Программирование на алгоритмическом языке БЕЙСИК

НОРИЛЬСК
«МБОУ СОШ №3»
2009-2010 уч.год
Абрамишвили А.Д.

Алгоритмы (свойства, виды, стадий создания алгоритма

Виды алгоритмов:

1. Линейный алгоритм (описание действий, которые выполняются однократно в заданном порядке);
2. Циклический алгоритм (описание действий, которые должны повторятся указанное число раз или пока не выполнено задание);
3. Разветвляющий алгоритм (алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий)
4. Вспомогательный алгоритм (алгоритм, который можно использовать в других алгоритмах, указав только его имя).

Свойства алгоритмов:

1. Дискретность (алгоритм должен состоять из конкретных действий, следующих в определенном порядке);
2. Детерминированность (любое действие должно быть строго и недвусмысленно определено в каждом случае);
3. Конечность (каждое действие и алгоритм в целом должны иметь возможность завершения);
4. Массовость (один и тот же алгоритм можно использовать с разными исходными данными);
5. Результативность (отсутствие ошибок, алгоритм должен приводить к правильному результату для всех допустимых входных значениях).

Ведение в язык программирования

Q BASIC

Для представления алгоритма в виде, понятном компьютеру, служат *языки программирования*. Сначала разрабатывается алгоритм действий, а потом он записывается на одном из таких языков. В итоге получается текст программы - полное, законченное и детальное описание алгоритма на языке программирования. Затем этот текст программы специальными служебными приложениями, которые называются *трансляторами*, либо переводится в машинный код (язык нулей и единиц), либо исполняется.

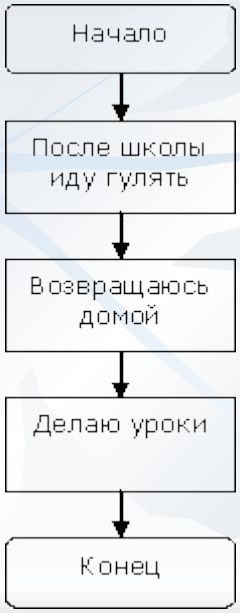
Языки программирования - искусственные языки. От естественных они отличаются ограниченным числом "слов", значение которых понятно транслятору, и очень строгими правилами записи команд (*операторов*).

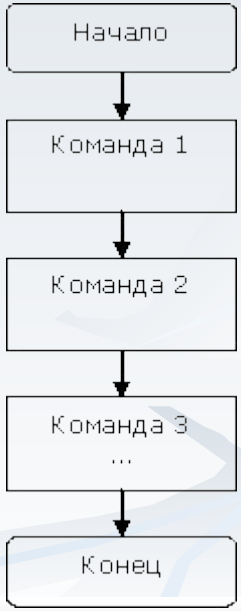
Для написания текста программы можно использовать обычный текстовый редактор (например, Блокнот), а затем с помощью компилятора перевести её в машинный код, т.е. получить исполняемую программу. Но проще и удобнее пользоваться специальными интегрированными средами программирования.

Basic (Бейсик) создавался в 60-х годах в качестве учебного языка и очень прост в изучении. По популярности занимает первое место в мире.

Линейная структура программы-

Программа имеет линейную структуру, если все операторы (команды) выполняются последовательно друг за другом.

