

0

1 курс

**ПЛАН – КОНСПЕКТ**  
проведения занятий (лекционного, практического) по дисциплине  
«Информатика»

**Раздел «Основы алгоритмизации и программирования.»**

**Тема: «Общие принципы построения базовых алгоритмических структур в среде программирования.»**

**часть 4**

Подготовил: преподаватель  
В.Н. Борисов

Рязань 2024

## Вопросы занятий:

1. Встроенные функции.
2. Правила записи математических выражений.
3. Компьютерное моделирование с помощью языка программирования.
4. Этапы решения тематических задач по физике, по геометрии с помощью компьютера.
5. Компьютерное моделирование задач с применением алгоритмической структуры «Следование» (практическое занятие, теоретическая часть).

**Время проведения занятий (лекционного, практического) – 4 часа.**

### **Первый вопрос: Встроенные функции Borland Pascal.**

**Процедура** (подпрограмма) – это часть программы, которая содержит набор операторов, выполняющих определенные действия над переданными ей данными (параметрами, аргументами). Процедура может быть вызвана из любой точки программы, причем при каждом вызове ей могут передаваться различные значения параметров.

**Функция** – это процедура, возвращающая результат, что позволяет использовать ее в выражениях.

В заголовке процедур и функций обычно описывается имя процедуры и список **формальных** параметров, то есть тех значений, которые анализирует процедура при своем выполнении.

Параметры, передаваемые в процедуру из программы при ее вызове называются **фактическими**. Дополнительно, в заголовке функции описывается тип возвращаемого функцией значения.

**Borland Pascal** имеет множество встроенных процедур и функций. Типичным примером могут служить арифметические и алгебраические функции: **sin, cos, arctan, abs, sqrt** и т.д. Каждая из них имеет только один числовой параметр, передаваемый ей в скобках и возвращает некоторое значение.

В языке Паскаль существует ряд заранее разработанных подпрограмм-функций, которые можно использовать как готовые объекты. В Turbo Pascal их количество увеличено по сравнению со стандартом языка, и все они объединены в стандартные модули. Рассмотрим наиболее часто используемые стандартные функции. Существуют стандартные функции ввода-вывода и динамического распределения памяти, стандартные функции (работы со строками, указателями и адресами и т. д.),

### Арифметические функции.

Функция	Назначение	Тип результата
Abs(X)	Абсолютное значение аргумента	Совпадает с типом X
Arctan(X)	Арктангенс аргумента	Вещественный
Cos(X)	Косинус аргумента	Вещественный
Exp(X)	$e^x$	Вещественный
Frac(X)	Дробная часть числа	Вещественный
Lnt(X)	Целая часть числа	Вещественный
Ln(X)	Натуральный логарифм	Вещественный
Pi	Значение величины $Pi=3.1415926535897932385$	Вещественный
Sin(X)	Синус аргумента	Вещественный
Sqr(X)	Квадрат аргумента	Совпадает с типом X
Sqrt(X)	Квадратный корень аргумента	Вещественный

### Функции преобразования типа.

Эти функции предназначены для преобразования типов величин, например вещественного числа в целое и т. д. К ним относятся следующие функции:

Chr (X) - преобразование ASCII-кода в символ.

Аргумент функции должен быть целого типа в диапазоне (0..255). Результатом является символ, соответствующий данному коду.

High(X) - получение максимального значения величины.

Аргумент функции - параметр или идентификатор порядкового типа, типа-массива типа-строки или открытый массив.

Результат функции для величины порядкового типа - максимальное значение этой величины, типа-массива - максимальное значение индекса, типа-строки - объявленный размер строки, открытого массива - количество компонент массива минус 1 (максимальный индекс, при начале нумерации с нуля).

Low(X) - получение минимального значения величины.

Аргумент функции - параметр или идентификатор порядкового типа, типа-массива, типа-строки или открытый массив. Результат функции для

величины порядкового типа - минимальное значение этой величины, типа-массива - минимальное значение индекса, типа-строки или открытого массива - 0.

