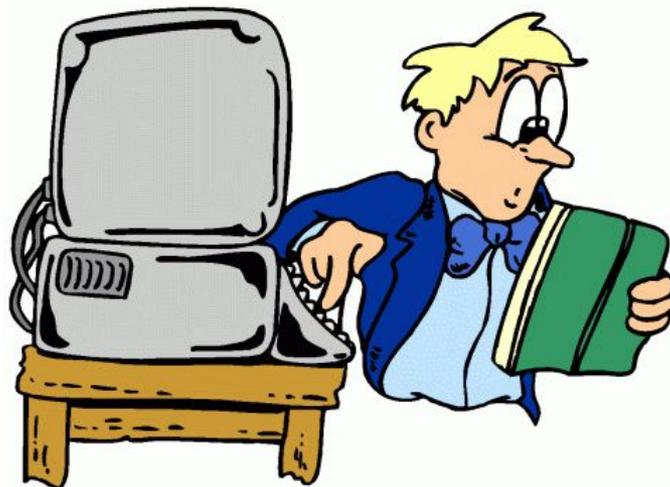




***Введение в язык
программирования.
Основные типы данных.
Выражения и функции***



Основные вопросы:

- Эволюция языков программирования
- Классификация языков программирования.
- Язык программирования Basic.
Компоненты алгоритмического языка
(алфавит, синтаксис, семантика)
- Величины. Константы и переменные.
Характеристики переменных.
- Запись арифметических выражений на языке программирования.

- Развитие вычислительной техники сопровождается созданием новых и совершенствованием существующих средств общения программистов с ЭВМ – *языков программирования (ЯП)*.
- К настоящему времени созданы десятки различных ЯП от самых примитивных до близких к естественному языку человека. Чтобы разобраться во всем многообразии ЯП, нужно знать их классификацию, а также историю создания, эволюцию и тенденции развития.

- Именно для удобного решения задач с помощью ЭВМ искусственно и создавались языки программирования. Естественным же языком, который «понимает» компьютер, является машинный.
-  *Машинный язык* – это такой язык, который компьютер воспринимает непосредственно, т.е. это язык машинных команд данной модели компьютера. А мы уже знаем, что ЭВМ «понимает» только язык двоичных знаков: нулей и единиц.
- Процесс описания конкретного алгоритма на языке машинных команд называется *программированием в машинных кодах*.

- Для выполнения этой работы программист должен знать коды всех машинных операций, назначение и особенности применения каждой из них, а также помнить адреса конкретных ячеек памяти, хранящие те или иные операции. Процесс разработки такой программы чрезвычайно трудоемок и непроизводителен. Поэтому для своего облегчения программисты придумали язык, переводящий символические имена в машинные коды. Такие программы, работающие с помощью мнемонических (символьных) обозначений, называют *ассемблерами*. Они и сейчас находят широкое применение, особенно при разработке эффективных, быстродействующих программ.
- Именно такие **машинные** и **полумашинные** языки программирования относятся к языкам **низкого уровня**. Их еще называют *машинно-ориентированными языками* (сюда относят и автокод). Практически любой класс ЭВМ имеет свой язык ассемблера. На сегодняшний день язык ассемблера используется для создания системных программ, использующих специфические аппаратные возможности данного класса ЭВМ.

Языки программирования низкого уровня — это языки программирования, ориентированные на команды процессора. Операторы языка низкого уровня представляют собой мнемокоды команд машинного кода.

Языки программирования высокого уровня — это языки программирования, ориентированные на человека (программиста).



Переводчик алгоритмов, записанных на языке высокого уровня, в машинный код конкретного процессора называется -

ТРАНСЛЯТОР

компиляторы

получают весь текст программы, потом просматривают его и создают машинный автономный код.

Языки программирования, ориентированные на трансляцию путем компиляции, называются **компилируемыми языками программирования**.

Паскаль, Си и многие другие

+ высокая скорость работы и автономность.

