

Лекция №4

**Инструментальные
программные
средства. Системы
программирования**

В данный класс программного обеспечения входят средства (инструментарии) для создания других программ и программных комплексов.

Система программирования — комплекс языковых и программных средств, предназначенных для автоматизации процесса составления, отладки программы и подготовки ее к выполнению.

Программа — это логически упорядоченная последовательность команд, необходимых для управления компьютером (выполнения им конкретных операций), поэтому программирование сводится к созданию последовательности команд, необходимой для решения определенной задачи.

Программа, с которой работает процессор, представляет собой последовательность чисел, называемую **машинным кодом**.

Управлять компьютером нужно по определенному алгоритму. **Алгоритм** — это точно определенное описание способа решения задачи в виде конечной (по времени) последовательности действий. Такое описание еще называется **формальным**.

Для представления алгоритма в виде, понятном компьютеру, служат **языки программирования**. Сначала всегда 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, либо выполняется.

Сегодня практически все программы создаются с помощью языков программирования. Теоретически программу можно написать и средствами обычного человеческого (естественного) языка — это называется программированием на **метаязыке** (подобный подход обычно используется на этапе составления алгоритма), но автоматически перевести такую программу в машинный код пока невозможно из-за высокой неоднозначности естественного языка.

Языки программирования — искусственные языки.

От естественных они отличаются ограниченным числом «слов», значение которых понятно транслятору, и очень строгими правилами записи команд (**операторов**).

Совокупность подобных требований образует **синтаксис языка программирования**, а **смысл** каждой команды и других конструкций языка — его **семантику**. Нарушение формы записи программы приводит к тому, что транслятор не может понять назначение оператора и выдает сообщение о синтаксической ошибке, а правильно написанное, но не отвечающее алгоритму использование команд языка приводит к семантическим ошибкам (называемым еще **логическими ошибками** или **ошибками времени выполнения**).

Процесс поиска ошибок в программе называется **тестированием**, процесс устранения ошибок — **отладкой**.

С помощью языка программирования создается не готовая программа, а только ее текст, описывающий ранее разработанный алгоритм. Чтобы получить работающую программу, надо этот текст либо автоматически перевести в машинный код (для этого служат **программы-компиляторы**) и затем использовать отдельно от исходного текста, либо сразу выполнять команды языка, указанные в тексте программы (этим занимаются **программы-интерпретаторы**).

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**.

В данном случае «низкий уровень» не значит «плохой». Имеется в виду, что операторы языка близки к **машинному** коду и ориентированы на конкретные команды процессора.

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 каждой модели процессора отличаются, конкретной компьютерной архитектуре соответствует свой язык ассемблера, и написанная на нем программа может быть использована только в этой среде.

Языки программирования высокого уровня значительно ближе и понятнее человеку, нежели компьютеру. 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. Разрабатывать программы на языках высокого уровня с помощью понятных и мощных команд значительно проще, а ошибок при создании программ допускается гораздо меньше.

Из **универсальных языков программирования** сегодня наиболее популярны следующие:

Бейсик (*Basic*) — для освоения требует начальной подготовки (общеобразовательная школа);

Паскаль (*Pascal*) — требует специальной подготовки (школы с углубленным изучением предмета и общетехнические вузы);

Си++ (C++), **Ява** (*Java*), **Си Шарп** (C#) — требуют профессиональной подготовки (специализированные средние и высшие учебные заведения).

Для каждого из этих языков программирования сегодня имеется немало систем программирования, выпускаемых различными фирмами и ориентированных на различные модели ПК и операционные системы.

Языки программирования баз данных отличается от алгоритмических языков прежде всего решаемыми задачами.

База данных — это файл (или группа файлов), представляющий собой упорядоченный набор записей, имеющих единообразную структуру и организованных по единому шаблону (как правило, в табличном виде). База данных может состоять из нескольких таблиц. Удобно хранить в базах данных различные сведения из справочников, картотек, журналов бухгалтерского учета и т. д.

Первые базы данных появились очень давно, как только появилась потребность в обработке больших массивов информации и выборки групп записей по определенным признакам. Для этого был создан **структурированный язык запросов SQL** (*Structured Query Language*). Он основан на мощной математической теории и позволяет выполнять эффективную обработку баз данных, манипулируя не отдельными записями, а *группами* записей.

Для управления большими базами данных и их эффективной обработки разработаны **СУБД** (Системы Управления Базами Данных). Практически в каждой СУБД помимо поддержки языка *SQL* имеется также свой уникальный язык, ориентированный на особенности этой СУБД и не переносимый на другие системы. Сегодня в мире насчитывается три ведущих производителя СУБД: *Microsoft (SQL Server)*, *IBM (DB2)* и *Oracle*.

С появлением персональных компьютеров были созданы так называемые настольные СУБД. Родоначальником современных языков программирования баз данных для ПК принято считать СУБД *dBase II*, язык которой был интерпретируемым. Затем для него были созданы компиляторы, появились СУБД *FoxPro* и *Clipper*, поддерживающие диалекты этого языка. Сегодня самой распространенной настольной СУБД стала система ***Microsoft Access***.

С активным развитием глобальной сети было создано немало реализаций популярных языков программирования, адаптированных специально для Интернета.

Все они отличаются характерными особенностями: языки являются интерпретируемыми, интерпретаторы для них распространяются бесплатно, а сами программы — в исходных текстах. Такие языки называют **скрипт-языками** (*HTML, Perl, PHP, Tcl/Tk, VRML, XML*).

Для создания программы необходимы:

- текстовый редактор;
- компилятор;
- редактор связей;
- библиотеки функций.

Как правило, в стандартную поставку входят как минимум три последних компонента, но хорошая **интегрированная система** включает в себя и специализированный текстовый редактор, причем почти все этапы создания программы в ней автоматизированы; после того как исходный текст введен, его компиляция и сборка выполняются одним нажатием клавиши. Это очень удобно, так как не требует ручной настройки множества параметров запуска компилятора и редактора связей, указания им нужных файлов вручную и т.д. Процесс компиляции обычно демонстрируется на экране: показывается, сколько строк исходного текста откомпилировано, или выдаются сообщения о найденных ошибках.

В современных интегрированных системах имеется еще один компонент — *отладчик*, который позволяет анализировать работу программы во время ее выполнения. С его помощью можно последовательно выполнять отдельные операторы исходного текста *по шагам*, наблюдая при этом, как меняются значения различных переменных.

Наиболее популярны следующие *визуальные среды быстрого проектирования программ для Windows*:

- *Basic: Microsoft Visual Basic;*
- *Pascal; Borland Delphi;*
- *C++: Microsoft Visual C++;*
- *Java: Borland J Builder,*
- *C#: Microsoft Visual Studio .NET, Borland C++Builder.*

Для разработки серверных и распределенных приложений можно использовать систему программирования *Microsoft Visual C++*, продукты фирмы *Borland*, практически любые средства программирования на *Java*.