Ord(X) - преобразование любого порядкового типа в целый тип.

Аргументом функции может быть величина любого порядкового типа (логический, символьный, перечисляемый). Результатом является величина типа Longint.

Round (X) - округление вещественного числа до ближайшего целого.

Аргумент функции - величина вещественного типа, а результат - округленная до ближайшего целого величина типа Longint. Если результат выходит за диапазон значений Longint, то при выполнении программы возникает ошибка.

Trunc(X) - получение целой части вещественного числа.

Аргумент функции - величина вещественного типа, а результат - целая часть этого числа. Тип результата - Longint. Если результат выходит за диапазон значений Longint, то во время выполнения программы возникает ошибка.

### **Функции для величин порядкового типа.**

Эти функции позволяют выполнить ряд действий над величинами порядкового  $i$  типа (найти предыдущий или последующий элемент, проверить число на нечетность). К этим функциям относятся следующие:

Odd(X) - проверка величины X на нечетность.

Аргументом функции является величина типа Longint, результат равен True, если аргумент нечетный, и False - если четный.

Pred(X) - определение предыдущего значения величины X.

Аргументом функции является величина любого порядкового типа, результатом - предшествующее значение того же типа (например, Pred(2) равно 1). При применении функции к первому элементу последовательности возникает ошибка.

Succ(X) - определение последующего значения величины X.

Аргументом функции является величина любого порядкового типа, результатом - последующее значение того же типа (например, Succ(2) равно 3). При применении функции к последнему элементу последовательности возникает ошибка.

**Второй вопрос: Правила записи математических выражений.**

# \* Арифметические выражения

▫ Turbo Pascal

## Правила записи арифметических выражений

1. Все выражения записываются в одну строку.
2. Все операции выполняются в порядке приоритета.
3. Приоритет можно изменить скобками.

Очень  
важно!!!



$$b + (4.5 * c - \text{EXP}(d - 1.4)) / 5.8$$

В соответствии с приоритетом, вычисления в этом примере выполняются в следующем порядке:

$$a1 = d - 1.4$$

$$a2 = e^{a1}$$

$$a3 = 4.5 * c$$

$$a4 = a3 - a2$$

$$a5 = a4 / 5.8$$

$$A = b + a5 \text{ (результат)}$$

## Примеры записи арифметических выражений

Эти выражения записаны правильно:

$$a + b / c - d$$

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

Эти выражения записаны неправильно:  
 $2 * b$  - два знака операции следуют подряд;

$3a + c$  - пропущен знак умножения;

$\text{SIN } x + b$  - отсутствуют скобки для аргумента функции.

## Арифметические функции языка Pascal

Функция	Описание	Пример
<b>abs</b> (x:R) :R <b>abs</b> (y:I) :I	вычисление модуля числа $ x ,  y $	<b>abs</b> (-5.5) $\Rightarrow$ 5.5 <b>abs</b> (-12) $\Rightarrow$ 12
<b>frac</b> (x:R) :R	вычисление дробной части числа	<b>frac</b> (5.18) $\Rightarrow$ 0.18 <b>frac</b> (4.58) $\Rightarrow$ 0.58
<b>int</b> (x:R) :R	вычисление целой части числа	<b>int</b> (5.18) $\Rightarrow$ 5.0 <b>int</b> (4.58) $\Rightarrow$ 4.0
<b>trunc</b> (x:R) :I	вычисление целой части числа	<b>trunc</b> (5.18) $\Rightarrow$ 5 <b>trunc</b> (4.58) $\Rightarrow$ 4
<b>round</b> (x:R) :I	округление числа	<b>round</b> (5.18) $\Rightarrow$ 5 <b>round</b> (4.58) $\Rightarrow$ 5
<b>sqr</b> (x:R) :R <b>sqr</b> (y:I) :I	возведение числа в квадрат	<b>sqr</b> (2.5) $\Rightarrow$ 6.25 <b>sqr</b> (9) $\Rightarrow$ 81
<b>sqrt</b> (x:R) :R	вычисление квадратного корня из числа	<b>sqrt</b> (4.84) $\Rightarrow$ 2.2 <b>sqrt</b> (36) $\Rightarrow$ 6.0