- трудоемкость отладки, большой размер полученного кода
Машино-зависимость.

интерпретаторы.

Он постепенно просматривает текст программы, последовательно переводит каждую строку в машинный код и тут же исполняет его инструкции.

QBASIC

простота написания и отладки
уменьшенный размер кода
машино-независимость

Обзор языков программирования

Ассемблер

- **К языкам низкого уровня относятся языки Ассемблера.** Свое название они получили от имени системной программы Ассемблер, которая преобразует исходные программы, написанные на таких языках, непосредственно в коды машинных команд. Частями здесь служат **операторы**, а результатом сборки последовательность машинных команд. Язык Ассемблера объединяет в себе достоинства языка машинных команд и некоторые черты языков высокого уровня.
- **Ассемблер** позволяет также гибко и полно использовать технические возможности компьютера, как и язык машинных команд. На **Ассемблере** можно написать столь же эффективную по размеру и времени выполнения программу, как и программу на языке машинных команд. Это достоинство отсутствует у языков высокого уровня. **Этот язык часто применяют для программирования систем реального времени, технологическими процессами и оборудованием, обеспечение работы информационно-измерительных комплексов.** К таким системам обычно предъявляются высокие требования по объему занимаемой машинной памяти. Часто язык **Ассемблера** дополняется средствами формирования макрокоманд, каждая из которых эквивалентна целой группе машинных команд. Такой язык называют языком макроассемблера.
- **Ассемблер** машинно-зависимый язык, т. е. он отражает особенности архитектуры конкретного типа компьютера

Обзор языков программирования

- Паскаль

Язык программирования Паскаль был разработан профессором кафедры вычислительной техники Швейцарского Федерального института технологии Николасом Виртом в 1968 году как альтернатива существующим и все усложняющимся языкам программирования, таким, как PL/1, Algol, Fortran. Интенсивное развитие Паскаля привело к появлению уже в 1973 году его стандарта в виде пересмотренного сообщения, а число трансляторов с этого языка в 1979 году перевалило за 80. В начале 80-х годов 20 века, Паскаль еще более упрочил свои позиции с появлением трансляторов MS-Pascal и Turbo-Pascal для ПЭВМ. С этого времени Паскаль становится одним из наиболее важных и широко используемых языков программирования.

Важнейшей особенностью Паскаля является воплощенная идея структурного программирования. Другой существенной особенностью является концепция структуры данных как одного из фундаментальных понятий.

Основные причины популярности Паскаля заключаются в следующем:

- простота языка позволяет быстро его освоить и создавать алгоритмически сложные программы
- наличие специальных методик создания трансляторов с Паскаля упростило их разработку и способствовало широкому распространению языка
- оптимизирующие свойства трансляторов с Паскаля позволяют создавать эффективные программы. Это послужило одной из причин использования Паскаля в качестве языка системного программирования
- в языке Паскаль реализуются идеи структурного программирования, что делает программу наглядной и дает хорошие возможности для разработки и отладки

Обзор языков программирования

Си

- Сотрудник фирмы Bell Labs Денис Ритчи создал язык Си в 1972 году во время совместной работы с Кеном Томпсоном, как инструментальное средство для реализации операционной системы Unix, однако популярность этого языка быстро переросла рамки конкретной операционной системы и конкретных задач системного программирования. В настоящее время любая инструментальная и операционная система не может считаться полной если в ее состав не входит компилятор языка Си. Ритчи не выдумывал Си просто из головы – прообразом служил язык Би разработанный Томпсоном.
- Си является орудием системного программиста и позволяет глубоко влезать в самые тонкие механизмы обработки информации на ЭВМ. Хотя язык требует от программиста высокой дисциплины, он не строг в формальных претензиях и допускает краткие формулировки.
- Си – **современный язык**. Он включает в себя те управляющие конструкции, которые рекомендованы теорией и практикой программирования
- Си – **мощный и гибкий язык**. Большая часть операционной системы Unix, компиляторы и интерпретаторы языков Фортран, Паскаль, Лисп, и Бейсик написаны именно с его помощью.
- Си – **удобный язык**. Он достаточно структурирован, чтобы поддерживать хороший стиль программирования и вместе с тем не связан жесткими ограничениями. В некотором смысле язык Си – **самый универсальный**, т.к. кроме набора средств, присущих современным языкам программирования высокого уровня (структурность, модульность, определенные типы данных), в него включены средства для программирования практически на уровне ассемблера. Большой набор операторов и средств требуют от программиста осторожности, аккуратности и хорошего знания языка со всеми его преимуществами и недостатками.

