

Линейный вычислительный процесс

- Линейным называется вычислительный процесс, этапы которого выполняются однократно и последовательно один за другим.
- Например, вычисление значения функции по формуле.

Этапы линейного вычислительного процесса –

- ввод исходных данных,
- вычисление значений искоемых переменных,
- вывод на экран результатов вычислений –

выполняются однократно и последовательно друг за другом вне зависимости от исходных данных

Пример

Лабораторная работа №2.

Разработка линейной программы

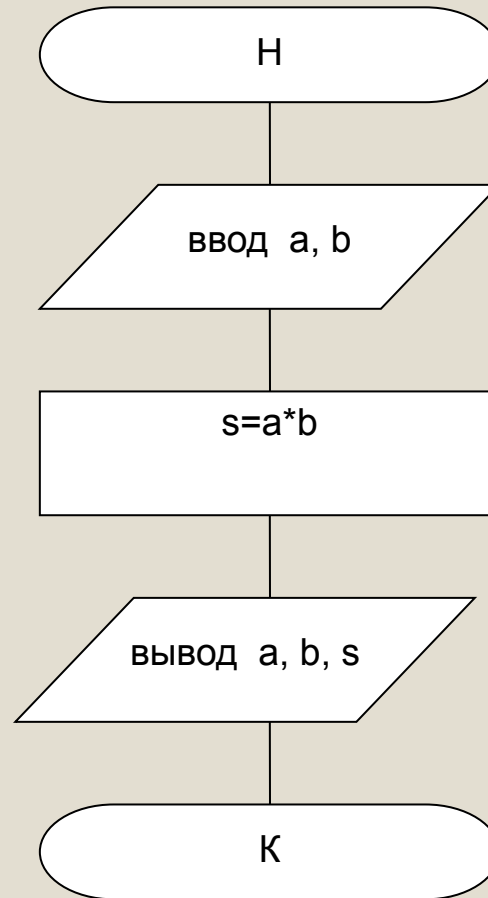
Цель работы: получение первичных навыков программирования (разработки простейших программ линейной структуры).

Задание по лабораторной работе

Составить и отладить программу, обеспечивающую решение задачи, формулировка которой приведена ниже. При тестировании программы на ПК организовать минимум трехкратное исполнение для разных исходных данных.

Вариант 28. По заданным сторонам A и B прямоугольника определить его площадь.

Схема алгоритма



Для программирования в настоящей работе достаточно использовать следующие предложения

- заголовок программы с именем программы (PROGRAM);
- комментарий с указанием фамилии и имени студента, шифра учебной группы, номера работы, номера варианта ({.....});
- описание используемых в программе переменных (VAR);
- начало исполняемой части программы (BEGIN);
- операторы вывода пользователю подсказок, какие исходные данные он должен ввести в компьютер (WRITE или WRITELN);
- операторы ввода исходных данных (READ или READLN);
- операторы присваивания, предусматривающие необходимые вычисления;
- оператор вывода форматированных результатов на экран монитора (WRITE или WRITELN);
- завершающее текст программы предложение (END.)

Структура программы

PROGRAM Имя программы;

VAR

Определение глобальных переменных программы;

BEGIN

Основной блок программы (тело программы)

.....

END.

1 блок программы

program labrab2;

{Иванов А., 15-ИЭ, лаб. раб. 2, вариант 28}

2 блок – описание переменных

Синтаксис блока *VAR*

Имя_переменной1: имя_типа1;

Имя_переменной2: имя_типа2;

.....;

Имя_переменной3, Имя_переменной4,

Имя_переменной5 : имя_типа3;

.....;

Простые типы

Название	Идентификатор	Множество допустимых значений
Порядковый		
Короткий целый	ShortInt	-128..127
Байтовый	Byte	0..255
Слово	Word	0..65 535
Целый	Integer	-32 768..32 767
Длинный целый	LongInt	-2 147 483 648..2 147 483 647
Символьный	Char	Символы из расширенного набора символов кода ASCII
Булев	Boolean	True, False
Вещественный		
Вещественный	Real	$-1,7 \times 10^{38} .. -2,9 \times 10^{-39}$, $2,9 \times 10^{-39} .. 1,7 \times 10^{38}$
С одинарной точностью	Single	$-3,4 \times 10^{38} .. -1,5 \times 10^{-45}$, $1,5 \times 10^{-45} .. 3,4 \times 10^{38}$
С двойной точностью	Double	$-1,7 \times 10^{308} .. -5,0 \times 10^{-324}$, $5,0 \times 10^{-324} .. 1,7 \times 10^{308}$
С повышенной точностью	Extended	$-1,1 \times 10^{4932} .. -1,9 \times 10^{-4951}$, $1,9 \times 10^{-4951} .. 1,1 \times 10^{4932}$
Сложный	Comp	$-2^{63} + 1 .. 2^{63} - 1$

