

Презентация

Язык программирования Basic

***Подготовил
ученик 11-А класса ОШ I-III ступеней № 15
г.Константиновки***

Волков Юрий

Оглавления

1. Введение
2. Виды алгоритмов
3. Линейный алгоритм
5. Безусловный переход
6. Условный переход
7. Оператор выбора
8. Оператор с заранее известным числом повторений
9. Запуск русского бейсика и начало работы
10. Алфавит языка
11. Переменная и что в ней меняется
12. Этапы решения алгоритмических задач
13. Примеры задач
14. Данные о авторе
15. Список использованной литературы





Basic принадлежит к языкам высокого уровня. Он служит своего рода переводчиком между человеком и компьютером, помогая им понять друг друга и совместно добиться решения поставленных задач. Разработан первый Бейсик в 1964 г. сотрудниками Дартмутского колледжа Дж. Кемени и Т. Курцем. Интересно происхождение названия языка. В прошлом веке один английский миссионер выделил из английского языка триста наиболее употребительных слов, назвал их Basic English и стал обучать туземцев. Опыт оказался весьма успешным, и контакты с аборигенами значительно упростились. Создатели языка Бейсик стремились достигнуть того же эффекта — облегчить понимание между "туземцами" — начинающими программистами, и компьютерами. Аббревиатура BASIC так и расшифровывается — "Beginner's All purpose Symbolic Instruction Code", что в переводе значит "многоцелевой язык символических команд для начинающих".

Идея оказалась удачной, и на десятилетия язык Бейсик стал основным в деле вовлечения в программирование новых и новых adeptов. В мире множество успешных людей, успешно работающих у нас и за рубежом, но начинавших с того же Бейсика — тогда еще для ZX-Spectrum. Даже в те времена на

Виды алгоритмов

Алгоритмы бывают трех основных видов, которые являются базовыми при написании программ.

Первый тип — **линейный алгоритм**. В нем все действия выполняются в строгом порядке, последовательно, одно за другим. Типичный жизненный пример такого алгоритма — рецепт пирога.

Второй тип — **разветвляющийся алгоритм**. Здесь те или иные действия выполняются в зависимости от выполнения или невыполнения некоего условия. Пример из жизни — правило перехода улицы по светофору. Если горит красный — стоим, если горит зеленый — идем.

Наконец, третий тип — **циклический алгоритм**. Он содержит повторяющиеся действия с какой-либо изменяющейся величиной, так называемым параметром. По циклическому алгоритму можно колоть дрова. Берем полено, ставим на попа, колем топором, берем второе полено и т. д., пока поленья не закончатся, и эта работа нам не надоест



Линейный алгоритм

Для написания программ с линейным алгоритмом мы уже готовы. Нам известно, как очистить экран и поприветствовать пользователя по имени, запросить у него необходимые данные и решить вычислительные задачи. Попробуем поупражняться.

Следите все же за интерфейсом своих программ!

В следующих заданиях вам предлагается написать программы, производящие требуемые вычисления и выдающие на экран результаты.

76. Расстояние до ближайшей к Земле звезды Альфа Центавра 4,3 световых года. Скорость света принять 300 000 км/с. Скорость земного звездолета 100 км/с. За сколько лет звездолет долетит до звезды?
77. Вычислить количество прожитых составителем программы дней. Учесть, что в високосном году 366 дней.
78. Известна теория биоритмов. С момента рождения жизнь человека подчиняется трем синусоидальным биоритмам. Физический цикл — 23 дня, эмоциональный — 28 дней и интеллектуальный — 33 дня. Первая половина каждого цикла — положительная, вторая — отрицательная. При переходе от положительной к отрицательной фазе в каждом цикле есть так



Разветвляющийся алгоритм

Ну что ж, надеюсь, что с линейными программами особых трудностей у вас не возникло. Пиши команды подряд, хотя, конечно, все сначала надо было продумать, и изображения у нас получались просто великолепные! *

Однако надо двигаться вперед. А дальше у нас — разветвляющийся алгоритм. Что это такое, мы уже говорили. Например, витязь на перепутье: "Если налево пойдешь, коня потеряешь, если прямо пойдешь — смерть найдешь", ну и т. д. А вот как он реализуется на практике в Бейсике, мы поговорим сейчас.

