Управление образования и науки Тамбовской области

Тамбовское государственное образовательное учреждение среднего профессионального образования "Тамбовский политехнический техникум им. М.С.Солнцева"

Пособие по информатике

раздел «Программирование»

Разработал:
преподаватель математики и информатики
Проскурякова И.С.

Тамбов 2012

Содержание

§1 Этапы решения задач с помощью ПК

Пусть вам надо решить какую-нибудь задачу, и вы хотите воспользоваться услугами компьютера. С чего начать? Прежде всего нужно разобраться, что в изучаемом объекте или явлении существенно для данной задачи, а чем можно и пренебречь. При этом существенные свойства объекта должны быть сформулированы так, чтобы их можно было записать на языке математических понятий (формул, уравнений, неравенств и т.д.). Выполнив такой «перевод» на язык математики, вы получите то, что называют ***математической моделью*** *задачи. Составляя математическую модель, из всей информации об объекте необходимо выделить те величины, которые будут исходными (****данные****), а также определить то, что требуется получить — **результаты**.*

Итак, создавая математическую модель для решения задачи, нужно:

1) выделить предположения, на которых будет основана математическая модель;

2) определить, что считать исходными данными и результатами;

3)записать математические соотношения (формулы, уравнения, неравенства и т.д.), связывающие результаты с исходными данными.

Созданием математической модели завершается первый этап решения задачи с помощью ПК. Но для того чтобы компьютер произвел необходимые вычисления и получил ответ, нужно составить для него четкую инструкцию – алгоритм, который должен быть записан на понятном компьютеру языке – языке программирования. Т.е. второй этап решения задач с помощь ПК – составление программы.

Проведя вычисления на ПК, необходимо проанализировать результаты. При этом может возникнуть необходимость уточнить математическую модель, поскольку может выясниться, что при ее разработке не были учтены какие-либо существенные свойства объекта. Уточнив модель, вносятся изменения в программу, проводятся расчеты на компьютере и анализируются

результаты. И опять может потребоваться уточнение модели... Так будет продолжаться до тех пор, пока анализ результатов не покажет их полное соответствие изучаемому объекту.

Итак, третий этап решения задачи — получение и анализ результатов работы ПК. Таким образом, процесс решения задачи с использованием компьютера можно представить следующей схемой:

✍Контрольные вопросы

1. Что такое математическая модель задачи?
2. Назовите этапы составления математичсекой модели задачи.
3. Что такое программа?

§2 Алгоритм, его свойства

Происхождение термина «алгоритм» связывают с именем великого узбекского математика и астронома Мухаммеда Аль-Хорезми, жившего в IX веке. В своих трудах по арифметике и алгебре Аль-Хорезми разработал правила выполнения четырех арифметических действий над многозначными десятичными числами. Эти правила определяют последовательность действий, которые необходимо выполнить, чтобы получить сумму чисел, произведение и т.д. Почти в таком же виде эти правила изучаются всеми школьниками в начальных классах. Первоначально только эти правила и назывались алгоритмами. В дальнейшем термин «алгоритм» стали использовать вообще для обозначения последовательности действий, приводящей к решению любой проблемы. Умение решать задачи определенного типа (не обязательно математические) всегда означает владение соответствующим алгоритмом.

Алгоритм – понятное и точное предписание исполнителю совершить последовательность действий, приводящую к решению поставленной задачи.

Исполнитель алгоритма – объект или субъект, для управления которым составлен алгоритм.

В качестве исполнителя алгоритма может выступать человек, робот, животное, компьютер и т.д. Использование в качестве исполнителя различных автоматов предъявляет очень строгие требования к точности записи алгоритмов. Это связано с тем, что каждое автоматическое устройство имеет ограниченный, строго определенный набор законченных действий, которые он может исполнять.

Указание выполнить конкретное действие называется командой.

Совокупность команд, которые могут быть выполнены некоторым исполнителем, называется системой команд данного исполнителя.

Пример: *Алгоритм нахождения середины отрезка **АВ** с помощью циркуля и линейки*

1. Поставить ножку циркуля в точку **А**.
2. Установить раствор циркуля, равный длине отрезка **АВ**.
3. Провести окружность.
4. Поставить ножку циркуля в точку **В**.
5. Провести окружность.
6. Через точки пересечения окружностей с помощью линейки провести прямую.
7. Отметить точку пересечения этой прямой с отрезком **АВ**. Эта точка – искомая.

Этот алгоритм состоит из семи команд. Команды алгоритма следует выполнять последовательно одну за другой, в соответствии с указанным порядком их записи. Правильное выполнение всех команд гарантирует решение задачи. Исполнителем этого алгоритма может быть, например, ученик 8 – 9 классов, а первоклассник некоторые команды алгоритма не выполнит, потому что они не входят в систему его команд.

Для того чтобы исполнитель мог однозначно и точно следовать предписаниям алгоритма и эффективно получать определенный результат, алгоритм должен удовлетворять ряду требований, т.е. обладать свойствами:

1. **Дискретность.** *Алгоритм состоит из последовательности законченных действий – шагов. Переход к следующему действию возможен лишь после завершения предыдущего.*
2. **Определенность (точность).** *Каждая команда алгоритма должна однозначно пониматься и точно исполняться.*
3. **Понятность.** *Алгоритм должен содержать только те предписания, которые входят в систему команд данного исполнителя, т.е. алгоритм должен быть понятен исполнителю.*

4. **Массовость.** *С помощью алгоритма можно решать не одну конкретную задачу, а множество однотипных задач и делать это неоднократно.*
5. **Результативность.** *Выполнение алгоритма должно приводить к конкретному результату – решению задачи – за конечное число шагов.*

Определенная последовательность действий исполнителя всегда применяется к некоторым ***исходным данным****. Например: для приготовления блюда по кулинарному рецепту нужны соответствующие продукты (данные). Для решения квадратного уравнения нужны коэффициенты уравнения.*

> **Полный набор данных – необходимый и достаточный набор данных для решения поставленной задачи.**

✍ Контрольные вопросы

1. Объясните происхождение термина «алгоритм».
2. Какими свойствами должен удовлетворять алгоритм?
3. Привести примеры алгоритмов из жизни. Удовлетворяют ли они всем требованиям, прдъявляемым к алгоритмам?
4. Что такое исполнитель алгоритма, команда и система команд?

1. Назовите исполнителей следующих видов работы: уборка мусора во дворе, перевозка пассажиров, выдача заработной платы, прием экзаменов, сдача экзаменов, обучение детей в школе. Сформулируйте систему команд для каждого из этих исполнителей.

2. Определите полный набор данных для решения следующих задач:

а) вычисление стоимости покупок в магазине;

б) вычисление суммы сдачи от данных продавцу денег;

в) определение месячной платы за электроэнергию.

3. Сформулируйте алгоритмы для заданий из п.2.

4. Упорядочьте следующий набор команд так, чтобы получился алгоритм заварки чая:

а) дать настояться;

б) довести воду до кипения;

в) добавить по вкусу сахар;

г) залить кипятком;

д) налить в стаканы;

е) насыпать заварку в чайник.

5. Напишите алгоритм перехода улицы без светофора.

