**Информатика, ПД-2**

*Законспектировать материал*

**Введение в язык программирования 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.

Алгоритмический язык Бейсик используется преимущественно в режиме диалога человека и ЭВМ. Этот язык ориентирован на решение различных задач вычислительного и не вычислительного характера. Название языка возникло от сокращения английских слов (Beginner’s Allpurpose Symbolic Instuction Code), что в переводе означает “многоцелевой язык символических инструкций для начинающих”.

**Алфавит языка (основные символы языка).**

В языке Бейсик используют следующие основные символы:

**а) 26 букв заглавного латинского алфавита A,B,C,...,X,Y,Z.**

**(допустимы также строчные).**

**б) цифры 0,1,...,9**

**в) Специальные символы:**

**1) знаки арифметических операций +,-,\*,/,\,^.**

**2) знаки отношений <, <=, =, >, >=, <>**

**3) ) ( . : ; , % ! # $ ? “**

Язык Бейсик использует ряд слов для обозначения команд, операторов, имен функций. Изменить значение этих слов пользователь не может, поэтому эти слова называются зарезервированными (служебными, ключевыми). К ним относятся:

**1) имена команд: NEW, RUN, LIST...**

**2) имена функций: SIN, COS, ATN,...**

**3) имена операторов: READ, INPUT, IF...**

**Основные элементы языка**.

К основным элементам языка относятся константы, переменные и специальные функции.

**Константы**.

**Константы представляют собой объекты, чьи значения не могут быть изменены в процессе выполнения программы.** В языке Бейсик различают числовые и символьные константы. Числовые константы делятся на целые и действительные.

1) **Целая константа** имеет вид обычных десятичных чисел со знаком или без него. Целый тип константы указывает знак %. Целые константы могут принимать значения в диапазоне от -32768% до 32768%.

2) **Вещественная константа** может быть записана в двух формах:

а) нормальный 3.21

б) экспотенциальный 1.2Е3 = 1200

Пример. Определить правильно или неправильно записаны константы: 37.25Е28, 1.5Е2, 3Е, Е5, 1Е2.5, 34Е 24, 33%, 4.2Е2%, 3Е44, 44.

3) **Символьные** (строковые или текстовые) константы представляют собой последовательность символов (цепочку литер) заключенную в кавычки “ ПРИВЕТ”. Максимальная длина текстовой константы 255 символов. В символьных константах могут использоваться любые символы алфавита, кроме кавычек.

**Переменные.**

**В отличие от констант переменные представляют такие объекты, значение которых могут изменяться в процессе выполнения программы**. Каждая переменная снабжается уникальным именем - **идентификатором,** состоящим из одних букв либо из букв и цифр. Имя переменной в конце может иметь знак, определяющий тип переменной:

% - целая переменная;

$ - символьная переменная.

Имя переменной может содержать от одного до 40 алфавитно-цифровых символов. В идентификаторах должны использоваться только буквы латинского алфавита и начинаться обязательно с буквы.

**Стандартные и нестандартные функции**.

**Функцию можно представить себе как некоторую вычислительную процедуру, на вход которой подается набор допустимых аргументов, а на выходе получается единственный результат - значение числового или символьного типа**. Любая функция реализуется с помощью программы, которая либо является постоянной компонентой системы программирования, либо должна быть сконструирована в той или иной форме пользователя. В соответствии с этим все функции подразделяются на системные (стандартные) и пользовательские (нестандартные). Подобно переменным каждая функция снабжается уникальным именем, которое образуется по тем же правилам, что и имя переменной. За стандартными функциями закреплены фиксированные имена, которые нельзя употреблять в качестве идентификаторов других объектов. Аргумент функции всегда записывается непосредственно за названием функции и заключается в скобки.

**Стандартные математические функции**:

SIN(X) - sinx

COS(X) - cosx

TAN(X) - tgx

ATN(X) - arctgx

ABS(X) - ⏐x⏐

SQR(X) -√x

EXP(X) - ex

LOG(X) - lnx

INT(X) - нахождение ближайшего целого, не превосходящего х

FIX(X) - ближайшее целое с округлением

SGN(X)-взятие знака числа ( -1 при x<0 , 1 при x>0, 0 при x=0)

RND(X)- генерация случайных чисел от 0 до 1.

Аргумент тригонометрических функций должен задаваться в радианах. Результаты работы функции ATN(X) также выдается в радианах. Аргументы функции LOG(X) должны быть положительными.

MOD – нахождение остатка от деления.

**Арифметические выражения.**

Арифметическое выражение является аналогом обычной алгебраической формулы. Оно задает правило выполнения конечной последовательности арифметических действий, результатом которой является число. В арифметическое выражение могут входить константы, переменные и функции, соединенные между собой знаками арифметических операций и круглыми скобками.

В языке Бейсик различают следующие арифметические операции:

+ сложение;

- вычитание;

\* умножение;

/ деление;

\ целочисленное деление;

^возведение в степень.

Записать на Бейсике следующее выражение:

(X^2+SIN(X)^2)/(2.5\*SQR(X))

Правила записи арифметических выражений.

1. Арифметические выражения записываются линейно, т.е. в строчку.

$\frac{а}{-b}$ = A/-B

2. Два знака операций не должны стоять рядом:

3. Круглые скобки используются для указания очередности выполнения операции.