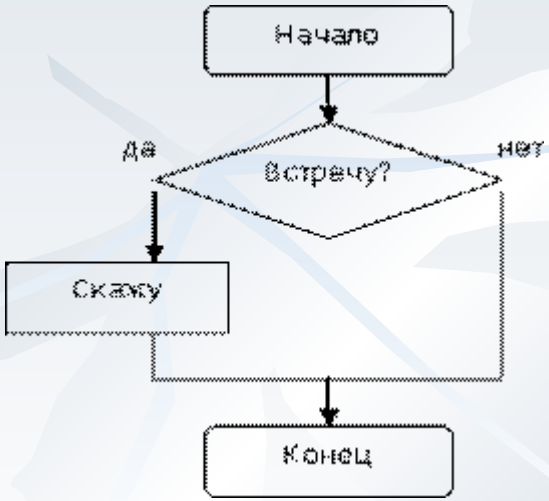


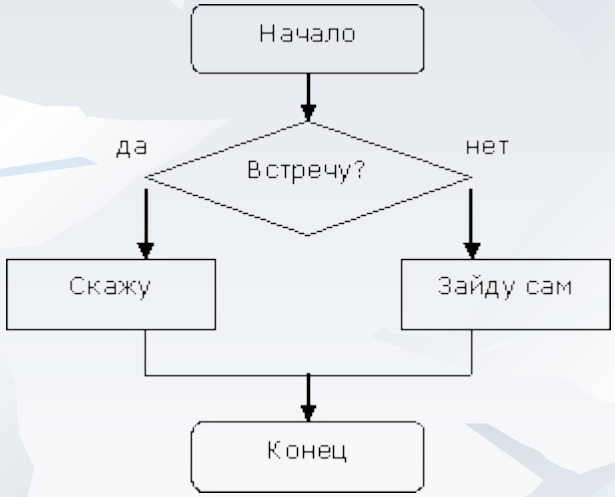


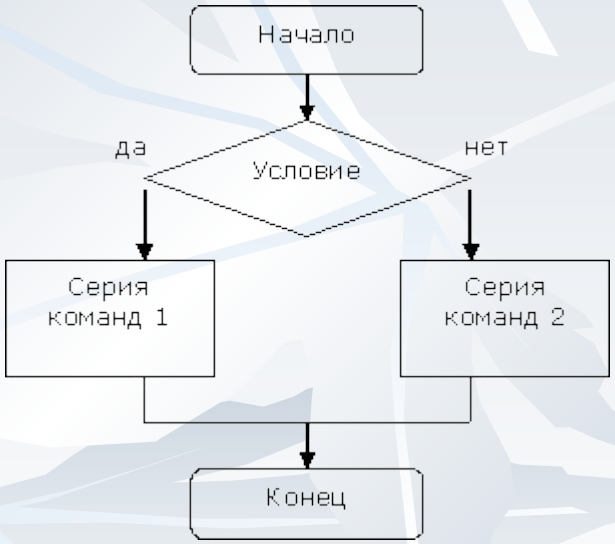
Ветвления алгоритмах и программах

Разветвляющий алгоритм – это алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий.

Во многих случаях требуется, чтобы при одних условиях выполнялась одна последовательность действий, а при других - другая.





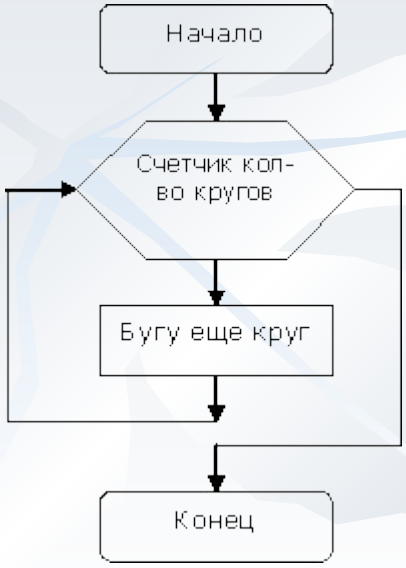


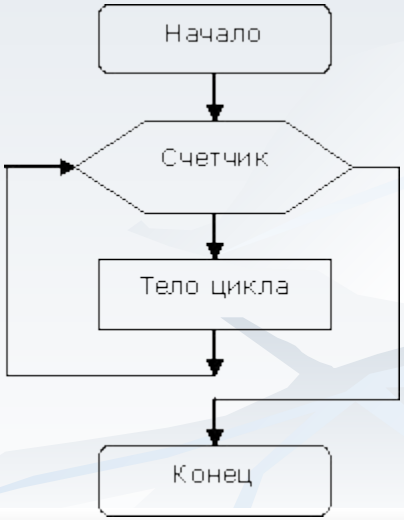
Циклы в алгоритмах

Циклический алгоритм - описание действий, которые должны повторяться указанное число раз или пока не выполнено заданное условие.