I – целые числа (integer, byte),  
R – вещественные числа (real)

## Вещественный тип

+ - \* /

= <> <> <=> >=

Abs(x) – модуль x

Sqr(x) – квадрат x

Sqrt(x) – корень квадратный из x

Exp(x) – экспонента, e в степени x

Sin(x) – синус x

Cos(x) – косинус x

Ln(x) – натуральный логарифм x

Arctan(x) – арктангенс x

Trunc(x) – отбрасывает вещественную часть (результат – целый)

Round(x) – округляет до целого (результат – целый)

## Правила записи арифметических выражений

Запись должна линейной

$$A^2 + B^2 - 12C$$

на Паскале записывается так:

$$A*A + B*B - 12*C$$

или

$$SQR(A) + SQR(B) - 12*C$$

Порядок выполнения операций (без скобок или внутри скобок):

1. Вычисление функций
2. Умножение и деление
3. Сложение и вычитание

### Третий вопрос: Компьютерное моделирование с помощью языка программирования.

Моделирование является одним из способов познания мира.

Понятие моделирования достаточно сложное, оно включает в себя огромное разнообразие способов моделирования: от создания натуральных моделей (уменьшенных и или увеличенных копий реальных объектов) до вывода математических формул.

Для различных явлений и процессов бывают уместными разные способы моделирования с целью исследования и познания.

Объект, который получается в результате моделирования, называется моделью. Должно быть понятно, что это совсем не обязательно реальный объект. Это может быть математическая формула, графическое представление и т.п. Однако он вполне может заменить оригинал при его изучении и описании поведения.

Хотя модель и может быть точной копией оригинала, но чаще всего в моделях воссоздаются какие-нибудь важные для данного исследования элементы, а остальными пренебрегают. Это упрощает модель. Но с другой стороны, создать модель – точную копию оригинала – бывает абсолютно нереальной задачей.

Например, если моделируется поведение объекта в условиях космоса. Можно сказать, что модель – это определенный способ описания реального мира.

Моделирование проходит три этапа:

- Создание модели.
- Изучение модели.
- Применение результатов исследования на практике и/или формулирование теоретических выводов.

Видов моделирования огромное количество. Вот некоторые примеры типов моделей:

- Математические модели. Это знаковые модели, описывающие определенные числовые соотношения.
- Графические модели. Визуальное представление объектов, которые настолько сложны, что их описание иными способами не дает человеку ясного понимания. Здесь наглядность модели выходит на первый план.
- Имитационные модели. Позволяют наблюдать изменение поведения элементов системы-модели, проводить эксперименты, изменяя некоторые параметры модели.

Над созданием модели могут работать специалисты из разных областей, т.к. в моделировании достаточно велика роль межпредметных связей.

### Особенности компьютерного моделирования.

Совершенствование вычислительной техники и широкое распространение персональных компьютеров открыло перед моделированием огромные перспективы для исследования процессов и явлений окружающего мира, включая сюда и человеческое общество.

Компьютерное моделирование – это в определенной степени, то же самое, описанное выше моделирование, но реализуемое с помощью компьютерной техники.

Для компьютерного моделирования важно наличие определенного программного обеспечения.

При этом программное обеспечение, средствами которого может осуществляться компьютерное моделирование, может быть как достаточно универсальным (например, обычные текстовые и графические процессоры), так и весьма специализированными, предназначенными лишь для определенного вида моделирования.

Очень часто компьютеры используются для математического моделирования. Здесь их роль неопределима в выполнении численных операций, в то время как анализ задачи обычно ложится на плечи человека.

Обычно в компьютерном моделировании различные виды моделирования дополняют друг друга. Так, если математическая формула очень сложна, что не дает явного



представления об описываемых ею процессах, то на помощь приходят графические и имитационные модели. Компьютерная визуализация может быть намного дешевле реального создания натуральных моделей.