6. Продолжите алгоритм работы «черного ящика»:

а)

вход	1	2	7	4	8	12	20	21	32	46
выход	Не могу	0	Не могу	1	3	5	9			

б)

вход	3	9	21	33	123	345	1234			
выход	3	9	3	6	6					

в)

вход	Квас	Змея	Мама	Коля	Три	Пять	Байт			
выход	Лгбт	Инеа	Нбнб	Лпма	Усй					

7. Выполните алгоритм, определяющий день недели, приходящийся на 1-е января для любого года XX века. Для этого обозначьте две последние цифры в записи года – N, а соответствующий день недели – D.

1. *Найти целую часть числа, полученного умножением N на 1,25. Результат обозначить X.*
2. *Найти остаток от деления X на 7 и обозначить через Y.*
3. *К единице прибавить Y. Результат считать значением D.*

8. Найдите НОД двух натуральных чисел, если X=26, Y=65, используя при этом алгоритм Евклида:

п.1 Если X >Y, то перейти на п.4, иначе перейти на п.2.

п.2 Если Y >X, то перейти на п.5, иначе перейти на п.3.

п.3 Считать НОД=X. Конец.

п.4 Вычислить X – Y, далее считать эту разность новым значением X, перейти на п.1.

п.5 Вычислить Y - X, далее считать эту разность новым значением Y, перейти на п.1

9. Напишите алгоритм решения следующей задачи: *Имеются три сосуда - A, B и C. Объем первого – 8 литров, второго – 5, третьего – 3. Первый заполнен до краев. Остальные пусты. За наименьшее количество переливаний добиться того, чтобы в первом сосуде остался 1 литр. Переливать в сосуд можно ровно столько, сколько в него помещается.*

§3 Способы записи алгоритмов. Основные алгоритмические конструкции.

Предпи- сания

Форма записи произвольная. Чаще используется для записи «бытовых» алгоритмов.
Пример: поиск слова в словаре.

Формулы

Задается формула и значения входящих в неё величин, требуется произвести вычисления и получить числовой результат.
Пример: Найти площадь треугольника со сторонами ***a, b, c*** по формуле Герона

$$S = \sqrt{p(p-a)(p-b)(p-c)}$$

Таблицы

Используется для организации вычислений по формуле с поочередной регистрацией промежуточных результатов.

Пример:

a	*b*	*c*	*p*	*S*
5	6	8	9,5	224,4375
2,4	3,1	3,7	4,6	4,554

Алгорит- мический язык

При записи алгоритма используется ограниченное число служебных слов, смысл которых строго определен (**алг, нач, кон, арг, рез**).

```
Пример:  алг  минимальное из двух
         арг  a, b
         рез  y
        нач                 и т. д.
```

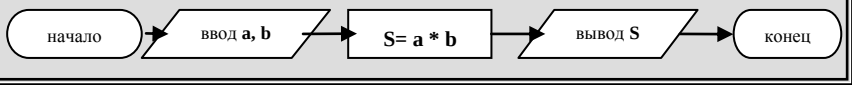
Язык програм- мирования

Алгоритм, записанный на языке программирования, называется **программой**. Служит для окончательной записи алгоритма для исполнения его ПК.

```
Пример:  10 INPUT  A, B
         20 IF  A<B  THEN Y=A  ELSE  Y=B
         30 PRINT  Y
```

Язык блок-схем

Каждому действию соответствует определенная геометрическая фигура (блок), последовательность действий указывается стрелками, соединяющими эти блоки. Является вспомогательным способом записи алгоритмов, облегчающим процесс написания программ.
Пример:



Основные алгоритмические конструкции

Линейный

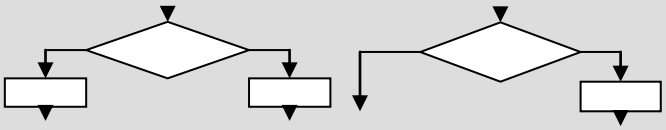
Алгоритм, у которого последовательность операций при исполнении совпадает с порядком их следования в записи алгоритма и не зависит от входных данных.

Блок-схема имеет вид:

Разветвляющийся

Алгоритм, в котором заложена операция выбора действия в зависимости от исходных данных.

Блок-схема имеет вид:

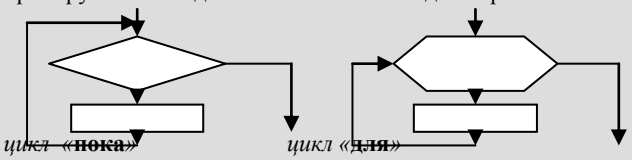


полная форма *сокращенная форма*

Циклический

Алгоритм, в котором некоторая группа команд выполняется более одного раза.

Блок-схема имеет вид:



*цикл «**пока**»* *цикл «**для**»*

Контрольные вопросы

1. Перечислите способы записи алгоритмов?
2. Какой способ записи алгоритма направлен на решение квадратного уравнения ученика у доски?
3. Какой способ записи алгоритма направлен на исполнитель компьютер?
4. Какой способ записи алгоритмов чаще всего встречается в быту? В кулинарной книге? В математике? Объясните ответ.
5. Какие виды алгоритмов вы знаете?
6. Какие блоки присутствуют при записи любого алгоритма?
7. В чем отличие цикла «пока» и цикла «для».

1. Запишите алгоритм вычисления площади треугольника по формуле Герона различными способами.

2. Запишите алгоритм вычисления дискриминанта с помощью таблицы.

3. Запишите алгоритм проверки условия: могут ли три данных числа быть сторонами треугольника.

4. Составьте для данных алгоритмов блок-схему. Какие значения примут переменные ***a*** и ***b*** после выполнения этих алгоритмов?

а)
Присвоить *a* значение 13.
Присвоить *b* значение 12.
Если *a* < *b* + 1, то:
Присвоить *b* значение *b* — *a.*
Присвоить *a* значение *a* — *b.*
Иначе:
Присвоить *a* значение a — *b.*
Присвоить *b* значение *b* — *a.*
Конец ветвления.
Присвоить *a* значение *ab.*
Если a>b^2, то:
Присвоить *b* значение *a*—*b.*
Конец ветвления.

б)
Присвоить *a* значение —3.
Присвоить *b* значение —1.
Пока *a*<*b,* повторять:
Если *b*<3, то:
Присвоить *a* значение *a*— *b.*
Присвоить *b* значение *b*+*2.*
Иначе:
Присвоить *a* значение *a*+*2.*
Присвоить *b* значение *b* — *a.*
Конец ветвления.
Конец цикла.

5. Составить блок-схему для решения следующих задач:

a) Вычислить площадь квадрата.
b) Вычислить периметр пятиугольника.
c) Найти максимальное среди четырех целых чисел.
d) Найти сумму первых N четных чисел.
e) Имеются три урны: белая, черная и полосатая. В полосатой урне находятся белые и черные шары. Надо все белые шары положить в белую урну, а черные – в черную.
f) Найти среднее арифметическое пяти натуральных чисел, введенных произвольным образом.

§4. Языки программирования

Еще в эпоху машин первого поколения возникла настоятельная потребность предоставить широкому кругу непрофессиональных пользователей удобные и эффективные средства общения с компьютерами. Так начали появляться машинно-независимые алгоритмические языки высокого уровня. Перед их создателями стояли следующие задачи: приблизить способ записи и систему выражений к обычной «человеческой» речи; ввести в языки общепринятую математическую символику и другие легко понимаемые изобразительные средства; обеспечить возможность описания широкого круга алгоритмов; позаботиться об удобстве автоматического перевода программ, написанных на языке, в программы для ЭВМ.

