть и с помощью клавиши Tab) и нажать Enter, то на экране появится общая информация о модуле Graph. Для продолжения нам предлагается два выбора:

Go to GRAPH.TPU Functions and Procedures

Go to GRAPH.TPU Constants and Types

Если выбрать пункт "Перейти к константам и типам GRAPH.TPU", то Вы сможете добраться до такой полезной информации, как названия графических режимов и их разрешение, названия цветов, названия стилей закрашки и т.п. На случай, если Вам захочется там что-нибудь поискать, приведем перевод названий некоторых пунктов соответствующего меню:

Color Constants	Цветовые константы
Fill Pattern Constants	Константы для образцов закрашки
Graphics Drivers	Графические драйверы
Graphics Modes for Each Driver	Графические режимы для каждого драйвера

Выбрав же пункт "Перейти к процедурам и функциям GRAPH.TPU", мы увидим внушительный список названий графических процедур и функций. Если поместить курсор на название приглянувшейся Вам процедуры (или функции), и нажать "перевод строки", то Вы увидите краткое описание этой процедуры (функции), и в конце - предложение посмотреть пример. (Если, скажем, Вы читаете информацию о процедуре

Ellipse, то в самом конце Вы увидите слова

Sample Code:

Ellipse.PAS

Если установить курсор на слова Ellipse.PAS и нажать "перевод строки", то на экране Вы увидите текст программы, иллюстрирующий применение этой процедуры). Эта информация может пригодиться даже тем, кто не знает ни одного английского слова - Вы увидите, сколько у процедуры параметров и какие у этих параметров типы; Вы можете обычным образом (клавишами управления курсором при нажатой клавише Shift) отметить текст примера, скопировать в пустое окно и запустить.

Пример 11 (Е.В.Баранова, РГПУ им.А.И.Герцена).

Изобразить график функции $y = \cos(x)$.

При изображении графика функции необходимо иметь ввиду, что начало графических координат находится в левом верхнем углу экрана и что графические координаты принимают целые неотрицательные значения в диапазоне $(0, \max_x)$ и $(0, \max_y)$. Значения

max_x и max_y для каждого режима можно определить, используя соответствующие функции. Таким образом, для получения "хорошего" графика необходимо выполнить поворот и масштабирование. Пусть, x_{max} - максимальное значение по оси x ; y_{max} - максимальное значение по оси y ;

(x_0, y_0) - графические координаты центра - точки $(0,0)$;

(x_g, y_g) - графические координаты точки (x,y) ;

m_x - масштаб по оси x , т.е. величина $Abs((x_g-x_0)/x)$;

m_y - масштаб по оси y , т.е. величина $Abs((y_g-y_0)/y)$.

Графические координаты точки (x,y) : $x_g=x_0+m_x*x$; $y_g=y_0-m_y*f(x)$.

```
PROGRAM Primer_11;
```

```
Uses Graph;
```

```
var x,y,a,b,h           : Real;
```

```
    x0,y0,xg,yg,xmax,ymax,mx,my,grd,grm,c: Integer;
```

```
BEGIN
```

```
WriteLn ('Введите координаты центра: '); ReadLn (x0,y0);
```

```
WriteLn ('Введите масштаб по x и y: '); ReadLn (mx,my);
```

```
WriteLn ('Введите область задания функции по x и шаг: ');
```

```
ReadLn (a,b,h); WriteLn ('Введите цвет изображения: ');
```

```
ReadLn (c); grd:=0; grm:=0; InitGraph (grd,grm,"");
```

```
c:=getcolor; xmax:=getmaxx; ymax:=getmaxy;
```

```
Line (10,y0,xmax-10,y0); { Ось OX }
```

```
Line (x0,10,x0,ymax-10); { Ось OY }
```

```
x:=a;
```

```
While x<=b do
```

```
begin
```

```
    xg:=x0+Trunc(mx*x); yg:=y0-Trunc(my*f(x));
```

```
    If (xg>=0) AND (xg<=xmax) AND (yg>=0) AND (yg<=ymax)
```

```
        then putpixel (xg,yg,c);
```

```
    x:=x+h
```

```
end;
```

```
ReadLn;
```

```
closegraph
```

```
END.
```

3. Итог урока.

4. Домашнее задание: Изменить программу рисунка в соответствии с темой урока.

Урок 30 Тема: Процедуры и анимация в Паскале
Цель: Научится «Оживлять изображения в Паскале»

1. Новый материал

Пример 12 (Е.В.Баранова, РГПУ им.А.И.Герцена).

Изобразить движение шарика по синусоиде.