Ветвление в алгоритме и программе осуществляется двумя способами:

- на основе безусловного перехода;
- на основе условного перехода. Расскажем о первом



Безусловный переход

Безусловный переход оператором `goto` предписывает программе свернуть с линейного пути и, беспрекословно повинаясь, перейти к метке, расположенной в любом месте программы.

В качестве метки используются натуральные числа с двоеточием после них. Метка указывается только в начале строки (т. е. если в строке программы несколько операторов, то нельзя ставить метку, например, перед вторым). Приведем пример программы, рисующей на экране три символа звездочки по диагонали:

```
CLS
?"?*" ?"??*"
```

Символ `?` в данном случае означает пробел.

Добавим теперь метку `1` рядом со второй строкой и оператор `goto` в конце программы. Получаем:

```
CLS
1: ?"*"
?"?*"
9 "77* "
GOTO 1
```

Казалось бы, изменения незначительные, но запустите программу и посмотрите что происходит! Вы в панике? Вы



Условный переход

Мало того, что условный переход подразумевает выполнение тех или иных команд в зависимости от проверяемого условия, так Бейсик предоставляет нам еще и несколько вариантов действий.

Начнем с классической конструкции:

IF ... THEN ... ELSE

Или по-русски:

ЕСЛИ ... ТОГДА ... ИНАЧЕ

В записи условия можно использовать следующие символы:

= (равно);

> (больше);

< (меньше);

o или >< (не равно);

<= или =< (меньше или равно);

>= или => (больше или равно).

Мне кажется, что уже что-то должно быть понятно из самой конструкции условного оператора. Но раскроем его смысл на примере следующей программы. Допустим, в некоторой стране живут люди, говорящие только правду. Но при этом и у них есть вредные привычки, и некоторые из них курят. Для продажи



Оператор выбора

Для организации меню, т. е. выбора из нескольких возможностей, применяется оператор выбора, который позволяет, в зависимости от значения переменной, делать те или иные действия.

Конструкция оператора несколько тяжеловесна, но ничего страшного в ней нет.

Она выполняет один из нескольких блоков операторов в зависимости от значения выражения.

```
SELECT CASE тест_выражение CASE список_выражений1
        [блок_операторов_1]
        [CASE список_выражений2
        [блок операторов_2]]
        [CASE ELSE
        [блок_операторов_n]] END SELECT
```

Здесь `тест_выражение` — любое числовое или строковое выражение; `список_выражений1`, `список_выражений2` — ОДНО ИЛИ несколько выражений для сравнения с параметром `тест_выражение`. В выражении ключевое слово `is` должно стоять перед любым знаком отношения. `блок операторов 1`,



Оператор с заранее известным числом повторений

Когда мы работали с графикой, то в одном из заданий вам предлагалось построить мишень — пять концентрических окружностей. Мы это легко сделали, пять раз написав оператор построения окружности `circle`, изменяя в нем каждый раз радиус. Ну а если бы таких окружностей надо было бы 100? Неужели 100 раз писать практически одно и то же? Очень нелегко, даже если пользоваться копированием строк. Неужели нельзя как-нибудь полегче? Конечно можно! Нам на помощь приходит оператор ЦИКЛА `FOR. . .NEXT`.

Итак, правила пользования оператором цикла `for. . .next`:
Рассмотреть повторяющиеся действия и выделить в них равномерно изменяющуюся величину (параметр).

Дать параметру имя.

Определить для параметра начальное значение, конечное значение и шаг изменения, т. е. насколько за один раз увеличивается (или уменьшается) параметр.

Написать оператор цикла, состоящий из трех частей:
заголовок цикла

`FOR параметр=начальное значение TO конечное значение STEP шаг`



Запуск русского бейсика и начало работы.