Одним из первых языков, удовлетворяющих этим требованиям, явился язык Фортран (FORmula TRANstation). Первое сообщение о языке относится к 1954 году, а опубликованная в 1958 г. версия языка получила название Фортран-II. В 1962 г. появился Фортран-IV, получивший во всем мире очень большое распространение.

Вскоре после создания Фортрана появился алгоритмический язык Алгол (ALGOrithmic Language). Описание языка было опубликовано в 1960г. Эта дата вошла в его название — Алгол-60.

К этому же времени относится появление языка для программирования экономических задач Кобол (COmmon Business Oriented Language).

Появление машин третьего поколения поставило вопрос о создании универсальных алгоритмических языков, пригодных для представления алгоритмов решения самых разнообразных задач. Одной из попыток такого рода являлось создание в середине 60-х годов алгоритмического языка ПЛ-1 (Programming Language/1). Другая попытка была связана с дальнейшим развитием языка Алгол. Группа математиков—членов IFIP (международная федерация по обработке информации) опубликовала в 1968 г. изложение основ универсального алгоритмического языка, получившего название Алгол-68.

По мере распространения идей программирования и расширения практики машинного решения разнообразных задач начали появляться языки, специально предназначенные для пользователей, находящихся на начальной стадии овладения компьютерной грамотностью. Одним из таких языков явился созданный в 1965г. язык Бейсик (Beginner's All-purpose Symbolic Instruction Code). Он был задуман как простое и удобное для начинающих средство решения вычислительных задач в непосредственном диалоге с компьютером.

В 1970 г. Н.Вирт предложил новый язык, ориентированный на задачи обучения программированию и названный в честь известного математика Блеза Паскаля языком Паскаль. Разработка новых алгоритмических языков продолжается и в настоящее время.

В основе каждого алгоритмического языка лежит некоторый набор основных символов (алфавит этого языка).Почти одновременно с ПАСКАЛем, в начале 70-х гг., был разработан и язык программирования СИ. Но если ПАСКАЛЬ шел больше от теории программирования, то язык СИ — типичный пример влияния практических потребностей системного программирования на разработку новых языков. Изначально он создавался как инструментальное средство для реализации операционной системы UNIX на ЭВМ фирмы DEC, но популярность его быстро переросла рамки конкретной машины, операционной системы и задач системного программирования. И сейчас язык СИ можно по праву назвать одним из универсальных языков программирования.

§ 5 Основные понятия языка Паскаль.

Первая версия языка программирования Паскаль (назван в честь французского ученого Блеза Паскаля (1623-1662) была разработана на кафедре информатики Стэндфордского университета швейцарским ученым Никлаусом Виртом в 1968 году как учебный язык программирования. Признание программистов и простых пользователей этот язык получил после появления его диалекта Турбо Паскаль, разработанного фирмой Borland International.

Алфавит языка Паскаль:

1. *латинские буквы от A до Z (прописные и строчные буквы не различаются.*
2. *русские буквы от А до Я (только в некоторых случаях);*
3. *арабские цифры 0…9;*
4. *знаки арифметических операций (+,-, /, *), знаки пунктуации*
5. *символы подчеркивания « _ » и пробела « ».*

Имя переменной (идентификатора) должно содержать только буквы латинского алфавита, цифры и символы подчеркивания и начинаться только с буквы.

Основные типы переменных

№ п/п	*тип*	*название*	*диапазон значений*
1.	***Integer***	целый	-32768 …32767
2.	***Real***	вещественный (действительный)	2.9E-39…1.7E38
3.	***Boolean***	логический	False (ложь), True (истина)
4.	***String***	строковый	Любые символы- буквы, цифры, знаки препинания

Основные математические выражения на языке Паскаль

функция	*запись на Паскале*	*пример*
умножение	*	2*3=6
деление	/	10/2=5
квадрат	Sqr (x)	Sqr(5)=25
модуль	abc(x)	Abc(-90)=90
Квадратный корень	Sqr(x)	Sqrt(25)=5
синус	Sin(x)	Sin(pi)=0
косинус	Cos(x)	Cos (pi)= -1
Целая часть от деления	div	17 div4=4
Остаток от деления	mod	17mod4=1
Генератор случайных чисел	Random	Случайное число из диапазона (0-1)
	Random(x)	Случайное число из диапазона (0-x)
Экспонента	Exp(x)	Exp(2)
Натуральный логарифм	Ln(x)	Ln(e)=1

Компилятор языка Паскаль работает с числами с плавающей запятой. Например, 0, 094=9.4E-0002

✍ **Контрольные вопросы**

1. Кто создал язык программирования Паскаль?
2. Что такое идентификатор?
3. Могут ли в Паскале использоваться римские цифры?
4. Какие типы данных имеются в Паскале?
5. Можно ли к переменным целого типа применять операцию деления?

Упражнения

1. Заполните таблицу.

С фиксиров. запятой	*С плавающей запятой*
0.00035716	
101325	
	1.0737E+09
	6.1803E-01

2. Из перечисленных ниже последовательностей символов выбрать идентификаторы:

*X; x_1; sin x; 2a; объем; delta; max 15; a-1; x*3.*

3. Объявите переменные, которые необходимы для:

а) вычисления значения функции y= x^2;

б) вычисления стоимости покупки, состоящей из нескольких тетрадей, карандашей и линейки.

4. Запишите на языке Паскаль следующие математические выражения:

1) $\frac{x+y}{x+1}-\frac{xy-12}{34+x}$;

2) $\sqrt{1+\sqrt{|x-2{,}5|}}$;

3) $\left|x^2-x^3\right|-\frac{7x}{x^4-15x}$

4) $\frac{a}{c}\cdot\frac{b}{d}-\frac{ab-c}{cd}$

5) $\frac{b+\sqrt{b^2+4ac}}{2a}-a^3c+3{,}5$

6) $2ctg(3x)-\frac{\ln\cos x}{\ln(1+x^2)}$

7) $\frac{xyz-3{,}3\left|x-\sqrt[4]{y}\right|}{10^7+\ln\sin^2 x}$

8) $btg^2x-\frac{a}{\sin^2\frac{x}{2}}+\frac{ae^x}{\cos(\frac{bx}{a})}$

5. Переписать следующие математические выражения, записанные по правилам Паскаля, в традиционной математической форме:

a) sqrt (a+b)- sqrt (a-b);

b) a+b/(c+d)-(a+b)/c+d;

c) 1+ sqr(cos((x+y)/2));

д) sin (sin(sqr(x)-1)+cos(x*x*x-1)*cos(abc(x-2)-1)/y*a+sqrt(abc(y)-x).

§6 Программирование линейных алгоритмов

Программа – это последовательность команд, предназначенных для реализации алгоритма на компьютере, написанная на языке программирования.