Обзор языков программирования

Си++

- Язык **С++** появился в начале 80-х годов 20 века. Созданный Бьерном Страуструпом с первоначальной целью избавить себя и своих друзей от программирования на ассемблере, **Си** или различных других языках высокого уровня.
- По мнению автора языка, различие между идеологией **Си** и **С++** заключается примерно в следующем: программа на **Си** отражает “способ мышления” процессора, а **С++** - способ мышления программиста. Отвечая требованиям современного программирования, **С++** делает акцент на разработке новых типов данных наиболее полно соответствующих концепциям выбранной области знаний и задачам приложения. Класс является ключевым понятием **С++**. Описание класса содержит описание данных, требующихся для представления объектов этого типа и набор операций для работы с подобными объектами.
- Язык **С++** является средством объектного программирования, новейшей методики проектирования и реализации программ, которая в текущем десятилетии, скорее всего, заменит традиционное процедурное программирование. Главной целью создателя языка доктора Бьерна Страустрапа было оснащение языка **С++** конструкциями, позволяющими увеличить производительность труда программистов и облегчить процесс овладения большими программными продуктами.

Обзор языков программирования

Кобол

- **Кобол** - это сравнительно старый язык, разработанный прежде всего для исследований в экономической сфере. Язык позволяет эффективно работать с большим количеством данных, он насыщен разнообразными возможностями поиска, сортировки и распределения. О программах на **Коболе**, основанных на широком использовании английского языка, говорят, что они понятны даже тем, кто не владеет **Коболом**, поскольку тексты на этом языке программирования не нуждаются в каких-либо специальных комментариях. Подобные программы принято называть самодокументирующимися. К числу других **плюсов Кобола** обычно относят его структурированность. Довольно мощные компиляторы с этого языка разработаны для персональных компьютеров. Некоторые из них столь эффективны, что программу, отлаженную на персональном компьютере, нетрудно перенести на большие ЭВМ.
- Перечисляя **минусы** нельзя не вспомнить о том, что на **Коболе** можно запрограммировать лишь простейшие алгебраические вычисления. Для инженерных расчетов этот язык не годится. Еще одна причина, которая в какой-то мере сдерживает развитие языка, - это наличие в США специально созданного отраслевого комитета, вырабатывающего стандарты, за соблюдением которых следит правительственная комиссия. Как это всегда бывает в подобных случаях, фирмы, занимающиеся разработкой программного обеспечения, не торопятся подгонять свои заготовки к жестким требованиям комиссии, отсутствует конкуренция версий, а в итоге проигрывает распространение языка

Обзор языков программирования

Пролог

- Язык логического программирования предназначен для представления и использования знаний о некоторой предметной области.
- Программы на этом языке состоят из некоторого множества отношений, а ее выполнение сводится к выводу нового отношения на основе заданных.
- В **Прологе** реализован декларативный подход, при котором достаточно описать задачу с помощью правил и утверждений относительно заданных объектов. Если это описание является достаточно точным, то ЭВМ может самостоятельно найти требуемое решение.