Движение реализуется с помощью процедур GetImage и PutImage. Процедура GetImage запоминает образ изображаемого объекта и образ области экрана такого же размера, закрашенной цветом фона. Процедура PutImage на каждом шаге последовательно заменяет старое изображение цветом фона и создает изображение на новом месте.

```
PROGRAM Primer_12;
{ Программа движения шарика со следом по синусоиде }
uses Graph;
var p1,p2: Pointer;
    { p1 - указатель на образ "следа",
      p2 - указатель на образ шарика }
    grm,grd,x,y,x1,y1: Integer;
    size,c          : Word;
BEGIN
    grd:=0; InitGraph (grd,grm,'D:\Гр\Bgi');
    { Инициализация графического режима с автоматическим
      определением подходящего драйвера }
    c:=GetColor; { c - цвет изображения }
    x1:=0;y1:=90; { x1,y1 - начальные координаты шарика }
    PutPixel (0,y1+5,c); { Изображение "следа" }
    size:=ImageSize(0,0,10,10); GetMem (p1,size);
    { size - количество байтов для изображения квадрата 11x11 }
    GetImage (0,y1,10,y1+10,p1^);
    { p1 указывает на область памяти с изображением следа }
    SetFillStyle (11,c); { Устанавливается тип и цвет закрашки }
    Circle (x1+5,y1+5,5); { Окружность с центром в (x1,y1) }
    FloodFill (x1+5,y1+5,c); { Закраска окружности }
    GetMem (p2,size); GetImage (x1,y1,x1+10,y1+10,p2^);
    { p2 указывает на область памяти с изображением шарика }
    For x:=1 to 300 do
        begin
            y:=Trunc (40*sin(0.1*x)+90);
            { x,y - графические координаты нового положения шарика }
            PutImage (x1,y1,p1^,0); { На месте шарика изображается след }
            PutImage (x,y,p2^,0); { Шарик изображается на новом месте }
            x1:=x; y1:=y { Запоминаются новые координаты шарика }
        end;
    ReadLn; Closegraph
END.
```

Пример 13 (Е.В.Баранова, РГПУ им.А.И.Герцена).

Управление движением объекта.

Направление движения определяется нажатой клавишей (стрелки влево, вправо, вверх, вниз). Шаг перемещения вводится. Реализация движения характеризуется тем, что на каждом

шаге запоминается образ области экрана, куда помещается курсор, затем при смещении курсора изображение восстанавливается.

```
PROGRAM Primer_13;
{ Программа управления движением курсора.
Курсор - прямоугольный объект,двигающийся вверх, вниз,
вправо, влево при нажатии соответствующих стрелок.
Выход при нажатии клавиши ESC }
uses Crt,Graph;
    { Модуль Crt необходим для использования Readkey }
PROCEDURE BadKey;
{ Процедура формирует звук при нажатии неправильной клавиши }
BEGIN
    Sound (500); Delay (100); Sound (400);
    Delay (200); Nosound
END;
var p,pc: Pointer;
{ pc - указатель на образ курсора,
p - указатель на образ изображения "под" курсором }
    grm,grd,curx,cury,curx0,cury0,lx,ly,hx,hy:integer;
    size,c:word; ch:char;
{ grd,grm - переменные для номеров графических драйверов и режима
curx,cury - координаты текущего положения курсора
curx0,cury0 - переменные для запоминания координат курсора
lx,ly - ширина и длина курсора прямоугольного вида
hx,hy - шаги движения курсора по горизонтали и вертикали }
BEGIN
    WriteLn ('Введите размеры курсора и шаги движения');
    ReadLn (lx,ly,hx,hy);
    { Установка значения системной переменной для обеспечения
совместимости работы модулей Crt и Graph }
    DirectVideo:=FALSE;
    grd:=0; InitGraph (grd,grm,'D:\Tp\Bgi');
    { Инициализация графического режима с автоматическим
определением подходящего драйвера }
    c:=GetColor; { c - цвет изображения }
    size:=ImageSize (0,0,lx,ly);
    { size - количество байтов для изображения курсора }
    GetMem (pc,size); GetMem (p,size);
    { Выделяются области для хранения образа курсора,
и образа изображения под курсором }
    SetFillStyle (1,c); { устанавливается тип и цвет закраски курсора }
    GetImage (0,0,lx,ly,p^);
    { p указывает на область памяти, где хранится изображение,
которое будет "закрыто" курсором }
    curx:=0; cury:=0;
    Bar (0,0,lx,ly); GetImage (0,0,lx,ly,pc^);
    { pc указывает на область памяти с изображением курсора }

    SetColor (6); SetFillStyle (1,2);
    Bar3d (150,150,200,30,10,TRUE);
    { Параллелограмм, на фоне которой происходит движение }
    Repeat { Цикл по вводу символа }
```

```

ch:=ReadKey; { Ввод очередного символа }
If Ord(ch)=0
then { Нажата управляющая клавиша }
begin
ch:=ReadKey;
curx0:=curx; cury0:=cury;
{ В переменных curx,cury запоминаются координаты курсора }
Case Ord(ch) of
77: If curx<getmaxx-hx
then curx:=curx+hx; { Шаг вправо }
75: If curx>hx
then curx:=curx-hx; { Шаг влево }
72: If cury>hy
then cury:=cury-hy; { Шаг вверх }
80: If cury<getmaxy-hy
then cury:=cury+hy { Шаг вниз }
else BadKey { Нажата "неправильная" клавиша }
end;
If (curx<>curx0) OR (cury<>cury0)
then begin
PutImage (curx0,cury0,p^,0);
{ Восстановить изображение, которое было "закры-
то" курсором }
GetImage (curx,cury,curx+lx, cury+ly,p^);
{ Запомнить то изображение, которое будет "зак-
рыто" курсором }
PutImage (curx,cury,pc^,0);
{ Установить курсор в новое положение }
end
end
else BadKey
until Ord(ch)=27;
CloseGraph
END.