Программу составляют по такому принципу: вводят исходные данные, вычисляют и выводят результаты. Программа на языке Паскаль состоит из двух частей: описательной части и тела программы.

```
Program      { Заголовок программы}
Const        { Подраздел описания констант}
Type         { Подраздел описания типов}
Var          { Подраздел описания переменных}
Procedure    { Подраздел описания процедур}
Function     { Подраздел описания функций}
Begin        { Тело программы}
end.
```

Программа начинается со служебного слова ***Program,*** *после которого идет имя программы, которое присваивает сам программист. Имя программы должно начинаться только с буквы и должно содержать только буквы, цифры и символы подчеркивания. Заголовок программы необязателен.*

Основным разделом описательной части является раздел описаний переменных ***Var****. В этом разделе описываются все переменные которые используются в программе, а также их типы.*

Тело программы начинается со служебного слов ***Begin*** *и заканчивается служебным словом* ***End****, после которого обязательно ставится точка. Внутри* ***Begin… End*** *содержатся операторы, которые осуществляют выполнение программы. Операторы в основном блоке программы разделяются точкой с запятой (;). После* ***Begin*** *и перед* ***end*** *точка с запятой не ставится.*

В программу могут также входить ***комментарии****.* ***Комментарий*** *- фрагмент текста программы, заключенный в скобки вида {} или (* *), который служит для объяснения работы программы и не влияет на работу всей программы.*

Оператор присваивания

<имя переменной>:=<выражение>

Переменная и выражение должны быть одного типа или согласованными.

Например, p:=(a+b+c)/2.

Операторы ввода данных

Read *(список имен),* ***Readln*** *(список имен)*

Различие между данными командами заключается в том, что оператор Readln переводит курсор после ввода значений на новую строку.

Пустой оператор ***Readln*** *используется в конце программы для задержки изображения, так как после вывода результатов выполнения программ на экран, система очень быстро возвращается в окно программы и пользователь не успевает увидеть результаты.*

Операторы вывода данных

Write (список имен), **Writeln** (список имен)

Добавление –ln переводит курсор на новую строку.

Пример. Составить программу вычисления площади треугольника по формуле Герона $S = \sqrt{p(p-a)(p-b)(p-c)}, p = \frac{a+b+c}{2}$

```
Program Geron;
Var a, b, c, p, S: Real;
Begin
  writeln (‘Введите стороны треугольника’);
  readln (a,b,c);
  p:=(a+b+c)/2;
  s:= sqrt (p*(p-a)*(p-b)*(p-c));
  writeln (s);
  readln;
end.
```

✍ Контрольные вопросы

1. Перечислите основные части Паскаль-программы.
2. Для чего используется раздел Var?
3. Как перевести курсор на новую строку после ввода данных?
4. Для чего используется оператор Readln?
5. Какой оператор служит для вывода данных?
6. С чего начинается тело программы?
7. Чем заканчивается любая Паскаль-программа?
8. Объявите переменные и напишите фрагмент программы вычисления объема цилиндра, обеспечивающий вввод исходных данных.

1. Запишите в виде инструкции присваивания формулу вычисления площади круга: $s= \pi r^2$.

2. Вычислить периметр и площадь прямоугольного треугольника по заданным длинам двух катетов.

3. Написать программу, которая выводит на экран дату начала второй мировой войны.

4. Написать программу, которая выводит на экран четверостишие:

Буря мглою небо кроет,
Вихри снежные крутя,
То как зверь она завоет,
То заплачет как дитя.

А.С. Пушкин.

5. Написать программу, которая вычисляет длину окружности и площадь круга радиуса 5 см.

6. Написать программу вычисления объема цилиндра.

7. Написать программу вычисления стоимости покупки, состоящей из нескольких тетрадей и такого же количества обложек к ним.

8. Составить программу вычисления значения функции:

$$\left|\cos\frac{x}{2,7}\right| + 9,1\sin(1,2x+1)$$

9. Найти сумму членов арифметической прогрессии, если известны ее первый член, знаменатель и число членов прогрессии.

10. Написать программу пересчета веса из фунтов в килограммы (1 российский фунт равен 409, 5 г).

11. Написать программу, вычисляющую подоходный налог от заработной платы (13%).

12. Написать программу для вычисления пути, пройденного лодкой, если ее скорость в стоячей воде Vкм/ч, скорость течения реки U км/ч, время движения по озеру t_1, а против течения – t_2 ч.

13. Написать программу вычисления стоимости поездки на автомобиле на дачу (туда и обратно). Исходными данными являются: расстояние до дачи (в километрах); количество бензина, которое потребляет автомобиль на 100 км пробега; цена одного литра бензина.

14. Текущее показание электронных часов: m часов ($0 \le m \ge 23$), n минут ($0 \le n \le 59$) , k секунд ($0 \le k \le 59$). Какое время будут показывать часы через p часов и q мин r сек.

§7. Описание среды программирования Turbo Pascal.

Для языка Паскаль наиболее распространенными являются две среды программирования: Turbo Pascal 7.0 Borland Pascal для MS-DOS и Windows. Они предназначены для подготовки текстов программ и их выполнения.

Для входа в среду нужно выполнить команду turbo. exe из папки TPas., В верхней строке экрана размещено главное меню, а в нижней – описание некоторых функциональных клавиш.

В распоряжение пользователя будут такие пункты меню:

File - для работы с файлами:

Edit – для редактирования файла;

Search – для поиска или замены заданного фрагмента текста;

Run – для выполнения программы;

Compile – для компиляции программы и создания exe- файла;

Debug – для настройки программы;

Options – для конфигурирования среды;

Window – для конфигурации окон и работы с ними;

Help – для вызова помощи.

Выход осуществляется комбинацией клавиш Alt+X или командой Exit из меню File.

Создание новой программы

Для того, чтобы открыть окно для написание новой программы, в меню File выбрать команду New. По умолчанию окно программы будет иметь название NONAME00. PAS (или вместо 00 другой номер). На экране появится окно, в котором можно набирать текст программы.

Запуск программы

Для того, чтобы запустить программу, требуется выйти в меню Run и дважды нажать Enter.

Сохранение программы

Для того, чтобы сохранить программу, необходимо в режиме File выбрать Safe as…, нажать клавишу Enter. Появится окно, в котором необходимо указать имя файла и, если необходимо, путь.

Отмена последнего действия

Отменить последнее действие можно выполнив команду Undo (Отменить) из меню, или с помощью клавиш Alt+ Backspase.

Выполнив эту команду несколько раз подряд, можно отменить предыдущую последовательность действий (в обратном порядке).

Отменить последнее использование команды «Отмена» можно командой Redo (Повторить) из меню Edit.

Копирование, вырезка и вставка

Для того, чтобы скопировать часть текста, необходимо:

1. *Выделить фрагмент текста клавишами управления курсором при нажатой клавиши Shift.*
2. *Выполнить команду Copy из меню: выделенный фрагмент скопируется в буфер обмена.*
3. *Установить курсор в то место, куда необходимо скопировать текст и выполнить команду Past из меню Edit.*

Для того, чтобы вырезать часть текста, необходимо:

1. *Выделить требуемый фрагмент.*
2. *Нажать сочетание Shift+ Del: фрагмент удаляется с экрана в буфер обмена.*

Для того, чтобы вставить из буфера, необходимо выполнить команду Past из меню Edit.

Типичные ошибки

Обнаружив грамматическую ошибку, Паскаль выдает золотыми буквами на красном фоне краткое описание ошибки и ставит курсор в то место программы, где, по его мнению, она находится.

Приведем примеры наиболее часть встречающихся ошибок:

Unknown identifer (Неизвестный идентификатор). Этот идентификатор не был описан.

Duplicate identifier (Повторение идентификатора).

Syntax error (Синтаксическая ошибка).