Обзор языков программирования

Бейсик

- Бейсик (Basic - Beginner's All-Purpose Symbolic Instruction Code – “универсальный символический код инструкций для начинающих”). Прямой потомок **Фортрана** и до сих пор самый популярный язык программирования для персональных компьютеров. Появился Бейсик в **1963 году** (назвать автора было бы трудно, но основная заслуга в его появлении несомненно принадлежит американцам Джону Кемени и Томасу Курцу). Как и любые преимущества, простота Бейсика оборачивалась, особенно в ранних версиях трудностями структурирования; кроме того, Бейсик не допускал рекурсию – интересный прием, позволяющий составлять эффективные и в то же время короткие программы.
- Разработаны мощные компиляторы Бейсика, которые обеспечивают не только богатую лексику и высокое быстродействие, но и возможность структурного программирования. По мнению некоторых программистов, наиболее интересными версиями являются **GWBASIC, Turbo-Basic и Quick Basic**.
- Безусловными лидерами среди различных версий Бейсика были **Quick Basic 4.5 и PDS 7.1** фирмы Microsoft, появившиеся в конце 80-х годов.

Языки программирования

- это совокупность набора символов (алфавит) системы, правил образования (синтаксис) и истолкование конструкций из символов (семантика) для задания алгоритмов с использованием символов естественного языка.***

Компоненты алгоритмического языка:

- *Алфавит* – фиксированный для данного языка набор основных символов, допускаемых для составления текста программы на этом языке.
- *Синтаксис* – система правил, определяющих допустимые конструкции языка программирования из букв алфавита.
- *Семантика* - система правил однозначного толкования отдельных языковых конструкций, позволяющих воспроизвести процесс обработки данных.



Алфавит языка Бейсик

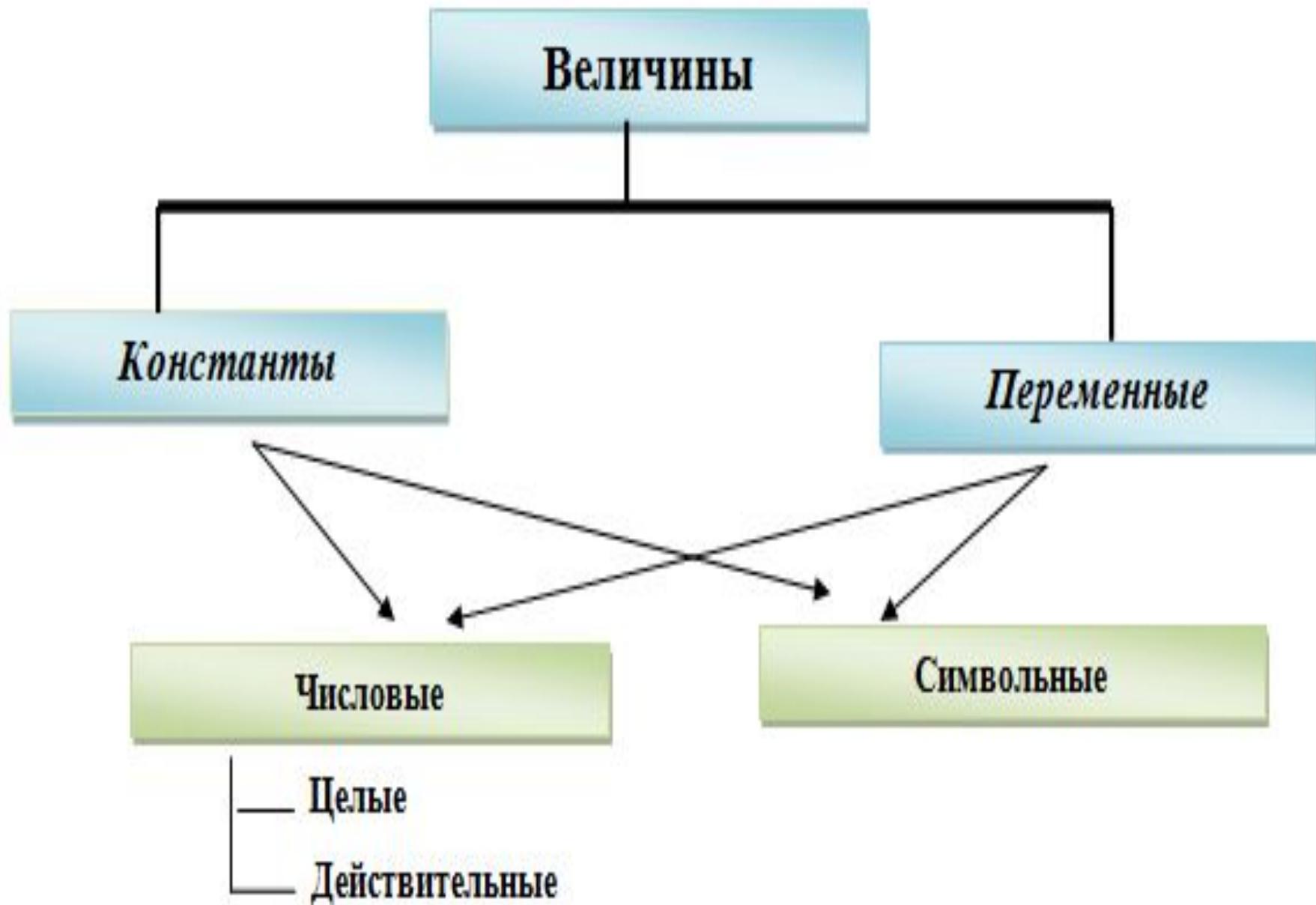
- **заглавные (прописные) буквы латинского алфавита** – А, В, С, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z;
- **арабские цифры** – 1,2,3,4,5,6,7,8,9,0;
- **разделители** – запятая (,), точка с запятой (;), точка (.), двоеточие (:), апостроф('), кавычки (“”), открывающаяся скобка (, закрывающаяся скобка) , символ Пробел;
- **знаки арифметических операций**
- **знаки операций отношений**