Целые числа

ЦЕЛЫЕ

типы	диапазон значений	требуемая память
Shortint	-128 .. 127	1 байт
Integer	-32768 .. 32767	2 байта
Longint	-2147483648 .. 2147483647	4 байта
Byte	0 .. 255	1 байт
Word	0 .. 65535	2 байта

Над целыми операндами можно выполнять следующие арифметические операции: сложение, вычитание, умножение, деление, получение остатка от деления. Знаки этих операций:

+ - * div mod

Результат арифметической операции над целыми операндами есть величина целого типа. Результат выполнения операции деления целых величин есть целая часть частного. Результат выполнения операции получения остатка от деления - остаток от деления целых. Например:

$$17 \operatorname{div} 2 = 8, \quad 3 \operatorname{div} 5 = 0.$$
$$17 \operatorname{mod} 2 = 1, \quad 3 \operatorname{mod} 5 = 3$$

Вещественные числа

Действительный тип	Диапазон значений	Количество цифр мантиссы	Требуемая память (байт)
Real	$2.9e-39 \dots 1.7e+38$	11	6
Single	$1.5e-45 \dots 3.4e+38$	7	4
Double	$5.0e-324 \dots 1.7e+308$	15	8
Extended	$3.4e-4932 \dots 1.1e+4932$	19	10
Comp	$-9.2e+18 \dots 9.2e+18$	19	8

Примеры

VAR

KOL_VO:

INTEGER;

VEC:

REAL;

SRED_BALL:

REAL;

OCENKA:

INTEGER;

KOL_VO, OCENKA: **INTEGER;**

SRED_BALL, VEC: **REAL;**

Для нашей задачи:

VAR A, B, S: REAL;

Начало и конец

BEGIN

Текст программы:

операции (операторы языка), описывающие выполнение алгоритма

END.

Операторы ввода

```
Read(<Список ввода>);  
Readln(<Список ввода>);
```

Команды позволяют вводить данные в переменные во время выполнения программы с клавиатуры.

Элементами списка ввода могут быть имена переменных, которые должны быть заполнены значениями, введенными с клавиатуры.

Разница между работой процедур `Read` и `Readln` (от `Read line`) состоит в следующем: после выполнения `Read` значение следующего данного считывается с этой же строки, а после выполнения `Readln` - с новой строки.

`ReadLn`; (оператор реализует пропуск строки исходных данных)

Операторы ввода

```
Read(<Список ввода>);  
Readln(<Список ввода>);
```

Выполнение операторов ввода происходит так: ход программы приостанавливается, на экран выводится курсор, компьютер ожидает от пользователя набора данных для переменных, имена которых указаны в списке ввода.

Пользователь с клавиатуры вводит необходимые значения в том порядке, в котором они требуются списком ввода, нажимает Enter.

После этого набранные данные попадают в соответствующие им переменные и выполнение программы продолжается.

Примечание: данные при вводе разделяются пробелами или нажатием клавиши Enter

Операторы вывода

Write(<Список вывода>);

WriteLn(<Список вывода>);

Операторы **Write** и **WriteLn** позволяют выводить на экран монитора данные из списка вывода.

Элементами списка вывода могут являться имена переменных, выражения, константы.

Элементы списка, также как и в операторах ввода, разделяются запятыми.

Операторы вывода

Write(<Список вывода>);

Writeln(<Список вывода>);

Различие между двумя операторами вывода таково: после выполнения оператора **Writeln** (от **Write line**) происходит переход на новую строчку, а после выполнения инструкции **Write**, переход на новую строчку не происходит и печать по последующим командам вывода **Write** или **Writeln** будет происходить на той же строчке.

При вызове оператора **Writeln** без параметров просто происходит переход на новую строчку.