С появлением мощных компьютеров распространилось графическое моделирование на основе инженерных систем для создания чертежей, схем, графиков.

Если система сложна, а требуется проследить за каждым ее элементом, то на помощь могут прийти компьютерные имитационные модели. На компьютере можно воспроизвести последовательность временных событий, а потом обработать большой объем информации.

Однако следует четко понимать, что компьютер является хорошим инструментом для создания и исследования моделей, но он их не придумывает. Абстрактный анализ окружающего мира с целью воссоздания его в модели выполняет человек.

Компьютерное моделирование – метод решения задачи анализа или синтеза сложной системы на основе использования ее компьютерной модели.

Суть компьютерного моделирования заключена в получении количественных и качественных результатов по имеющейся модели. Качественные выводы, получаемые по результатам анализа, позволяют обнаружить неизвестные ранее свойства сложной системы: ее структуру, динамику развития, устойчивость целостность и т.д. компьютерное моделирование для рождения новой информации использует любую информацию, которую можно актуализировать с помощью ПК.

Компьютерная модель сложной системы должна по возможности отображать все основные факторы и взаимосвязи, характеризующие реальные ситуации, критерии и ограничения. Модель должна быть универсальной, что бы по возможности описывать близкие по назначению объекты, и в тоже время достаточно простой. Что бы позволить выполнить необходимые исследования с разумными затратами.

Результативность компьютерной модели определяется качеством использованного программного обеспечения. Основные требования, предъявляемые к программам – это, конечно простота ввода и корректировки исходных данных, а также визуализация программирования (специальные программы, в которых реализуются удобные графические пользовательские возможности).

Алгоритмические языки программирования издавна используются для построения моделей. Если нет возможности использовать для построения моделей другие средства, то с помощью языков программирования можно строить модели из самых различных классов моделей (физические и логические, геометрические и экологические и т.п.).

Компьютерная модель — модель, реализованная средствами программной среды. Компьютерная модель, или численная модель — компьютерная программа, работающая на отдельном компьютере, суперкомпьютере или множестве взаимодействующих компьютеров (вычислительных узлов), реализующая представление объекта, системы или понятия в форме, отличной от реальной, но приближенной к алгоритмическому описанию, включающей и набор данных,

характеризующих свойства системы и динамику их изменения со временем. Пути построения моделей 1. построение модели с помощью одного из приложений: текстовых редакторов, графических редакторов, электронных таблиц, СУБД, Macromedia Flash и др. 2. построение алгоритма решения задачи и его кодировка на одном из языков программирования (Visual Basic, Паскаль, Basic и т. д.)

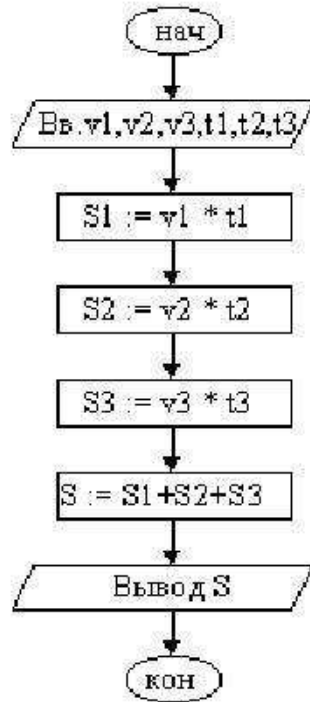
**Четвертый вопрос: Этапы решения тематических задач по физике, по геометрии с помощью компьютера.**

## Этапы решения задачи на компьютере



## Основные алгоритмические структуры

### • **Следование**



Пешеход шел по пересеченной местности. Его скорость движения по равнине  $v_1$  км/ч, в гору —  $v_2$  км/ч и под гору —  $v_3$  км/ч. Время движения соответственно  $t_1$ ,  $t_2$  и  $t_3$  ч. Какой путь прошел пешеход?