Арифметические операции на языке Basic.

Операция	Обозначение	Пример	Результат
<i>Сложение</i>	+	2+5	7
<i>Вычитание</i>	-	10-8	2
<i>Умножение</i>	*	3*4	12
<i>Деление</i>	/	15/3	5
		15/4	3.75
<i>Целочисленное деление</i>	\	15\4	3
<i>Возведение в степень</i>	^	2^3	8
<i>Остаток от деления</i>	MOD	13 MOD 5	3

Знаки операций отношения

Значение операции	Знак операции	Выражение в Бейсике
<i>Равенство</i>	=	$x=y$
<i>Неравенство</i>	$\langle \rangle$	$x\langle \rangle y$
<i>Меньше</i>	<	$x<y$
<i>Больше</i>	>	$x>y$
<i>Меньше или равно</i>	\leq	$x\leq y$
<i>Больше или равно</i>	\geq	$x\geq y$



Переменная - это величина, которая может меняться при выполнении программы.

Кроме указанных в таблице, целые числа записывают также в **экспоненциальной записи**, например,
 $2,55 \cdot 10^5 = 2.55E5$; $7,15 \cdot 10^{-7} = 7.15E-7$

Тип переменной	Символ	Описание	Объём	Пример
Целые числа	%	integer	2 байта	17; 123
Действит. числа	нет	real	4 байта	3.1415
Символьный	\$	string	min 1бай	язык
Пользовательский	нет	type		
Целый 2 точности	#	double	8 байт	1.2543786

Идентификаторы

- это произвольный набор символов, который может содержать от 1 до 40 символов, причём **первый символ должен быть латинской буквой**, а остальные – латинские буквы или цифры или символы типа @, #, % и т.д.

Пример:

A, Ds, SodRan, k1, n123, dlina!

Тип идентификатора в Бейсике опознаётся по последнему символу в имени переменной.

% - целое число;

& - длинное целое число;

! – вещественное число обычной точности;

- вещественное число двойной точности;

\$ - символьный тип.