Система форматов для вывода информации

В списке оператора **WRITE** следует указывать размер поля.

Для вывода числа 999 будут требоваться три позиции.

Для вывода числа -23.045 требуется семь позиций (с учетом знака и десятичной точки), из которых три позиции будет занимать дробная часть.

Система форматов для вывода информации

Если пользователь укажет больше позиций, чем необходимо, то левые от значения числа позиции заполнятся пробелами.

Если указанный размер поля меньше требуемого, то значение печатается без пробелов и учета указанного пользователем поля.

Для вывода вещественных данных указывается общая длина поля и количество позиций под дробную часть (в том числе).

Если требуется, то дробная часть числа округляется до указанного количества позиций.

Примеры вывода переменных d (=1234) и s (= -123.451)

Оператор	Результат	Примечание
writeln (d:4)	1234	
writeln (d:7)	___1234	
writeln (d:12)	_____1234	
writeln (d:2)	1234	вывод без учета указанного поля
writeln (s:9:3)	_-123.451	
writeln (s:8:1)	__-123.5	
writeln (s:12:2)	_____ -123.45	
writeln (s:4:3)	-123.451	вывод без учета указанного поля

Пример

```
WRITELN ('ВВЕДИТЕ ЗНАЧЕНИЯ СТОРОНЫ А ');  
READLN (A);  
WRITELN ('ВВЕДИТЕ ЗНАЧЕНИЯ СТОРОНЫ В ');  
READLN ( B);
```

Примечание:

Перед каждым оператором READLN следует указать соответствующий оператор WRITELN

```
.....  
WRITELN (' СТОРОНЫ ПРЯМОУГОЛЬНИКА= ',  
A:7:2,B:7:2,  
      ' ПЛОЩАДЬ = ',S:7:2);
```

Оператор присваивания

Оператор присваивания в языке Паскаль имеет следующую форму:

Переменная := Арифметическое выражение;

Арифметическое выражение может содержать числовые константы и переменные, знаки арифметических операций, круглые скобки, функции

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

Арифметические функции

ФУНКЦИЯ	НАЗНАЧЕНИЕ	ПРИМЕР ВЫЗОВА	РЕЗУЛЬТАТ
ABS(ЧИСЛО)	АБС. ЗНАЧЕНИЕ ЧИСЛА	ABS(-3.5)	+3.5
ARCTAN(ТАНГЕНС-УГЛА)	АРКТАНГЕНС ЧИСЛА	ARCTAN(0)	0
COS(УГОЛ)	КОСИНУС УГЛА(РАД.)	COS(PI)	-1
EXP(ЧИСЛО)	ЭКСПОНЕНТА	EXP(1)	2.718281828...
FRAC(ЧИСЛО)	ДРОБНАЯ ЧАСТЬ ЧИСЛА	FRAC(3.5)	0.5
INT(ЧИСЛО)	ЦЕЛАЯ ЧАСТЬ ЧИСЛА	INT(3.5)	3.0
LN(ЧИСЛО)	НАТ. ЛОГАРИФМ	LN(2.718281828)	~1.0
ODD(ЧИСЛО)	ПРОВЕРКА НЕЧЕТНОСТИ	ODD(3)	TRUE
PI	ЧИСЛО ПИ	PI	3.141592...
RANDOM(ЧИСЛО)	«СЛУЧАЙНОЕ» ЧИСЛО	RANDOM(10)	ЧИСЛО В [0;10]
SIN(УГОЛ)	СИНУС УГЛА(РАД.)	SIN(PI)	0
SQR(ЧИСЛО)	КВАДРАТ ЧИСЛА	SQR(2.0)	4.0
SQRT(ЧИСЛО)	КВАДРАТНЫЙ КОРЕНЬ	SQRT(25.0)	5.0

Пример

Текст программы

```
program labrab2 ;  
{Иванов А., 15-ИЭ, лаб. раб. 2, вариант 28}  
  
var a, b, s: real;  
begin  
    writeln ('введите значения стороны a ');  
    readln (a);  
    writeln ('введите значения стороны b ');  
    readln ( b);  
    s:=a*b;  
    writeln (' стороны прямоугольника= ', a:7:2,b:7:2,  
            ' площадь = ',s:7:2);  
end.
```