

***Начала  
программирования  
на языке Basic.  
Линейные программы.***

Борисов В.А.

КАСК – филиал ФГБОУ ВПО РАНХ и ГС

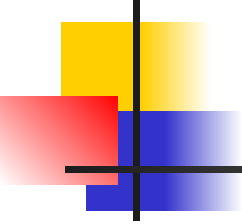
Красноармейск 2011 г.



# Общие положения

---

- Говоря о программировании, т.е. о составлении программы решения некоторой задачи, в дальнейшем будем считать, что схема алгоритма задачи задана.

- 
- 
- В зависимости от их структуры алгоритмы делятся на линейные, разветвляющиеся, циклические.

# Структура программы на языке Бейсик

1)

*REM* Название программы

2)

Оператор *DIM* (при наличии массивов)

3)

Операторы ввода (например, *INPUT*)

4)

Операторы вычисления и вывода результатов

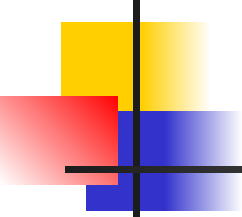
5)

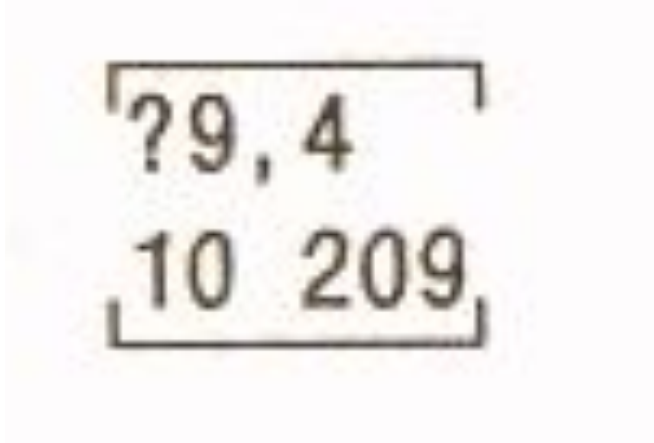
*END*



## Пример программы на языке Бейсик:

```
10 REM ВЫЧИСЛЕНИЕ ОРБИТЫ СПУТНИКА
20 INPUT X, Y
30 A=SQR(X)*3+2/Y^0.5
40 B=X^2+2*Y^3
50 PRINT A; B
60 END
```

- 
- 
- После выполнения этой программы при  $X=9$  и  $Y=4$  данные и результаты будут выведены на экран дисплея в таком виде:



```
?9, 4
10 209
```

# Рекомендации по организации программ

---

- Перед оператором INPUT следует записывать оператор вывода имен вводимых величин («подсказку») или использовать возможность вывода «подсказки» оператором INPUT.



# Рекомендации по организации программ

---

- В программе следует чаще использовать оператор REM.





# Рекомендации по организации программ

---

- В операторе PRINT следует предусматривать вывод имен выводимых величин.

# Рекомендации по организации программ

---

- В программе следует предусматривать вывод не только результатов решения задачи, но и всех исходных данных.

# Рекомендации по организации программ

---

- Не следует записывать в одной строке программы большое число операторов.



# Линейные программы

---

- Линейной называется программа, являющаяся записью линейного алгоритма.

# Составление простейших программ



---

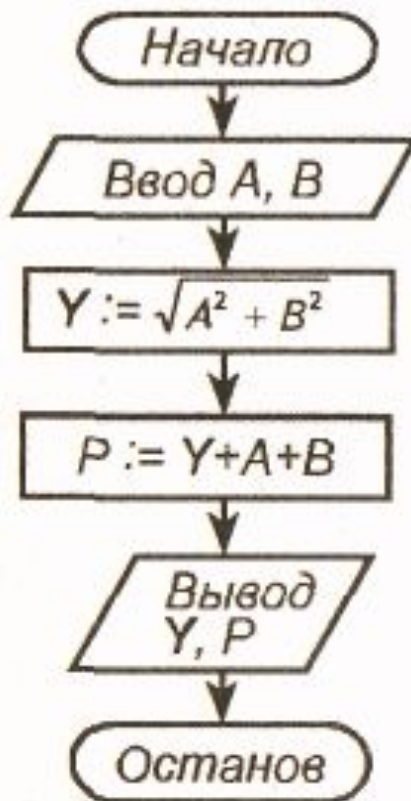
- Простейшими будем называть линейные программы, не содержащие массивов.

# Блоки схемы алгоритма по порядку

---

- для блока Начало — оператор REM с названием программы;
- для блока Ввод — оператор ввода;
- для блока Процесс — оператор присваивания;
- для блока Вывод — оператор вывода;
- для блока Останов — оператор END.

# Вычислить периметр прямоугольного треугольника, если заданы длины его катетов



Программа на Бейсике

```
10 REM ПЕРИМЕТР
```

```
20 INPUT A, B
```

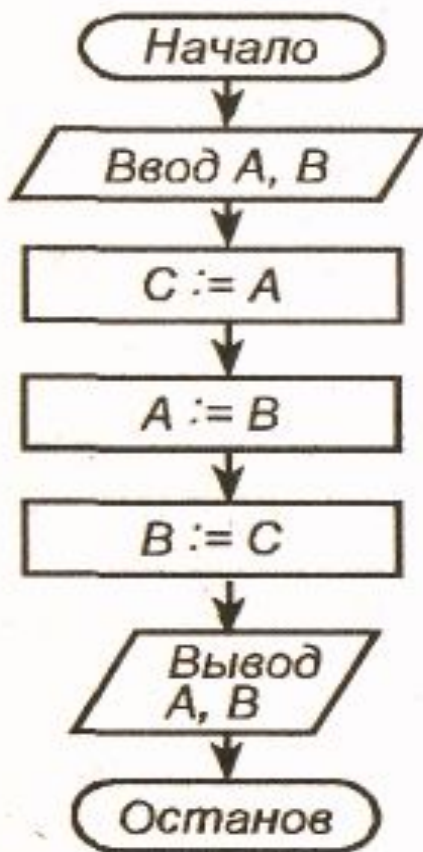
```
30 Y=(A^2+B^2)^0,5
```

```
40 P=Y+A+B
```

```
50 PRINT "Y="; Y; "P="; P
```

```
60 END
```