Например, на уроке физкультуры вы должны пробежать некоторое количество кругов вокруг стадиона.

Перечень повторяющихся действий называют телом цикла.





Такие циклы называются - циклы со счетчиком

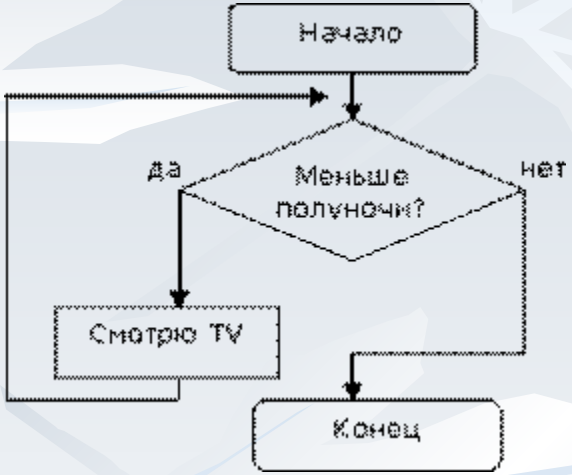
На языке Basic они записываются следующим образом:

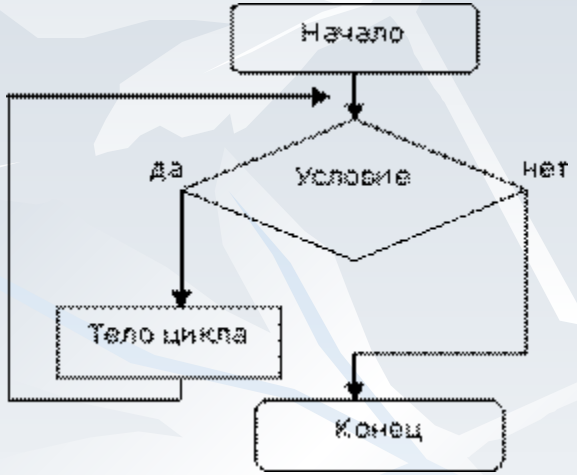
FOR Счетчик=НачЗнач TO КонЗнач [STEP шаг]
тело цикла
NEXT [Счетчик]

Пример: вычислить факториал числа a (записывается так: a!). Факториал - это произведение чисел от 1 до a. Например, 5! (факториал пяти) - это 5!=1*2*3*4*5

```
REM Вычислить факториал числа
a=5
f=1
FOR I=1 TO a
f=f*I
NEXT
PRINT f
END
```

В субботу вечером вы смотрите телевизор. Время от времени поглядываете на часы и если время меньше полуночи, то продолжаете смотреть телевизор, если это не так, то вы прекращаете просмотр телепередач.





Циклы такого вида называют - **циклы с предусловием**.

На языке Basic они записываются следующим образом:

DO WHILE условие

Тело цикла

LOOP

Массивы. Одномерные массивы-

Итак, что же такое массивы...

Массив, это разновидность переменной. Он дает возможность хранить сколько угодно значений под одним и тем же именем. К каждому конкретному значению массива, необходимо обращаться через числовой индекс.

Массив - это набор переменных, имеющих одинаковое имя (идентификатор), но различающихся порядковыми номерами (индексами).

Для того чтобы использовать массив его надо сначала объявить в программе. Для этого используют оператор **DIM**. По умолчанию (если нет оператора **DIM** в программе) считается заданным массив из 10 элементов.

Пример:

DIM a(100) AS INTEGER
Это массив из ста элементов, каждый из которых может быть целым числом.
DIM name(30) AS STRING
DIM mas(20)

Массивы. Двумерные массивы

Двумерные массивы можно представить себе как таблицы, в ячейках которых хранятся значения элементов массива, а индексы элементов массива являются номерами строк и столбцов.