Invalid file name (Недопустимое имя файла). Имя файла неправильное или указан несуществующий путь.

Type mismatch (Несоответствие типов).

Error in expression (Ошибка в выражении).

Division by zero (Деление на нуль).

Unexpected end of file (Неожиданный конец файла). Не поставлена точка в конце программы или не совпадает количество begin с количеством end.

«; » expected (Ожидается символ « ; »).

«:=» expected (Ожидается символ «:=»).

« . » expected (Ожидается символ « . »).

✍ Контрольные вопросы

1. Как запускается система программирования Турбо Паскаль?
2. Что такое компиляция программы?
3. Как осуществляется сохранение программы?
4. Как выйти из системы программирования Турбо Паскаль?
5. В чем различие опций Save (сохранить файл), Save as… (сохранить файл под именем…) и Save all (сохранить все измененные файлы)?

§8 Программирование ветвящихся алгоритмов.

Для программирования ветвящихся алгоритмов применяются условный оператор (оператор ветвления) и оператор выбора.

1. Условный оператор.

Условный оператор имеет следующий формат:

Полная форма:

If <логическое выражение> ***then*** <оператор 1> ***else*** <оператор 2>

Неполная форма:

If <логическое выражение> ***then*** <оператор >

Логическое выражение – это способ записи на языке программирования условий поиска необходимых данных. Логическое выражение может принимать значения true (истина) или false (ложь). Логические выражения могут быть простыми(a>b, a<=-10 и т.д.) и сложными (когда простые логические выражения соединены логическими операциями not(отрицание) and (и) or (или)).

Оператор 1 *— оператор, который выполняется, если условие принимает значение True,* ***оператор 2*** *— оператор, который выполняется если* ***условие*** *принимает значение False. В полной форме оператора* ***If*** *обязательно будет выполнен один из двух операторов: оператор1 или оператор2, а затем управление будет передано следующему оператору, стоящему в тексте программы за оператором* ***If.*** *В неполной форме — либо будет выполнен оператор, либо управление будет передано следующему оператору, стоящему в тексте программы за оператором* ***If.***

Пример 1. **If** a>0 **then** b:=1 **else** b:=2 – полная форма, в которой при положительном значении переменной `A`, переменной `B` присваивается значение 1, а при отрицательном или нулевом значении переменной A, переменная `B` получает значение 2.

Пример 2. Пусть a=5. Тогда в результате выполнения команд:

```
If  a<7 then
   begin
     b:= a-2;
     c:=1+2*a
end;
else
begin
b:=2+5 *a;
c:=12- 4*(a-3)
end;
```

получим b=3, c=11.

2. Оператор выбора.

Оператор выбора позволяет программировать по многим направлениям в зависимости от значения заданного выражения

Формат оператора выбора:

```
Case K of
  A1: <оператор 1>;
  A2: <оператор 2>;
  …
  AN: <оператор N>
else  <оператор N+1>
end;
```

K – простая переменная целого, символьного типов;

Выполнение оператора начинается с вычисления выражения K, полученное значение сравнивается с A1, A2, …, AN и выполняется соответствующий оператор. Если ни одно из значений не совпало с K, то выполняется оператор после слова else.

Возможно использование неполного оператора выбора без ветви else.

Пример 3. Составить программу, которая, на вводимый с клавиатуры год, выдавала бы — к какому этапу развития средств работы с информацией он относится.

Решение. Временные периоды соответствующие каждому этапу оформим в виде диапазонов констант типа Integer. В зависимости от значения переменной-переключателя year типа Integer будет напечатано соответствующее окончание фразы “Этот год относится к этапу…”.

```
Program Definer_of_Year;
Var Year: integer;
Begin
   Writeln (‘Введите год’);
   Readln (year);
   Writeln (‘Этот год относится к этапу  ’);
   Case Year of
     400..700:    Writeln (“письменной  информации’’);
     701..1275:  Writeln (“напечатанной” информации’’);
     1276..1932: Writeln (‘‘механических средств работы  с информацией’’);
     1933..1967: Writeln (‘первых вычислительных машин’’);
     1967..1984: Writeln (‘‘микропроцессорной техники’’);
     1985..2005: Writeln (‘‘новых информационных технологий’’)
          else   Writeln (’’нет исторических данных’’)
  end;
end.
```

✍ Контрольные вопросы

1. Объясните принцип работы оператора условия.
2. В чем отличие между полной и неполной формой условного оператора?
3. Отсутствие какого символа обязательно перед служебным словом `else` в условном операторе?
4. Для чего предназначен оператор выбора?
5. Является ли условным оператором последовательность символов:

   а) `If` X<Y `then` X:=0 `else` Y:=0;

   б) `If` X>Y `then` X:=0 `else` `Read`(Y);

6. Какое значение будет иметь переменная C после выполнения операторов

```
C:=0; If A > 0 then If B>0 then C:=1 else C:=2;
```

при следующих значениях переменных A и B:

а) A = B = 1;

б) A = 1, B = -1.

Упражнения

1. Составить программу решения полного квадратного уравнения.

2. Составить программу проверки знания даты основания города Тамбова.

3. Составить программу, которая проверяет, делится ли на три целое число, введенное с клавиатуры.

4. Составить программу, вычисляющую значение функции:

$$y=\begin{cases} x^2-3x+9, \text{ если } x\le 3 \\ \dfrac{1}{x^3+6}, если x>3 \end{cases}$$

5. Составить программу нахождения суммы большего и меньшего из трех чисел.

6. Подсчитать количество отрицательных чисел среди чисел a, b,c

7. Составить программу, позволяющую получить словесное описание школьных отметок (1- плохо, 2 – неудовлетворительно, 3- удовлетворительно, 4 – хорошо, 5 - отлично).

8. Составить программу, которая запрашивает у пользователя номер месяца и выводит соответствующее название времени года. В случае, если пользователь укажет недопустимое число, программа должна вывести сообщение «Ошибка ввода данных».

9. В небоскребе N этажей и всего один подъезд; на каждом этаже по 3 квартиры; лифт может останавливаться только на нечетных этажах. Человек садится в лифт и набирает номер нужной ему квартиры M. На какой этаж должен доставить лифт пассажира?

10.Даны числа x,y,z. Найти значение выражения:

$$u=\frac{\max^2(x,y,z)-2^x\cdot\min(x,y,z)}{\sin 2x+\max(x,y,z)/\min(x,y,zz}$$

§9 Программирование циклических алгоритмов

Цикл *– это выполнение определенного набора команд некоторое количество раз.*

В Паскале существует три вида циклов:

1) цикл с предварительным условием (предусловием);

2) цикла с последующим условием (постусловием);

3) цикл цикласо счетчиком (с параметром).

1. Цикл с предусловием. *Цикл с предварительным условием используется в тех случаях, когда заранее неизвестно число повторений цикла, даже неизвестно нужно ли выполнять его хотя бы один раз.*

Форма записи оператора цикла с предусловием:

While условие **do** оператор

Нельзя после служебного слова do ставить точку с запятой. В этом случае компилятор ошибки не выдаст, но программа "зависнет".

Работа этого цикла заключается в следующем. Предварительно проверяется значение условия. Пока оно принимает значение True, выполняется тело цикла. Как только условие становится False, происходит выход из цикла и передача управления следующему за ним оператору. Если с самого начала условие False, то тело цикла не выполнятся ни разу.