- ***Значение переменной*** – это ***содержимое тех ячеек памяти, в которых хранится переменная.***
- ***Имя переменной*** – это «где хранится», а значение переменной – «что хранится».
- **Присваивая новое значение переменной, мы теряем старое. Рекомендуется имена переменных давать по смыслу содержащихся в них значений.**

- ***Константами*** называются величины, которые имеют постоянные значения, не изменяющиеся во время выполнения операций.
- В Basic существует два типа констант – ***числовые*** и ***символьные*** (или литерные).

- ***Символьные (литерные) константы*** – последовательность любых отображений символов, заключается в кавычках. В цепочке символов могут быть любые символы языка (запятая, точки, пробелы и т.п.), кроме кавычек.

Примеры:

«Введите два разных числа»

«10.03.1994год»

- Допускается случай, когда ***литерная константа не содержит ни одного символа***; соответствующий текст (пусто) обозначается двумя кавычками, следующими друг за другом: **“ ”**.

Арифметические выражения

*- представляют собой
совокупность констант,
переменных, стандартных
функций, соединенных знаками
арифметических операций и
круглыми скобками.*



АРИФМЕТИКА В БЕЙСИКЕ

Элементарные арифметические выражения необходимо представить в виде, понятном для компьютера:

1. выражение должно быть записано в одну строку
2. для записи арифметических действий использовать знаки: + - * / ^ ()
3. недопустим пропуск знака умножения между *коэффициентом и переменной* (нельзя $2x$, надо $2*X$);
4. дробная часть отделяется от целой точкой, а не запятой (нельзя $3,14$, надо 3.14);
5. допустимо опускать в записи десятичной дроби ноль, стоящий перед точкой (вместо 0.123 можно $.123$);
6. необходимо соблюдать приоритет выполнения действий –
 - действия в скобках
 - вычисление функций
 - возведение в степень
 - умножение и деление
 - сложение и вычитание
 - действия одинаковой очередности выполняются слева направо.

Математические функции на языке Basic.

Название функции	Запись в математике	Запись в Бейсике	Пример в Бейсике
Абсолютная величина	$ x $	ABS(X)	ABS(-5)
Экспонента	e^x	EXP(X)	EXP(5)
Логарифм натуральн.	$\ln x$	LOG(X)	LOG(3)
Остаток целочисленного деления (MOD)	-	x MOD y	19 MOD 6.7
Квадратный корень	\sqrt{x}	SQR(X)	SQR(4)
Синус	$\sin x$	SIN(X)	SIN(3.14)
Косинус	$\cos x$	COS(X)	COS(1)
Тангенс	$\operatorname{tg} x$	TAN(X)	TAN(2)
Арктангенс	$\operatorname{arctg} x$	ATN(X)	ATN(3)
Знак числового выражения	-	SGN(X)	SGN(-5)

Правила записи функций

- Стандартные функции могут входить в **арифметические выражения**. В качестве аргументов также можно использовать арифметические выражения. Аргумент обязательно записывается в скобках **SIN (3), ABS (X)**

Примеры правильной записи стандартных функций

Алгебра	Бейсик
$\sin 3 + \cos 6 + \operatorname{tg} 4$	<code>SIN(3) +COS(6) +TAN(4)</code>
$\sqrt{ \sin 45^{\circ} - \cos^2 30^{\circ} }$	<code>SQR(ABS(SIN(45*3.14/180))-COS(30*3.14/180)^2)</code>

Тригонометрические функции

- ◆ 30° записывается как $30*3.14/180$ так как $1^{\circ} = \pi/180$
- ◆ Таким образом $\operatorname{SIN} 30^{\circ}$ запишется как `SIN (30*3.14/180)`

Домашнее задание



1. Используя материал презентации написать опорный конспект по данной теме.
2. Выполнить **практическую работу**.
3. Выполненные задания отправить преподавателю.