Объявляются двумерные массивы так же, как переменные и одномерные массивы. Например, целочисленный числовой массив, содержащий 3 строк и 4 столбца объявляется следующим образом:

```
DIM tabl(3 ,4)
DIM tabl(3 ,4) AS INTEGER
```

Таблица умножения

```
REM  Таблица  умножения
DIM tabum (1 TO 9, 1 TO 9) AS INTEGER
REM Заполнение  массива - создание таблицы умножения
FOR I=1 TO 9
FOR J=1 TO 9
tabum(I, J)=I*J
NEXT J
NEXT I
REM Вывод массива на экран в виде таблицы
FOR I=1 TO 9
FOR J=1 TO 9
PRINT tabum(I,J);
NEXT J
PRINT
NEXT I
END
```

Символьные и строчные переменные

Очень часто в программах требуется использовать символьные или строчные переменные. Что же это такое? Это переменные, значениями которых являются либо алфавитно-цифровые символы, либо несколько таких символов.

Строки - последовательность алфавитно-цифровых символов.

Для того, чтобы использовать такие переменные в программе необходимо их соответствующим образом объявить. Для этого используется уже известный оператор **DIM**.

```
DIM s AS STRING
 s="Строка123"
```

Или добавлять справа от переменной символ **$**.

```
s$="Тоже строка 987"
```

Функции для работы со строками:

- LEN(s$) Вычисляет длину строки (количество символов).
- MID$(s$,n,k) Выделяет из строки s$ k символов начиная с n-го символа.
- VAL(s$) Преобразует числовую часть начала строки в число.
- STR$(x) Преобразует число в символьную форму.
- ASC(s$) Вычисляет десятичный код символа.
- CHR$(x) Преобразует код в символ.
- INKEY$ Функция опроса клавиш, нажатых на клавиатуре.

Пример: Эта программа выводит на экран две строки. Обратите внимание на два способа использования и объявления строковых переменных.

```
DIM stroka AS STRING
stroka="Один"
stroka2$="Два"
PRINT stroka
PRINT stroka2$
END
```

Строчные переменные можно склеивать и сравнивать друг с другом. Для склеивания строк (конкатенации) используют знак плюс (+).

Пример.

```
REM конкатенация строк
s1$="Привет! "
s2$="Меня зовут Саша."
s$=s1$+s2$
PRINT s$
END
```

Подпрограммы и процедуры

При создании средних по размеру программ используется ***структурное программирование,*** идея которого заключается в том, что структура программы должна отражать структуру решаемой задачи, чтобы алгоритм решения был ясно виден из исходного текста.

С этой целью в программирование введено понятие ***подпрограммы*** - ***набора операторов (команд), выполняющих нужное действие и не зависящих от других частей исходного кода.*** Программа разбивается на множество подпрограмм, каждая из которых выполняет какое-то действие, предусмотренное исходным заданием.

Подпрограммой называется группа операторов, к которой обращаются из основной программы несколько раз.

Очень важная характеристика подпрограмм - это возможность их повторного использования

Процедуры состоят из трех частей: заголовка, тела процедуры, завершения процедуры.

SUB имя (список параметров)
тело процедуры - список операторов
END SUB

SUB hello (s$)

Пример:

```
PRINT "Привет, ", s$,"! Как твои дела?"
END SUB
   REM приветствие
name1$="Саша"
name2$="Вася"
   REM процедуру можно вызвать так
CALL hello(name1$)
   REM а можно вызвать так
hello(name2$)
    REM или даже так
hello("Марина")
END
```

Подпрограммы и функции

Функции отличаются от процедур тем, что не только выполняют определенные действия, но еще и возвращают вызывающей программе какое-то значение.

Процедуры и функции бывают стандартными и нестандартными. Стандартные подпрограммы входят в библиотеку, которая поставляется вместе с системой программирования. Нестандартные процедуры и функции программисты пишут сами.

FUNCTION имя (список параметров)
тело функции - список операторов
END FUNCTION

Графический режим работы

Ну и теперь, наверное, самое интересное. Будем рисовать. Кто же не любит это занятие?!