Существует три основных варианта. Если у вас на компьютере Ничего, кроме MS DOS, нет, то придется открыть каталог с бейсиком, а затем в командной строке набрать имя запускающего файла qbasic и нажать клавишу <enter>. Если у вас есть операционная оболочка norton (или volkov) commander, то задача упрощается — переходите на панель, где содержится каталог с бейсиком, открываете ее клавишей <enter> или двойным щелчком левой кнопкой мыши, затем при помощи стрелок управления курсором находите файл qbasic.Exe, и запускаете его нажатием клавиши <enter> или двойным щелчком левой кнопкой мыши. Самый модный вариант, если у вас MS windows: найдите на рабочем столе ярлык quickbasic и дважды щелкните по нему левой кнопкой мыши.

Во всех трех случаях экран очищается и появляется среда русского бейсика (рис. 1.1).





Далее следуйте инструкции, появившейся на экране. При нажатии клавиши <Enter> вы попадаете в Руководство для начинающих, а, нажав клавишу <Esc> — в окно редактора (рис. 1.2).

Компьютер теперь готов к вводу и редактированию ваших программ.

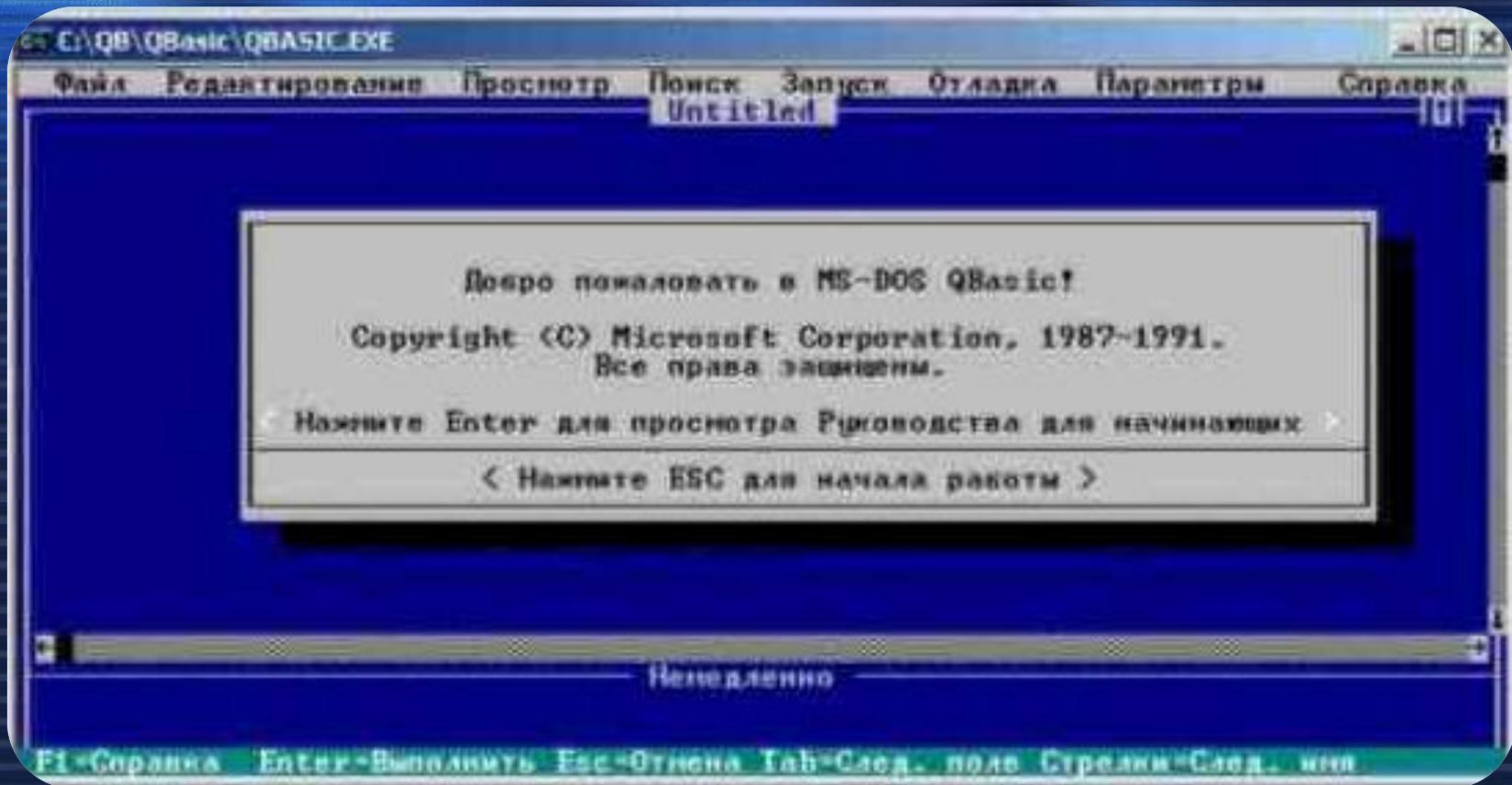
Вам для работы предоставляется два окна:

верхнее — окно редактирования;

нижнее — окно немедленного выполнения

Рис. 1.2. Окно редактора





F1=Справка Enter=Выполнить Esc=Отмена Tab=След. поле Стрелки=След. или

Неопределенно

Переход из одного окна в другое осуществляется клавишей <F6> или щелчком в соответствующем окне левой кнопкой мыши.

В окне немедленного выполнения вы можете сразу видеть результаты работы набранных команд после нажатия клавиши <Enter>.

Как вводить текст программы в окне редактора

В окне редактора мигает курсор — горизонтальная светящаяся черточка, указывающая, где будет вводиться текст программы при наборе его с клавиатуры.

Если вы заметили, что что-то набрали неправильно, есть несколько путей исправить допущенные ошибки.

Удаление лишних символов. Слева от курсора — клавиша <Backspace>. Символ, под которым мигает курсор и справа от курсора — клавиша <Delete>.

Вставка недостающих символов. Курсор установить под тот символ, слева от которого необходимо произвести вставку и набрать нужные символы. Текст при этом раздвинется автоматически.

Переход из режима вставки в режим замены. Режим вставки включается автоматически. В режим замены и обратно мы переходим нажатием клавиши <Insert>. Курсор при этом



Алфавит языка

В любом учебнике иностранного языка вначале дается его алфавит, т. е. набор символов для записи слов, предложений и всевозможных понятий этого языка. У языка Бейсик тоже есть алфавит, который содержит в себе следующие символы:
Заглавные (или прописные) буквы латинского алфавита: A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z. При наборе программы, впрочем, нет нужды следить за тем, чтобы буквы были заглавными. Интерпретатор сам изменит строчные буквы на заглавные.

Арабские цифры: 1, 2, 3, 4, 5, 6, 7, 8, 9, 0.

Разделители: , (запятая), ; (точка с запятой), . (точка), : (двоеточие), ' (апостроф), " (кавычки), ((открывающая скобка),) (закрывающая скобка), символ <Пробел>.

Знаки арифметических операций: + (сложение), — (вычитание), * (умножение), / (деление), л (возведение в степень).

Знаки операций отношений: > (больше), < (меньше), = (равно), <> (не равно), >= (больше либо равно), <= (меньше либо равно).

Переменная и что в ней меняется

Компьютер, как и вы, уважаемые читатели, обладает памятью. Она бывает разная. В процессе отладки вашей программы компьютер напрягает эту самую память, размещая в ней исходные данные, обрабатывая их, используя ваш алгоритм, получая результаты и доводя их до вашего сведения — чаще на экран монитора. Я думаю, для вас не секрет, что память эта называется ОЗУ (Оперативное Запоминающее Устройство), или по-английски RAM (Random Access Memory). Это, собственно, одно из основных устройств компьютера, имеющее, правда, ограниченный объем, измеряющийся в Мегабайтах.