4. Если последовательность выполнения операций не полностью определены скобками, то действия выполняются в следующем порядке:

Приоритет операций.

1) Функции.

2) Возведение в степень.

3)Умножение и деление.

4) Деление нацело.

5) Сложение и вычитание.

5. Возведение в целочисленную степень выполняется путем последовательных умножений.

6. Если показатель степени вещественный, то основание Х должно быть положительным.

7. Операция извлечения корня любой степени кроме квадратного заменяется возведением в дробную степень $\sqrt[n]{x^{m}}$ = x^(m/n).

**Операторы**

**CLS – очистка экрана**.

**REM** – оператор комментария. Все что следует после этого оператора до конца строки игнорируется компилятором и предназначено исключительно для человека. Т.е. здесь можно писать что угодно. Удобно использовать комментарий в начале программы для указания её названия и назначения.

пример:

REM Это комментарий

можно и так:

' Это тоже комментарий

**1.Оператор присваивания.**

Программа на языке Бейсик представляет собой последовательность конкретных указаний ЭВМ: ввести информацию, вычислить значение выражений, напечатать результат и т.д.

Каждое конкретное указание называется оператором. Все операторы можно разделить на 2 группы: выполняемые и невыполняемые.

**Оператор присваивания имеет вид**

**[LET] V=A,** где V – переменная (простая или индексированная), А - арифметическое выражение.

**Оператор присваивания выполняется следующим образом: вычисляется значение арифметического выражения, полученный результат присваивается переменой, т.е. заносится в ячейку памяти ЭВМ, соответствующую переменной V.**

Пример: x = 3.4 z = x+y z = 8.9

 y=5.5 z=z+1 z=9.9

1. К моменту выполнения оператора присваивания все переменные, входящие в арифметическое выражение должны быть определены.

2. Тип переменной, стоящей слева от знака присваивания может не совпадать с типом выражения, стоящим справа от знака присваивания. В этом случае осуществляется автоматическое преобразование числового значения из одного формата в другой. Никаких округлений при этом не происходит.

**2. Оператор ввода.**

Операторы ввода-вывода информации предназначены для обмена информации между оперативной памятью ЭВМ и ее внешними устройствами.

Оператор ввода INPUT

При его использовании данные вводятся с клавиатуры дисплея во время выполнения программы.

**Общий вид: INPUT V1,V2, ...,**

где V1,V2,.. - список переменных, разделенных запятой.

При выполнении этого оператора ЭВМ делает паузу и на экране появляется ?, и ЭВМ ждет ввода данных. На клавиатуре набираются числа, подлежащие вводу в порядке, задаваемом списком ввода. Если чисел несколько, то они отделяются друг от друга запятой. После того, как ввод данных закончен необходимо нажать ввод.

В общем виде оператор INPUT записывается следующим образом:

INPUT[;][“text”][;] V1,V2,...

Выполнение оператора INPUT начинается с приостанова работы программы. В этот момент на экране дисплея появляется сообщения, заданное параметром “text”, вслед за которым в этой же строке или в начале следующей появляется знак “?”. Позиция, в которой появляется знак вопроса зависит от наличия или отсутствия символа “;” сразу после наименования оператора. Если символ “;” присутствует, то знак “?” появляется в той же строке, что и подсказка “text”. Если в качестве разделителя, стоящего за параметром “text” используется “,”, то выдача “?” отменяется.

**3. Оператор вывода.**

**Общий вид PRINT список вывода (на экран).**

Список вывода может содержать имена переменных, константы, арифметические выражения, тексты.

Тексты применяются для печати всевозможных заголовков и пояснений. Текст может содержать любые символы, включая буквы русского алфавита. Сам текст заключается в кавычки.

Если в списке вывода содержатся арифметические выражения, то они сначала вычисляются.

Для удобства восприятия информации строка экрана разбивается на 6 зон по 14 позиций первые пять и 10 позиций последняя (всего 80). Если разделителем в списке вывода является запятая, то вывод осуществляется по зонам. Если данные не помещаются в зоне, то берется следующая зона.

Появление запятой всегда указывает на переход в следующую зону. Если в качестве разделителя в списке вывода используется “;”, то вывод осуществляется в так называемом плотном формате, т.е. каждая следующая данная печатается после предыдущей через пробел. При последнем “,” или”;” переход на новую строку не осуществляется.

Простейшие примеры алгоритма с командами ввода-вывода.

3.1. Простой ввод-вывод

10 INPUT A,B

20 PRINT A,B

3.2. Ввод-вывод с комментариями

10 INPUT “Введите два числа”; А,В

20 PRINT “Вы задали А=“;А;”В=“;В

3.3. Простейшие вычисления формул:

3.4. Решение геометрических задач.

Задана сторона квадрата А. Найти площадь квадрата S.

10 INPUT “Чему равна сторона квадрата”;А

20 S=A\*A

30 PRINT “Площадь квадрата S=“; S

3.5. Решение физических задач.

Велосипедист едет со скоростью V в течении t времени. Какой путь он проедет.

10 INPUT “Чему равна скорость”;V

20 INPUT ”Какое время”; t

30 S=V\*t

40 PRINT “Путь S=“;S