Программы могут выводит данные на экран в текстовом и графическом режиме работы. Для перехода в графический режим работы служит оператор:

SCREEN <mode>

<mode> - целочисленная константа, указывающая режим работы для данного экрана и адаптера. Пример:

SCREEN 1
SCREEN 2
...
SCREEN 11
...

- CLS Очистка экрана
- PSET(X,Y),C Изобразить точку. X,Y - координаты точки, C -цвет.
- PSET STEP(X,Y),C Изобразить точку. X,Y - смешение от данной точки, C - цвет.
- LINE(X1,Y1)-(X2,Y2),C Прямая линия.X1,Y2 и X2,Y2- координаты концов линии, C - цвет.
- LINE -(X2,Y2),C Прямая линия. От текущего положения курсора до X2,Y2- координаты конца линии, C - цвет.
- LINE(X1,Y1)-(X2,Y2),C,B Прямоугольник. X1,Y2 и X2,Y2- координаты концов диагонали, C - цвет.
- LINE(X1,Y1)-(X2,Y2),C,BR Закрашенный прямоугольник. X1,Y2 и X2,Y2- координаты концов диагонали, C - цвет.
- CIRCLE(X,Y),R,C Окружность. X,Y - координаты центра, C -цвет.
- CIRCLE STEP(X,Y),R,C Окружность. X,Y - смешение от данной точки, C - цвет.
- CIRCLE(X,Y),R,C,A1,A2 Дуга окружности. X,Y - координаты центра, C -цвет, A1,A2 - угловые меры начальной и конечной точки дуги.
- CIRCLE(X,Y),R,C,,,K

- CIRCLE(X,Y),R,C,A1,A2,K Элипс. K - коэффициент сжатия.
- PAINT(X,Y),C1,C2 Закрасить область. C1 - цвет закраски, C2 - цвет границы.
- LOCATE T1,T2 Установка курсора в данную позицию. T1, T2 - номер строки и столбца. PRINT Оператор вывода текста

Создания движущихся изображений

Как нарисовать графический объект вам уже понятно. Но как заставить его двигаться?

Очень просто!

1. Рисуем объект цветом отличным от цвета фона.
2. Рисуем объект цветом фона.
3. Изменяем координаты.
4. Повторяем 1-3 столько раз сколько потребуется.

Пример: Движущийся круг.

```
REM Движущийся круг
  SCREEN 1
  x = 1
  y = 1
  REM цвет фона - 0(черный), цвет рисунка - 1

  FOR i = 1 TO 150

  REM Рисуем объект цветом отличным от цвета фона.
  c = 1
  CIRCLE (x, y), 2, c



  REM задержка
  FOR j = 1 TO 250000
  NEXT j

  REM Рисуем объект цветом цветом фона.
  c = 0
  CIRCLE (x, y), 2, c

  REM Изменяем координаты
  x = x + 2
  y = y + 1

  NEXT i
  END
```

Работа с файлами

Файлы широко применяются для решения различных задач. В них размещаются данные, предназначенные для длительного хранения. Каждому файлу присваивается уникальное имя, которое используется для обращения к нему. Использование файлов освобождает разработчика от хранения требуемых данных в тексте программы или многократном вводе их с клавиатуры, что само по себе весьма утомительно и приводит к появлению различных ошибок в программах. Гораздо удобнее ввести эту информацию один раз и сохранить ее в файле на диске.

Basic предлагает три различных способа сохранения и востребования информации с диска: последовательный, прямой и двоичный ввод/вывод файла. У каждого есть свои преимущества и недостатки.

Комбинированные типы

Описание комбинированного типа представляет собой список описаний его элементов; каждое описание похоже на описание простой переменной. Для примера, описание комбинированного типа PUPIL (ученик) может выглядеть следующим образом:

- фамилия, имя и отчество (строки);
- возраст (integer);
- пол (строка);
- класс (integer);
- буква класса (символ);
  и т.д.

```
TYPE Pupil
fio AS STRING * 20
age AS INTEGER


sex AS STRING * 6
class AS INTEGER
classname AS STRING * 1
END TYPE
```

спасибо за внимание