```

2. Итог урока.

3. Домашнее задание: Анимировать элементы своего рисунка.

Урок 31 Тема: Решение задач на графику

Цель: Отработать навык рисования в Паскале, проверить работу своей программы (рисунка)

1. Проверка домашнего задания

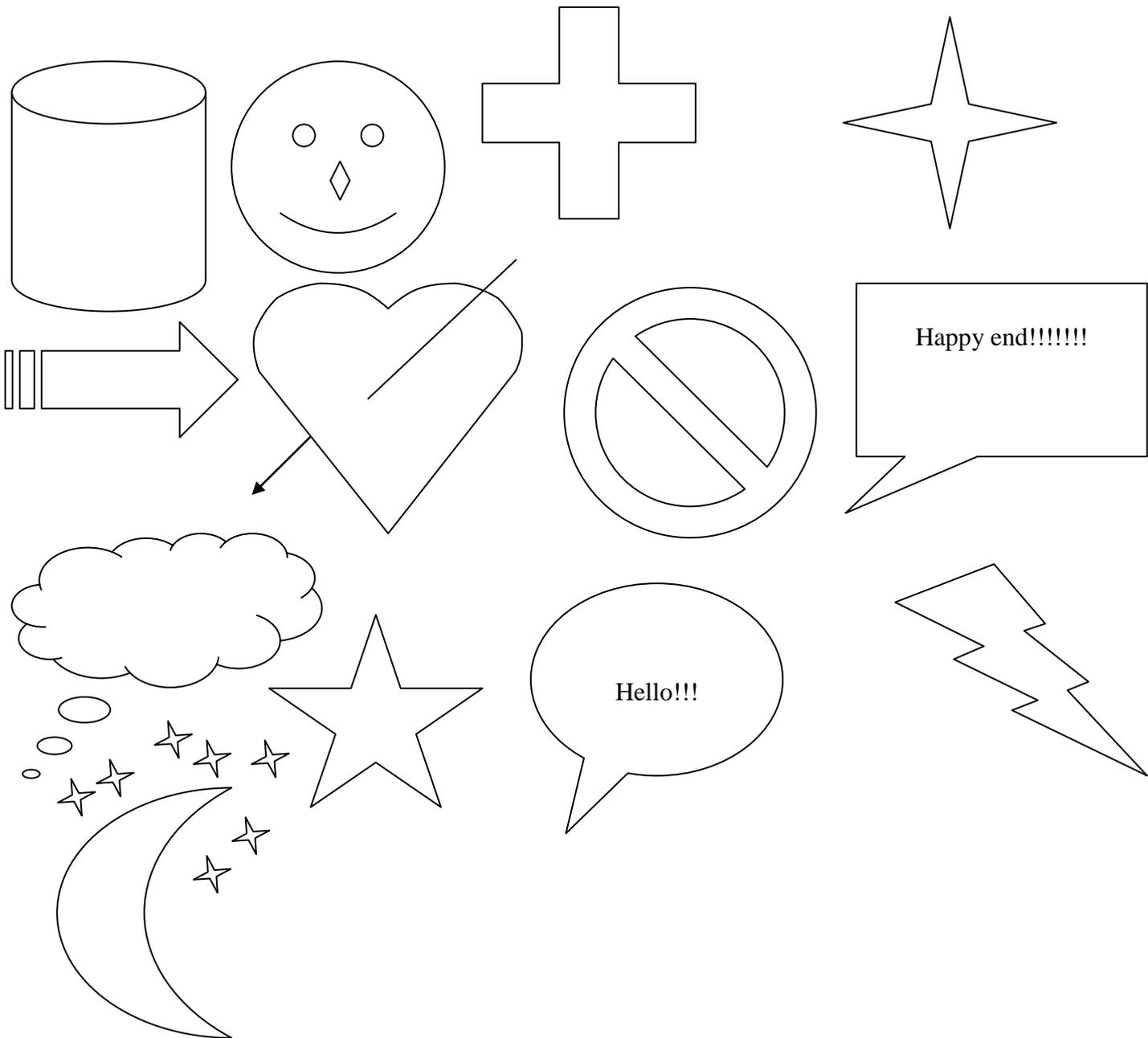
2. Итог урока.

3. Домашнее задание: Выучить процедуры рисования

Урок 32 Тема: Самостоятельная работа по теме «Графика»

Цель: Контроль знаний учащихся

Анимировать изображения (по вариантам)



Резерв времени 4 часа можно использовать по своему усмотрению. По ходу выполнения программы добавлять время на ту или иную тему.