Если вы пишете достаточно сложную и уже нелинейную программу, то наверняка потребуются переменные, т. е. такие области этой самой оперативной памяти, которые имеют имя, данное нами, и значения, которые могут меняться. Имя переменной в ходе выполнения программы постоянно, а значение может меняться многократно. Этот процесс можно сравнить со сдаваемой на лето дачей. Дача — это участок (область памяти компьютера), имеющий уникальный и неповторимый адрес, по которому его можно найти (имя переменной) и который не меняется, и каждое лето на дачу



Каковы правила на этот счет в Бейсик? Так как Russian Quick-Basic — это язык, рассчитанный на использование из-под DOS (Disk Operation System, дисковая операционная система), то существуют ограничения на имена переменных:

имя переменной должно состоять не более чем из сорока символов;

в качестве символов можно использовать только латинские буквы, цифры;

имя переменной не может начинаться с цифры;

категорически запрещены в именах файлов символы точки, запятой, звездочки, вопросительного знака, пробела.

Примеры правильных имен переменных:

X, Y, Z, IVAN;

IVAN3, SI, T234, LOVE7, R6N8F43;

NM, MAX, GAVGAV.

Примеры неправильных имен переменных:

(использована русская буква);

ИВАН (использована кириллица);

YOU+ME (использован недопустимый символ "+");

23DROVA (имя переменной начинается с цифры).

Переменные различаются по типу хранимой в них информации.



Итак, этапы решения алгоритмических задач на компьютере.

Постановка задачи. Один из самых главных этапов. Вы должны добиться от того, кто дает вам задачу (это можете быть и вы сами) ясной и четкой ее постановки. Вы однозначно и вполне определенно должны понять, что будет результатом решения задачи. Каковы исходные данные? Существуют ли ограничения для этих данных? Можно сказать, что точность и четкость в постановке задачи — это половина дела. Напротив, в случае недопонимания каких-то моментов вероятность непроизводительной траты времени и отрицательного результата резко возрастает.

Следующий этап — решение вопроса "Как будет реализовываться поставленная задача?". Как достичь требуемых результатов? Каковы способы и методы достижения уясненных на первом этапе целей?

После первых двух этапов наступает пора еще одного очень важного момента — этапа разработки алгоритма решения поставленной задачи т е структуризация разбиение задачи на



Примеры задач

Задание:

Задано количество хоккеистов N и для каждого указаны количество забитых шайб и штрафное время, заработанное в течение матча. Необходимо определить, какое количество шайб забил самый корректный игрок, т.е. игрок с минимальным штрафным временем.

Решение:

Исходный код:

```
10 CLS
20 INPUT "ВВЕДИТЕ КОЛИЧЕСТВО ХОККЕИСТОВ";N
30 PRINT "ВВЕДИТЕ КОЛИЧЕСТВО ШАЙБ, ЗАБИТЫХ КАЖДЫМ
      ХОККЕИСТОМ"
40 DIM SHAIB(N)
50 FOR I=1 TO N
60 LOCATE 4,I*8
70 INPUT SHAIB(I-1)
80 NEXT I
```



Данные о авторе

Я Волков Юрий Олегович. Сейчас я ученик школы № 15 города Константиновки, учусь в 11-А классе. Нынешний 2008 год и следующий 2009 год очень сложен для всех учащихся 11-х классов по всей Украине. Всех заботит будущее обучение после окончания школы. И я – один из них. В школьных стенах я получаю базовые знания, и качество этих знаний зависит только от меня. У меня есть склонности к изучению истории, моему любимому предмету, но жизнь с ним связывать я не собираюсь. Ныне я серьёзно занимаюсь математикой и украинским языком, как предметами, необходимыми для поступления в интересующий меня ВУЗ. Мир моих увлечений велик, но я расскажу о некоторых из них. Это изучение инновационных технологий, занятие лёгкой атлетикой, чтение исторических и фантастических книг. Я буду стараться через 5 – 10 лет иметь престижную, хорошо оплачиваемую и желательную любимую работу, и, безусловно, хороший автомобиль. Чтобы учиться на интересующем меня профиле я обладаю желанием учиться и дополнительными знаниями в области украинского языка и математики, получаемыми за стенами школы. В 2009 году в мои планы входит хорошее написание внешнего тестирования и



Список использованной литературы

Раздел: Программирование, Базы данных, Программное обеспечение

Ф.И.О.: Абверов Михаил Николаевич

контактные сведения: MikKirin@rambler.ru

наименование: Учебник по языку Basic в задачах и примерах

вид работы: Учебник

пожелания: Удачи!





Конец