1. Ввести  $v_1, v_2, v_3, t_1, t_2, t_3$ .
2.  $S_1 := v_1 * t_1$ .
3.  $S_2 := v_2 * t_2$ .
4.  $S_3 := v_3 * t_3$ .
5.  $S := S_1 + S_2 + S_3$ .
6. Вывести значение  $S$ .
7. Конец.

# Задача на линейный алгоритм

**Вводятся объём и масса тела. Определить плотность материала этого тела.**

## Входные данные:

V – объём тела (real);

m - масса тела (real);

## Выходные данные:

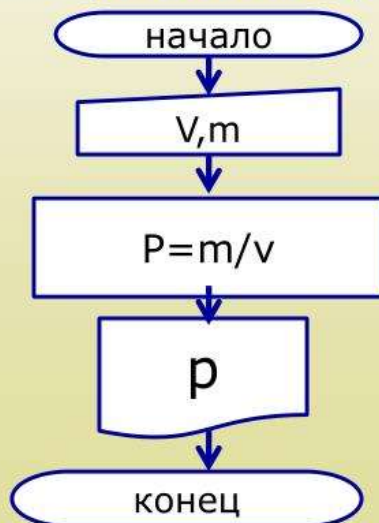
p – плотность тела (real).

## Паскаль-программа

```

Program z1;
  uses crt;
  Var V,m,p:real;
BEGIN
  Clrscr;
  Write('Введите значения объёма и
  массы тела: ');
  ReadLn(v,m);
  P:=m/V;
  Write('Плотность тела
  объёмом',V:7:1,' и массой' ,m:7:1,'
  равна',p:8:2);
  ReadLn;
END.
  
```

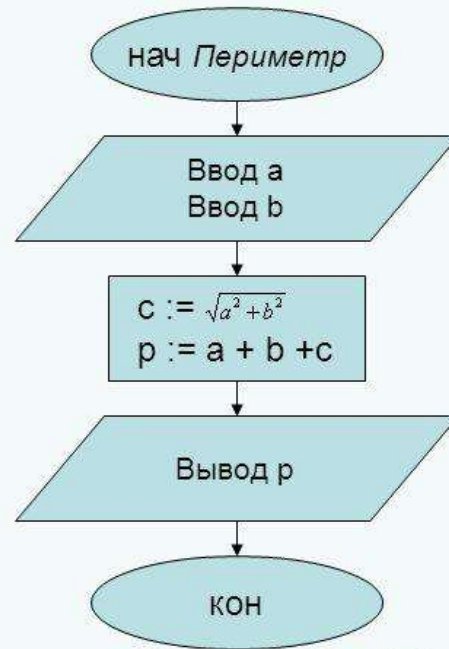
## БЛОК-СХЕМА



## Линейные алгоритмы

**Задача 2.** Написать программу нахождения периметра прямоугольного треугольника по известным катетам **a** и **b** (**a** и **b** – целые числа, вводимые с клавиатуры).

```
алг периметр
цел a, b
вещ c, p
нач
    ввод a
    ввод b
     $c := \sqrt{a^2 + b^2}$ 
     $p := a + b + c$ 
    вывод p
кон
```



# Запись алгоритма

Блок-схема	Программа на Паскале
<pre> graph TD     Start([начало]) --&gt; A[A=4]     A --&gt; B[B=3]     B --&gt; C["C = √(A² + B²)"]     C --&gt; P["P = A + B + C"]     P --&gt; Output[/Вывод C, P/]     Output --&gt; End([конец]) </pre>	<p>Ввод данных с помощью присваивания</p> <pre> Var A, B, C, P : Real;  Begin   A:=4;   B:=3;   C:=Sqrt(Sqr(A)+Sqr(B));   P:=A+B+C;   Writeln ('C=',C:5:2);   Writeln ('P=',P:5:2);   Readln End. </pre>

22.07.2015

# Пример линейного алгоритма и программы в Паскале

**Задача 1.** Составить алгоритм сложения двух чисел  $A$  и  $B$ , вводимых с клавиатуры. Результат сложения записать в виде переменной  $C$ . По данному алгоритму написать программу в Паскале