Если требуется, чтобы тело цикла While состояло не из одного оператора, а из списка операторов, то их объединяют в один составной оператор, заключая в операторные скобки Begin [список операторов] end.

Пример1.

```
A:= -50; C:=4; {задание начальных значений переменным A и C}
While A<=0 do  {пока значение переменной A не положительно}
Begin          {выполняется составной оператор тела цикла}
   C:=C*2;{изменение значений переменных A и C}
    A:=A+10
end; {конец цикла While}
```

2. Цикл с постусловием. *Цикл с последующим условием используется в тех случаях, когда заранее не известно число повторений цикла, но требуется, чтобы хотя бы один раз тело цикла было выполнено. Форма записи оператора цикла с постусловием:*

Repeat [список операторов] **until** условие

где ***список операторов*** *— тело цикла, операторы в списке операторов разделяются точкой с запятой (;).*

Работает цикл следующим образом. Список операторов тела цикла выполняется (по крайней мере, один раз) до тех пор, пока условие False. То есть сначала выполняются операторы тела цикла, а затем проверяется условие. В этом отличие работы циклов While…do и Repeat…Until. Цикл прекращает свою работу и передает управление следующему после него оператору, когда условие True.

Пример2.

```
A:= -50;  C:=4; {Задание начальных значений переменным  A и C.}
      Repeat {Начало цикла.}
  C:=C*2; A:=A+10        {Изменение значений переменных A и C}
until A>0; {до тех пор, пока значение переменной A не станет >0}
```

3. Цикл со счетчиком (с параметром). Этот цикл используется в том случае, когда заранее известно число повторений.

Существует два варианта оператора цикла со счетчиком:

1) ***For i:= K to N do*** *оператор;*
2) ***For i:= N downto K do*** *оператор,*

— где i — это счетчик (параметр) цикла; K- начальное значение, N- конечное значение счетчика. Оператор — тело цикла, которое выполняется повторно для каждого значения счетчика цикла от его начального значения до его конечного значения включительно. Первая конструкция применяется, когда начальное значение счетчика цикла меньше его конечного значения, и после каждого выполнения тела цикла значение счетчика цикла автоматически увеличивается на 1. Вторая конструкция применяется, когда начальное

значение счетчика цикла больше его конечного значения, и после каждого выполнения тела цикла значение счетчика цикла автоматически уменьшается на 1.

В качестве переменной цикла должна выступать порядковая (перечисляемая) переменная. Использование переменной типа Real не допускается, в этом случае компилятор выдаст следующее сообщение об ошибке: "Error #97: Invalid For control variable" (т.е. недопустимая переменная цикла).

Пример 3. For I:=1 to 10 do A:=A+1;

Если требуется, чтобы тело цикла состояло не из одного оператора, а из списка операторов, то их объединяют в один составной оператор, заключая в операторные скобки Begin [список операторов] end.

Пример 4. Составить программу, которая вычисляет сумму натурального ряда чисел от 1 до N.

```
Program summa;
Var i,  n, S: integer;
Begin
   Readln (n);
   S:=0;
   For i:=1 to n do
   S:=S+i;
   Write (S);
   Readln;
End.
```

✍ Контрольные вопросы

1. Что такое цикл?
2. Какие циклы вы знаете? В каких случаях применяется каждый из циклов?
3. В чем различие цикла с предусловием и цикла с постусловием?
4. Сколько раз выполнится оператор цикла repeat, если условие после слова until истинно?
5. Переменной какого типа может быть счетчик цикла?

Упражнения

1. Составить программу, которая 10 раз выводит на экран слово «футбол».

2. Составить программу, которая выводит на экран таблицу квадратов первых десяти целых положительных чисел.

3. Составить программу, которая вычисляет факториал числа, введенного с клавиатуры. (Факториалом числа n называется произведение целых чисел от 1 до n.

4. Составить программу, вычисляющую сумму и среднее арифметическое последовательности положительных чисел, которые вводятся с клавиатуры.

5. Составить программу, которая выводит таблицу умножения, например, на 7.

6. Составить программу, которая вычисляет значение функции f(x)= x- sin x на отрезке [a,b] с шагом h. Результат представить в виде таблицы, первый столбец которой – значения аргумента, второй – соответствующие значения функций.

7. Составить программу, которая вычисляет наибольший общий делитель двух целых чисел.

8. Дано натуральное N. Вычислить:

$$S = 1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{8} + \ldots + (-1)^n \cdot \frac{1}{2^n}$$

9. Дано действительное a. Найти среди чисел 1, $1+\frac{1}{2}, 1+\frac{1}{2}+\frac{1}{3}, \ldots$ первое, большее a.

10. Написать программу, которая «задумывает» число в диапазоне от 1 до 10 и предлагает пользователю угадать число за 5 попыток.

§10 Массивы

Массив - это упорядоченный набор однотипных значений. Массиву присваивается имя, посредством которого можно ссылаться на него, как на единое целое, та и на любую из его компонент.

Описание массива определяет имя, размер массива и базовый тип (тип его элементов).

Формат описания массива:

<Имя массива>=**Array**[<диапозон>] **of** <базовый тип >;

Массивы бывают одномерные (векторы), двумерные (таблицы, матрицы), трехмерные и т.д. Как правило, массивы размерности больше 3 встречаются на практике редко..

Пример1. Vector = Array[1..20] of real; {Описание одномерного массива из 20 действительных элементов}

Пример2. Matrix = Array[1..10 , 1..10] of integer; {Описание двумерного массива (матрицы) из 100 целых элементов}

Способы задания элементов массива:

1) *ввод элементов с клавиатуры;*
2) *заполнение случайными числами, при помощи функции Random.*

Ввод и вывод массива производится поэлементно. Обычно для этого используется цикл с параметром, где в качестве параметра применяется индексная переменная.

Работа с массивами чисел:

1. *Арифметические операции с элементами массива.*
2. *Поиск в массиве.*
3. *Подсчет элементов, удовлетворяющих заданному критерию.*
4. *Вставка и удаление элементов массива.*
5. *Разбиение массива на несколько массивов.*
6. *Сортировка массива.*

Пример 3. Сформировать таблицу Пифагора и вывести ее на экран. Таблица будет представлять собой двумерный массив, элементы строк которого пронумеруем через I, а элементы столбцов через j. Значения элементов вычисляются следующим образом P[i,j]=i*j.

```
Program Pifagor;
Var P: array[1..9, 1..9] of integer; i, j:integer;
begin
    for i:=1 to 9 do
    for j:=1 to 9 do
        P[i,j]:= i*j;
for i:=1 to 9 do
    begin   for j:=1 to 9 do
            write (P[i,j])4
            writeln;
    end;
end.
```

✍ Контрольные вопросы

1. Что такое массив?
2. Можно ли назвать массивом список учеников в классном журнале?
3. Какие типы массивов вы знаете?
4. Как можно создать массив?
5. Какие операции можно выполнить над массивом?

1. Дан массив натуральных чисел. Составить программу, которая определяет сумму элементов, кратных данному К.

2. Даны действительные числа a_1, a_2, $a_{3, \ldots,}$ $a_{n.}$ Составить программу, меняющую местами наибольший и наименьший элементы.

3. Составить программу, которая определяет, сколько раз встречается в массиве введенное с клавиатуры число.

4. Составить программу, которая определяет количество учеников в классе, чей рост превышает средний.