Практическая работа

Тема: «Запись стандартных функций»

Цель работы: Изучить правила записи стандартных функций и арифметических выражений. Научиться записывать арифметические выражения на языке Бейсик.



Ход работы:

При решении многих прикладных и математических задач на ЭВМ часто возникает необходимость, проведения вычислений при помощи выражений.

Выражение представляет собой запись, указывающую, какие операции следует произвести над данными, чтобы получить требуемое значение.

Арифметическое выражение - это символьная запись, составленная из чисел, имен переменных, встроенных функций, знаков арифметических операций, круглых скобок и имеющая смысл с точки зрения математики.

Арифметические операции:

Сложение	+
Вычитание	-
Умножение	*
Деление	/
Возведение в степень	^

Стандартные функции:

Название функции	Запись в математике	Запись в Бейсике	Пример в Бейсике
Абсолютная величина	$ x $	ABS(X)	ABS(-5)
Экспонента	e^x	EXP(X)	EXP(5)
Логарифм натуральный	$\ln x$	LOG(X)	LOG(3)
Остаток целочисленного деления (MOD)	-	x MOD y	19 MOD 6.7
Квадратный корень		SQR(X)	SQR(4)
Синус	$\sin x$	SIN(X)	SIN(3.14)
Косинус	$\cos x$	COS(X)	COS(1)
Тангенс	$\operatorname{tg} x$	TAN(X)	TAN(2)
Арктангенс	$\operatorname{arctg} x$	ATN(X)	ATN(3)
Знак числового выражения	-	SGN(X)	SGN(-5)

- $\log_a z = \frac{\ln Z}{\ln A} \Rightarrow \text{LOG}(Z)/\text{LOG}(A)$
- $\arcsin x = \operatorname{arctg} \frac{x}{\sqrt{1-x^2}} \Rightarrow \text{ARCSIN}(X) = \text{ATN}(X/\text{SQR}(1-X^2))$
- $\arccos x = \operatorname{arctg} \frac{\sqrt{1-x^2}}{x} \Rightarrow \text{ARCCOS}(X) = \text{ATN}(\text{SQR}(1-X^2)/X)$

При записи арифметических выражений необходимо придерживаться следующих правил и ограничений:

I. Все символы выражения записываются в одну строку. Многоэтажные выражения, верхние и нижние индексы запрещены.

II. Два знака арифметических операций не должны располагаться рядом. Знак умножения опускать нельзя.

III. Операции в арифметическом выражении выполняются в порядке старшинства, т.е.:

1. Операции внутри скобок;
2. Вычисление встроенных функций;
3. Возведение в степень;
4. Операции умножения и деления;
5. Операции целочисленного деления;
6. Операция деления по модулю;
7. Операция сложения и вычитания.

Операция равного старшинства выполняются по порядку слева направо.

Исключение: $A^C = A^(B^C)$.

ПРИМЕРЫ:

$$\frac{\sqrt[3]{3X - \text{COS}X}}{2.5 + |Y|^5} \rightarrow (3 * X - \text{COS}(X))^{(1/3)} / (2.5 + \text{ABS}(Y)^5)$$

$$\frac{\sqrt{5x^3 + 3.6}}{Y - 5} * A + D \rightarrow \text{SQR}(5 * X^3 + 3.6) / (Y - 5) * A + D$$



Задания для самостоятельной работы

Записать на языке Бейсик выражения. Задания выполнить письменно на листа и сдать преподавателю.

1) $\frac{\sqrt{15x - 15y}}{(5x + 1)^2}$	4) $\frac{3 + e^{y-1}}{1 + x^2 y - \text{tgz} }$
2) $\sin^2 x + \sqrt[5]{7y - e^3}$	5) $-\frac{3c - 5}{4 - c^2} + \frac{3 - 2c}{c^2 - 4}$
3) $\text{arctg } x^2 - \ln 2$	6) $(9x^{-3} - x^{-3}y^2) \left(\frac{1}{x}\right)^{-3}$

