

**Программирование в....**  
**Приборостроении**  
**Биотехнических системах**

Управление в сфере производства, на транспорте, в сфере обслуживания, медицине и др. областях сегодня перешло на новый уровень – вместо жесткого механического контроля повсеместно внедряется гибкое интеллектуальное управление, базирующиеся на микропроцессорной технике, математических моделях и программном обеспечении.

## Дисциплины:

- ✓ Информатика
- ✓ Информационные технологии
- ✓ Электроника и микропроцессорная техника
- ✓ Микропроцессорные устройства в ИИТ
- ✓ Компьютерные технологии в приборостроении

# Языки программирования

Язык программирования — формальная знаковая система, предназначенная для записи программ.

Программа обычно представляет собой некоторый алгоритм в форме, понятной для исполнителя (например, компьютера).

Язык программирования определяет набор лексических, синтаксических и семантических правил, используемых при составлении компьютерной программы. Он позволяет программисту точно определить то, на какие события будет реагировать компьютер, как будут храниться и передаваться данные, а также какие именно действия следует выполнять над этими данными при различных обстоятельствах.

# Языки программирования

- ▶ Низкого уровня
- ▶ Высокого уровня
  - ▶ Компилируемые
  - ▶ Интерпретируемые

# Языки низкого уровня

Язык программирования, близкий к программированию непосредственно в машинных кодах используемого процессора.

Разные типы процессоров имеют разные наборы команд. Если язык программирования ориентирован на конкретный тип процессора и учитывает его особенности, то он называется языком программирования низкого уровня.

Для обозначения машинных команд обычно применяется мнемоническое обозначение. Это позволяет запоминать команды не в виде последовательности двоичных нулей и единиц, а в виде осмысленных сокращений слов человеческого языка (обычно английских).

# Языки высокого уровня

Языки программирования высокого уровня значительно ближе и понятнее человеку, нежели компьютеру.

Основная черта высокоуровневых языков — это абстракция, то есть введение смысловых конструкций, кратко описывающих такие структуры данных и операции над ними, описания которых на машинном коде (или другом низкоуровневом языке программирования) очень длинны и сложны для понимания.

Особенности конкретных компьютерных архитектур в них не учитываются, поэтому создаваемые программы на уровне исходных текстов легко переносимы на другие платформы, для которых создан транслятор этого языка.

# Компилируемые и интерпретируемые языки

- ▶ Программа на компилируемом языке при помощи специальной программы компилятора преобразуется (компилируется) в набор инструкций для данного типа процессора (машинный код) и далее записывается в исполняемый файл, который может быть запущен на выполнение как отдельная программа. Другими словами, компилятор переводит программу с языка высокого уровня на низкоуровневый язык, понятный процессору.
- ▶ Интерпретатор непосредственно выполняет (интерпретирует) ее текст без предварительного перевода. При этом программа остается на исходном языке и не может быть запущена без интерпретатора. Можно сказать, что процессор компьютера — это интерпретатор машинного кода.



# Ассемблер

Языком самого низкого уровня является язык ассемблера, который просто представляет каждую команду машинного кода, но не в виде чисел, а с помощью символьных условных обозначений, называемых мнемониками.

Однозначное преобразование одной машинной инструкции в одну команду ассемблера называется транслитерацией. Так как наборы инструкций для каждого модели процессора отличаются, конкретной компьютерной архитектуре соответствует свой язык ассемблера, и написанная на нем программа может быть использована только в этой среде.

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.

# Обзор языков программирования высокого уровня

**Фортран** (FORTRAN — FORmula TRANslator — переводчик формул). Это первый компилируемый язык, созданный Джимом Бэкусом в 50-е годы. Программисты, разрабатывавшие программы исключительно на ассемблере, выражали серьезное сомнение в возможности появления высоко производительного языка высокого уровня, поэтому основным критерием при разработке компиляторов Фортрана являлась эффективность исполняемого кода.

Хотя в Фортране впервые был реализован ряд важнейших понятий программирования, удобство создания программ было принесено в жертву возможности получения эффективного машинного кода.

**COBOL** (Кобол). Это компилируемый язык для применения в экономической области и решения бизнес-задач, разработанный в начале 60-х годов. Язык позволяет эффективно работать с большим количеством данных, он насыщен разнообразными возможностями поиска, сортировки и распределения. Он отличается большой «многословностью» – его операторы иногда выглядят как обычные английские фразы. Подобные программы принято называть самодокументирующимися.

В Коболе были реализованы очень мощные средства работы с большими объемами данных, хранящимися на различных внешних носителях. На этом языке создано очень много приложений, которые активно эксплуатируются и сегодня. Достаточно сказать, что наибольшую зарплату в США получали программисты на Коболе.

- ▶ **Pascal** (Паскаль). Язык Паскаль, создан в конце 70-х годов основоположником множества идей современного программирования Никлаусом Виртом, как альтернатива существующим и все усложняющимся языкам программирования, таким, как PL/1, Algol, Fortran. В начале 80-х годов 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.

- ▶ **Бейсик** (BASIC - Beginner's All-Purpose Symbolic Instruction Code – “универсальный символический код инструкций для начинающих”). Для этого языка имеются и компиляторы, и интерпретаторы. Он создавался в 60-х годах в качестве учебного языка и очень прост в изучении.

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.