## Алгоритмический язык

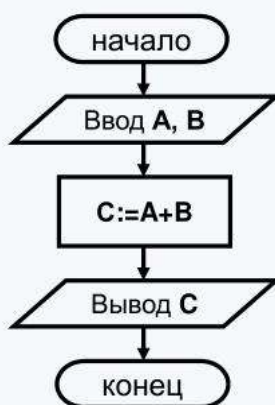
```

алг сложение
цел A, B, C
нач

    ввод A
    ввод B
    C := A + B
    вывод C

кон
  
```

## Блок-схема



## Программа в Паскале

```

Program summa;
Var
    a,b,sum : integer;

Begin
    readln(a);
    readln(b);
    sum := a + b;
    writeln(sum)

end.
  
```

**Пятый вопрос:** Компьютерное моделирование задач с применением алгоритмической структуры «Следование» (практическое занятие, теоретическая часть).

Моделирование полета тела, брошенного под углом к горизонту

1. *Постановка задачи*  
 Под углом  $\alpha$  к горизонту с начальной скоростью  $v_0$  брошен мячик.  
 Вопросы:  
 1) как долго будет находиться мячик в пути;  
 2) как далеко от места бросания упадет мячик;  
 3) чему равна максимальная высота подъема мячика при заданных начальных условиях;  
 4) через какой промежуток времени с момента бросания достигается максимальная высота?  
 В качестве начальных условий принять  $v_0 = 25$  м/с,  $\alpha = 30^\circ$ .

2. *Разработка плана создания модели*  
 Объектом исследования является положение в пространстве летящего мячика в разные моменты времени. Исходными данными являются: начальная скорость  $v_0$ , угол бросания  $\alpha$ . Для ответа на поставленные в задаче вопросы необходимо вначале построить математическую модель, выражающую зависимость положения (координат) тела от исходных данных и времени.

3. *Создание модели*  
 а) *построение математической модели*  
 Для решения задачи необходимо сделать некоторые допущения:  
 1) считать мячик материальной точкой, поскольку он мал по сравнению с Землей;  
 2) пренебречь кривизной Земли, считая ее поверхность плоской;

3) пренебречь движением Земли;  
 4) считать ускорение свободного падения  $g$  постоянным;  
 5) пренебречь сопротивлением воздуха.

Из курса физики известно, что при таких допущениях положение тела (мячика) в пространстве, т. е. значения координат дальности полета  $x$  и высоты  $y$  в момент времени  $t$ , описывается системой уравнений:

$$\begin{cases} x(t) = (v_0 \cos \alpha)t, \\ y(t) = (v_0 \sin \alpha)t - \frac{gt^2}{2} \end{cases} \quad (7)$$

б) компьютерная реализация модели  
 Воспользуемся общей схемой размещения модели в электронных таблицах.

4

---

Лекция 3    Информационные системы и сети    3 курс    2014-15 уч. год

Значение начальной скорости и соответствующий комментарий поме-

Значение угла бросания в градусах и соответствующий комментарий поместим в ячейки A4 и B4.

Для представления угла бросания в радианах воспользуемся ячейкой A5. Соответствующий комментарий разместим в ячейке B5. Такое представление связано с тем, что в электронных таблицах аргументами тригонометрических функций, в том числе, COS и SIN, являются радианы.

	A	B	C
1	МОДЕЛЬ ПОЛЕТА МЯЧИКА		
2	Исходные данные		
3	25	Начальная скорость, м/с	
4	30	Угол бросания, град	
5	ТАБЛИЦА (A4) Угол бросания, рад		
6	0,1	Временной шаг, с	
7			
8	Расчетная таблица		
9	Время	Координата (x)	Координата (y)
10	0	0	0
11	=A9+A8	=A5*A11*COS(A6)	=A5*A11*SIN(A6)-0,5*A11^2

Положение мячика в полете будем рассматривать через одинаковый промежуток времени, или временной шаг. Для нашей задачи зададим его равным 0,1 с. Поместим значение временного шага и комментарий к нему в ячейки A6 и B6 соответственно.