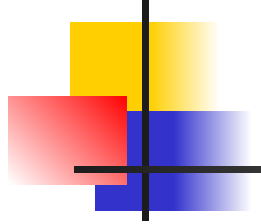
# Переставить значения величин А и В



Программа на Бейсике

```
10 REM ПЕРЕСТАНОВКА
20 PRINT "ВВЕСТИ А, В"
30 INPUT А, В
40 C=A
50 A=B
60 B=C
70 PRINT "А="; А; "В="; В
80 END
```





---

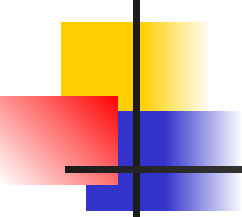
# ***Линейные программы с массивами***



# Массивы в языке Бейсик

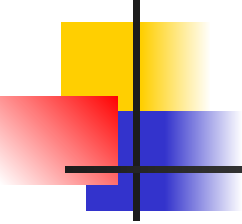
---

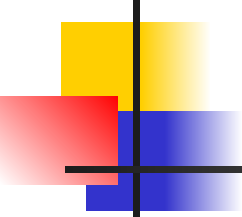
- Массивом называется упорядоченная совокупность однородных величин, обозначенных каждая одним и тем же именем с различными целочисленными индексами, изменяющимися по порядку.

- 
- 
- В Бейсике используются одно- и двумерные массивы (в QBASIC допустимы даже восьмимерные).
  - Они, как и простые переменные, могут быть различных типов: целые, вещественные, текстовые (строковые) и т.д.

# Общий вид элемента массива в Бейсике

- элемент одномерного массива:  $\langle \text{имя} \rangle (k)$ ;
  - элемент двумерного массива:  $\langle \text{имя} \rangle (i, j)$ ,
- где  $\langle \text{имя} \rangle$  — имя массива, должно отвечать тем же правилам, что и имя простой переменной;
- $k$  — индекс (номер) элемента одномерного массива,  $k \geq 0$ ;
- $i, j$  — индексы элемента двумерного массива (номера строки и столбца, на пересечении которых он находится),  $i > 0, j \geq 0$ .

- 
- 
- В QBASIC можно установить начальные значения  $k, i, j$  равными 1.
  - Индексы  $k, i, j$  могут быть представлены любыми арифметическими выражениями.

- 
- 
- При вычислении выражения, представляющего индекс, в QBASIC результат округляется до ближайшего целого.

# Общий вид оператора

## DIM



- в случае одномерного массива: DIM <имя>(d),
  - в случае двумерного массива: DIM <имя>(n, m),
- где DIM — имя оператора (от слова «dimension» — «размерность»);
- <имя> — имя массива;
- d, n, m — размеры массива, т.е. d — номер последнего элемента одномерного массива;
- n(m) — номер последней строки (последнего столбца) двумерного массива.

# Особенности записи оператора DIM

---

- в одном операторе DIM можно объявлять любое число массивов;
- оператор DIM рекомендуется помещать в начале программы;
- не следует использовать в программе простую переменную и массив с одним и тем же именем.





# Пример: оператор DIM D%(2), A(2,3), K\$(3) сообщает

---

- массив D% — одномерный целый, содержащий элементы D%(0), D%(1), D%(2);
- массив K — одномерный текстовый, включает элементы K\$(0), K\$(1), K\$(2), K\$(3);
- массив A — двумерный вещественный, включает такие элементы:  
A(0,0) A(0,1) A(0,2) A(0,3)  
A(1,0) A(1,1) A(1,2) A(1,3)  
A(2,0) A(2,1) A(2,2) A(2,3), т.е. содержат три строки и четыре столбца.



---

# ***Составление линейных программ с массивами***

# Особенности работы с массивами в программе



---

- Элементы массивов получают значения с помощью операторов ввода или присваивания как простые переменные.
- При вводе (выводе) массивов в операторах ввода (вывода) перечисляются имена всех вводимых (выводимых) элементов массива.

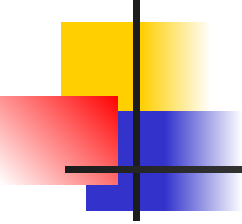
# Программа ввода и вывода массива P(1:3)

---

```
20 DIM P(3)
30 INPUT P(1),P(2),P(3)
40 PRINT P(1),P(2),P(3)
50 END
```

# Особенности работы с массивами в программе

- Все массивы можно разделить на два вида:
  - массивы постоянного размера [например,  $P(1:7)$ ,  $V(1:4)$ ,  $1:8$ ];
  - массивы переменного размера [например,  $A(1:k)$ ;  $C(1:m, 1:d)$ ].

- 
- 
- Переменные — размеры массивов должны быть определены до обращения к оператору DIM.

- Пример:

```
10 INPUT M
```

```
20 DIM X(M)
```