5. Задан массив, содержащий несколько ненулевых элементов. Составить программу, которая образовывала новый массив из данного, выбросив нулевые элементы.

6. Составить программу, которая вводит по строкам с клавиатуры двумерный массив и вычисляет среднее арифметическое его элементов.

7. Составить программу, определяющую номера строк двумерного массива, хотя бы один элемент которых равен с.

8. Составить программу, которая выводит на экран двумерный массив вида:

1	2	3	…	n
n	n-1	n-2	…	1
1	2	3	…	n
n	n-1	n-2	…	1
…	…..	….		
n	n-1	n-2	…	1

§11 Строковые переменные в языке Паскаль

Строка – *упорядоченная последовательность символов.*

Количество символов в строке называется ее длиной. Длина строки может лежать в диапазоне от 0 до 255.

Значения переменных этого типа заключаются в апострофы.

Например, A:= ‘информатика’.

Два следующих друг за другом апострофа ‘’ обозначают пустую строку.

Элементы строки идентифицируются именем строки с индексом, заключенным в квадратные скобки.

Например, A[4]=’o’, A [1]=’u’.

Операции над строками

1. *Операция сцепления (конкатенации)* ***Concat*** $(S_1, S_2,..,S_n)$ *применяется для соединения нескольких строк в одну результирующую.*

Значение S	*Результат*
Concat(‘мама’, ‘ мыла’, ’раму’)	‘мама мыла раму’

2. *Операция* ***Copy (S, Poz, N)*** *выделяет из строки S подстроку длиной N символов, начиная с позиции Poz.*

Значение S	*Выражение*	*Результат*
‘мама мыла раму’	*Copy (S,6,4)*	‘ мыла ’

3. *Операция* ***Length (S)*** *определяет текущую длину строки S. Результат – значение целого типа.*

Значение S	*Выражение*	*Результат*
‘мама мыла раму’	Length (S)	*15*

4. *Функция* ***Pos*** (S_1, S_2) *обнаруживает первое появление в строке* S_2 *подстроки* S_1. *Результат – целое число, равное номеру позиции, где находится первый символ подстроки* S_1.

Значение S	*Выражение*	*Результат*
‘abcdef’	*Pos (‘cd’,S)*	*3*

5. Функция ***Delete (S, Poz, N)*** *удаляет N символов из строки S, начиная с позиции Poz.*

Значение S	*Выражение*	*Результат*
‘abcdef’	*Delete (S,3,2)*	‘abcd’

6. Функция ***Insert(S_1, S_2, Poz)*** *вставляет строку S_1 в строку S_2, начиная с позиции Poz.*

Значение S	*Выражение*	*Результат*
‘abcdef’	*Insert (‘WR’,S,5)*	‘abcdWRef’

✍ Контрольные вопросы

1. Что такое строковая переменная?
2. Какие операции можно выполнять со строками?
3. Каким образом можно произвести сравнение строк?
4. Что произойдет в результате сложеня переменных a= ‘123’ и b=’459’?

1. Дана строка, заканчивающая точкой. Подсчитать, сколько в ней слов.

2. Дана строка. Подсчитать в ней количество вхождений букв r, k, t.

3. Дана строка. Определить, сколько раз входит в нее группа букв abc.

4. В строке удалить все двоеточия (:) и подсчитать количество удаленных символов.

5. Дана строка. Преобразовать ее, удалив каждый символ * и повторив каждый символ, отличный от *.

6. Имеется строка, содержащая буквы латинского алфавита и цифры. Вывести на экран длину наибольшей последовательности цифр, идущих подряд.

7. Строка содержит слово. Проверить, будет ли оно читаться одинаково справа налево и слева направо (т.е. является ли оно палиндромом).

8. Удалить часть символьной строки, заключенный в скобки (вместе со скобками).

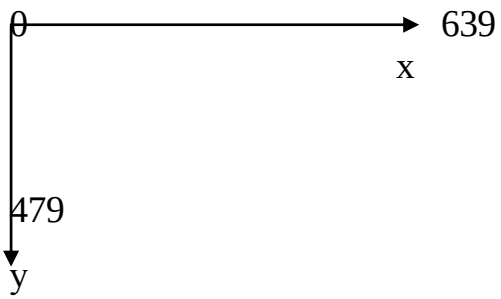
9. Дана строка, содержащая текст на русском языке. Определить, сколько раз встречается в ней самое длинное слово.

10. Дана строка, содержащая текст. Провести частный анализ текса, т.е. указать (в процентах), сколько раз встречается та или иная буква.

§12 Составление графических программ

Средства языка Паскаль позволяют создавать графические изображения на экране.

Графический экран монитора состоит из точек, которые можно засвечивать определенным цветом или гасить, в результате чего создается изображение. Эти точки называются пикселями. Количество точек может быть различным. Это зависит от качества монитора. Рассмотрим экран, который имеет размеры 640x480 точек. Каждая точка имеет две координаты (x,y).



Программы, которые выполняют графические построения, состоят из вызовов стандартных графических процедур и функций, собранных в модуле Graph.

Для построения изображений сначала нужно задать графический режим. Это делается так:

```
Var driver, mode: integer;
Begin
driver:= detect { detect – стандартная постоянная, задание графического режима}
initgraph (driver, mode, '');
```

Цвета в Паскале задают числами или английскими названиями

0	Black	Черный
1	Blue	Синий
2	Green	Зеленый
3	Cyan	Голубой
4	Red	Красный
5	Magenta	Фиолетовый
6	Brown	Коричневый
7	LightGray	Светло-серый
8	DarkGray	Темно-серый
9	LightBlue	Светло-синий
10	LightGreen	Светло-зеленый
11	LightCyan	Светло-голубой
12	LightRed	Светло-красный
13	LightMagenta	Светло-фиолетовый
14	Yellow	Желтый
15	White	Белый

Процедуры и функции для графических изображений

1	Initgraph (driver, mode, ″)	Задает графический режим
2	Setcolor (<цвет>)	Задает цвет будущего изображения
3	Setbkcolor (<цвет>)	Задает цвет фона
4	Putpixel (x,y,<цвет>)	Высвечивает точку (x, y) заданным цветом
5	Line (x1,y1, x2,y2)	Рисует линию между двумя заданными точками
6	LineTo (x1,y1, x2,y2)	Рисует линию от текущей точки до точки (x,y)
7	LineRel (dx, dy)	Рисует линию от текущей точки с заданными приращениями
8	Rectangle (x1, y1, x2, y2)	Рисует прямоугольник с заданными координатами диагонально противополож-ных вершин (верхней левой и нижней правой)
9	Bar (x1, y1, x2, y2)	Рисует закрашенный прямоугольник
10	Bar (x1, y1, x2, y2, <объемная глубина>, true)	Рисует параллелепипед
11	Circle (x, y, R)	Рисует окружность радиуса R с центром в точке (x, y)
12	Arc (x, y, <начальный угол>, <конечный угол>, R)	Рисует дугу
13	Piecslice (x, y, <начальный угол>, <конечный угол>, R)	Рисует закрашенный сектор