- ▶ **C** (Си) был создан в лаборатории Bell Labs Денисом Ритчи, как инструментальное средство для реализации операционной системы Unix, однако популярность этого языка быстро переросла рамки конкретной операционной системы и конкретных задач системного программирования.

Си является орудием системного программиста и позволяет глубоко влезать в самые тонкие механизмы обработки информации на ЭВМ.

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.

- ▶ **C++** (Си++). Это объектно-ориентированное расширение языка Си, созданное Бьярном Страуструпом в 1980 году. По мнению автора языка, различие между идеологией Си и C++ заключается в следующем: программа на Си отражает “способ мышления” процессора, а C++ - способ мышления программиста. Отвечая требованиям современного программирования, C++ делает акцент на разработке новых типов данных наиболее полно соответствующих концепциям выбранной области знаний и задачам приложения. Класс является ключевым понятием C++. Членами класса являются не только данные, но и функции. Функции – члены класса имеют привилегированный доступ к данным внутри объектов этого класса и обеспечивают интерфейс между этими объектами и остальной программой.

C++ программы компилируются компилятором C++ в двоичный формат, который затем исполняется непосредственно процессором; таким образом, выполнение программы осуществляется аппаратными средствами, на определенном процессоре и в определенной ОС.

- ▶ **C#** (Си Шарп). В конце 90-х годов в компании Microsoft был разработан язык C#. В нем воплотились лучшие идеи Си и Си++, а также достоинства Java. Правда, C#, как и другие технологии Microsoft, ориентирован на платформу Windows. Однако формально он не отличается от прочих универсальных языков, а корпорация даже планирует его стандартизацию.

Для C# предусмотрена компиляция программ на промежуточный язык IL (Intermediate Language). Код IL может запускаться на любой платформе при использовании среды .NET. С помощью JIT-компилятора он может быть преобразован в машинно-зависимый код для выполнения на платформе .NET



- ▶ **Java** (Джава, Ява). Этот язык был создан компанией Sun в начале 90-х годов на основе Си++. Он призван упростить разработку приложений на основе Си++ путем исключения из него всех низкоуровневых возможностей.

Но главная особенность этого языка – компиляция не в машинный код, а в платформенно-независимый байт-код (каждая команда занимает один байт) который непосредственно исполняется не процессором, а с помощью другого программного обеспечения, виртуальной машины Java (Java Virtual Machine, JVM). В свою очередь, JVM исполняется процессором. Таким образом, выполнение байт-кода Java-программ осуществляется не быстрыми аппаратными средствами, а с помощью более медленной программной эмуляции.

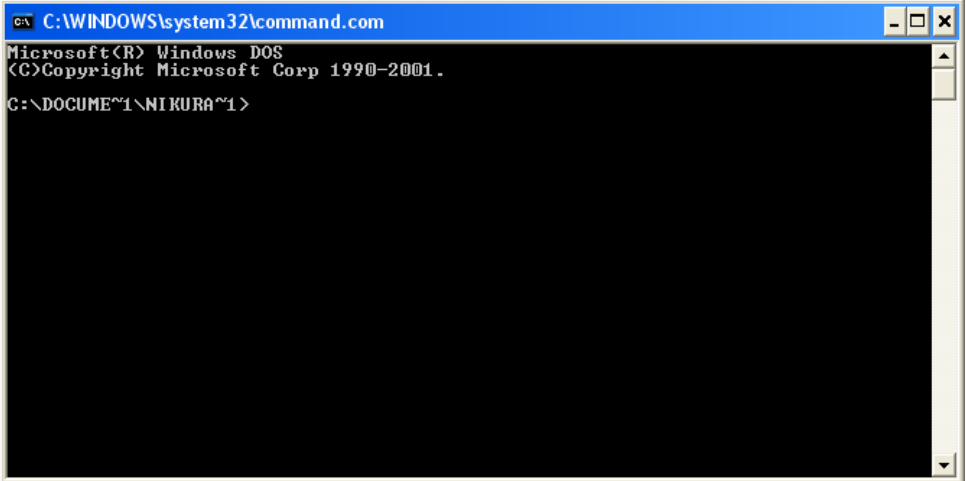
## Консольное приложение – это

**Консольное приложение** — программа, предназначенная для работы в операционной системе MS-DOS (или в окне DOS), для которой устройством ввода является клавиатура, а устройством вывода — монитор, работающий в режиме отображения символьной информации (буквы, цифры и специальные знаки).

**Консольное приложение** — программа, работающая в режиме текстового интерфейса.

-Экономия ресурсов

- Повышение быстродействия



```
C:\WINDOWS\system32\command.com
Microsoft(R) Windows DOS
(C) Copyright Microsoft Corp 1990-2001.
C:\DOCUMENTS\NIKURAN\1>
```

## Программные системы

- ✓ КОМПАС – система трехмерного моделирования для деталей и чертежей
- ✓ MATLAB – интерактивная среда для численных расчетов, содержащая большой набор математических функций
- ✓ Mathcad – система компьютерной алгебры из класса систем автоматизированного проектирования
- ✓ STATISTICA – программный комплекс для проведения статистического анализа, выявления корреляционных зависимостей
- ✓ LabView – среда графического программирования, являющаяся удобным средством для проектирования измерительных каналов, приборов, систем. Обеспечивает построение и моделирование измерительных структур различной сложности

**Спасибо за внимание!**