14	Ellipse (x, y, <начальный угол>, <конечный угол>, <вертикальный радиус>, <горизонтальный радиус>)	Рисует эллипс или дугу эллипса
15	Setfillstyle (<заполнение>, <цвет>)	Задает способ заполнения замкнутой области в зависимости от параметра заполнения: 0- заполнение цветом фона, 1- сплошное заполнение, 2- заполнение толстыми горизонтальными линиями, 3- заполнение наклонными линиями, …, 10 – заполнение точками, 11- плотное заполнение точками
16	Floodfill (x, y, <цвет границы>)	Заполняет замкнутую область, в которой находится точка (x, y)
17	Outtext (<текст>)	Выводит заданный текст на текущей позиции
18	Outtext (x, y, <текст>)	Выводит текст в заданной точке (x, y)
19	Settextstyle (<шрифт>, <направление>, <размер>)	Задает вид символов, направление вывода: 0 – горизонтально, 1 – вертикально, размеры символов: 1, 2, 3
20	Closegraf	Закрывает графический режим

Пример. Нарисовать десять концентрических окружной сообщим центром в центре экрана и описать вокруг них красный прямоугольник.

```
Program Circle10;
Uses Crt,  Graph;
Var  driver,  mode, r: integer;
begin
      driver:= detect;
      initgraph (driver, mode, ‘’);
      r:=10
  while  r<100 do
  begin
    setcolor (r div 10);
    circle (320, 240, r);
    r:=r+10;
  end;
  setcolor (4);
  rectangle (220,240, 420, 340);
  readln;
end.
```

Правила построения сложных изображений

1. *Обязательно начертите изображение на бумаге.*
2. *Разбейте изображение на графические примитивы (точки, отрезки, дуги)*
3. *Определите координаты всех опорных точек, значения радиусов окружностей, исходя из размеров экрана.*
4. *Определите последовательность построения и закраски замкнутых контуров.*
5. *Напишите программу, снабжая ее как можно большим количеством комментариев и отладьте ее.*

Совет

Запускать программу необходимо не после ее полного написания, а после уже первого логически законного фрагмента изображения, чтобы иметь возможность сразу оценивать правильность того или иного программного решения.

Динамическая графика

Чередование засвечиваний и гашений изображения используют для имитации движения этого изображения на экране. Перед очередным засвечиванием объект необходимо переместить в направлении его движении, а предыдущее изображение гасят цветом фона. Движение изображения на экране называют анимацией.

Для имитации движения объекта на экране нужно выполнить такой алгоритм:

1. *Нарисовать объект в нужной точке, сделать паузу.*
2. *Удалить объект, закрасив его цветом фона.*
3. *Изменить координаты объекта.*

Возвратиться к пункту 1.

Упражнения

1. Составить и выполнить программу, которая построит треугольник с вершинами в пикселях (100, 100), (150, 100), (80,170).

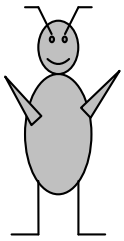
2. Составить и выполнить программу, которая построит квадрат со стороной 50 пикселей, центр которого совмещен с центром экрана.

3.Составить и выполнить программу, которая строит декартову систему координат с началом координат в центре экрана.

4. Составить и выполнить программу, которая строит стилизованное изображение домика.

5. Составить и выполнить программу, которая строит стилизованное изображение часов.

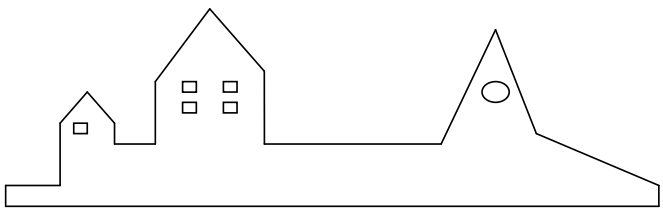
6. Составить и выполнить программу, которая выводит на экран данную фигуру. Цвета выбрать произвольно.



7. Составить и выполнить программу, которая строит стилизованный светофор.

8. Составить и выполнить программу, которая строит стилизованное изображение машины.

9. Составить и выполнить программу, которая рисует на экране данную фигуру. Цвета выбрать произвольно.



10. Составить и выполнить программу, которая рисует на экране пятиконечную звезду.

§9 Подпрограммы

При решении некоторых задач приходится выполнять одну и туже операцию несколько раз. Например, при вычислении площади пятиугольника он разбивается на несколько треугольников, вычисляется их площадь, а затем суммируется. Исполнитель данного алгоритма будет вычислять площадь треугольника 3 раза, использую разные значения.

В таких задачах используются вспомогательные небольшие программки, называемые подпрограммами.

Подпрограмма – это специальным образом оформленный алгоритм, который может многократно использоваться при решении более общей задачи.

В Паскале существуют два типа подпрограмм: подпрограммы-функции и подпрограммы- процедуры.

***Подпрограмма-процедура** имеет следующий формат описания:*

Procedure <имя процедуры> (список формальных параметров);

<Описательная часть>;

Begin

<тело процедуры>

End.

Формальные параметры используются в процедуре для определения типа и места подстановки фактических параметров. Количество и тип формальных параметров должны совпадать в точности их следования.

Подпрограмма-функция имеет следующий формат описания:

Function <имя> (список формальных параметров): <тип результата>;

<Описательная часть>

Begin

<тело процедуры>

End.

Упражнения

1. Написать функцию, которая вычисляет объем цилиндра. Параметрами функции должны быть радиус и высота.

2. Написать функцию для решения квадратного уравнения. Параметрами функции должны быть коэффициенты и корни уравнения.

3. Написать процедуру, которая вычисляет объем и площадь поверхности параллелепипеда.

4. Треугольник задан координатами своих вершин. Составить программу вычисляющую его площадь.

5. Вычислить площадь правильного шестиугольника со стороной a, используя подпрограмму вычисления площади треугольника.

6. Составить программу вычисления суммы факториалов всех чисел от 1 до 9.

7. Составить программу нахождения суммы большего и меньшего из трех чисел.

Литература

1. Информатика. Задачник-практикум. Т.1 /Под ред. Семакина И.Г., Е.К.Хеннера, М.: БИНОМ. Лаборатория знаний, 2002г.
2. Лапчик М.П. Вычисления. Алгоритмизация. Программирование. /Пособие для учителя. М.: Просвещение, 1988г.
3. Заварыкин В.М. и др. Основы информатики и ВТ. /Уч.пособие для студентов пед. ин-тов по физ-мат. спец. М.: Просвещение, 1989г.
4. Урнов В.А., Климов Д.Ю. Преподавание информатики в компьютерном классе. /Книга для учителя. М.: Просвещение, 1990г.
5. Лукин С.Н. Turbo Pascal 7.0. Самоучитель для начинающих. М.: «Диалог-МИФИ», 2000г.
6. Абрамов С.А., Зима Е.В. Начала информатики. М.: Наука, 1989г.
7. Епанешников А.М., Епанешников В.А. Программирование в среде Turbo Pascal 7.0., М.: Диалог-МИФИ, 1996г.
8. Культин Н.Б. Turbo Pascal в задачах и примерах, Санкт-Петербург, 2000г.
9. Коффман Эллиот Turbo Pascal, М.: «Вильямс», 2000г.
10. Королева Н.А., Солопанова Н.Л. Основы программирования на языке Турбо Паскаль, Тамбов, 2003 г.
11. Глинский Я.Н., Анохин В.Е. Turbo Pascal 7.0 Delphi. Учебное пособие. – СПб: ООО «ДиаСофтЮП», 2001 г.