

Системы программирования.

Система программирования — это система для разработки новых программ на конкретном языке программирования.

Современные системы программирования обычно предоставляют пользователям мощные и удобные средства разработки программ. В них входят:

компилятор или интерпретатор;
интегрированная среда разработки;
средства создания и редактирования текстов программ;
обширные библиотеки стандартных программ и функций;
отладочные программы, т.е. программы, помогающие находить и устранять ошибки в программе;
дружественная к пользователю диалоговая среда;
многооконный режим работы;
мощные графические библиотеки; утилиты для работы с библиотеками;
встроенный ассемблер;
встроенная справочная служба;
другие специфические особенности.

Популярные системы программирования – Turbo Basic, Quick Basic, Turbo Pascal, Turbo C.

В последнее время получили распространение системы программирования, ориентированные на создание Windows-приложений:

пакет Borland Delphi (Дельфи) — блестящий наследник семейства компиляторов Borland Pascal, предоставляющий качественные и очень удобные средства визуальной разработки. Его исключительно быстрый компилятор позволяет эффективно и быстро решать практически любые задачи прикладного программирования.

пакет Microsoft Visual Basic — удобный и популярный инструмент для создания Windows-программ с использованием визуальных средств. Содержит инструментарий для создания диаграмм и презентаций.

пакет Borland C++ — одно из самых распространённых средств для разработки DOS и Windows приложений.

Транслятор (англ. translator — переводчик) — это программа-переводчик. Она преобразует программу, написанную на одном из языков высокого уровня, в программу, состоящую из машинных команд.

Трансляторы реализуются в виде компиляторов или интерпретаторов. С точки зрения выполнения работы компилятор и интерпретатор существенно различаются.

Компилятор (англ. compiler — составитель, собиратель) читает всю программу целиком, делает ее перевод и создает законченный вариант программы на машинном языке, который затем и выполняется.

Интерпретатор (англ. interpreter — истолкователь, устный переводчик) переводит и выполняет программу строка за строкой.

После того, как программа откомпилирована, ни сама исходная программа, ни компилятор более не нужны. В то же время программа, обрабатываемая интерпретатором, должна заново переводиться на машинный язык при каждом очередном запуске программы.

Откомпилированные программы работают быстрее, но интерпретируемые проще исправлять и изменять.

Каждый конкретный язык ориентирован либо на компиляцию, либо на интерпретацию — в зависимости от того, для каких целей он создавался. Например, Паскаль обычно используется для решения довольно сложных задач, в которых важна скорость работы программ. Поэтому данный язык обычно реализуется с помощью компилятора.

С другой стороны, Бейсик создавался как язык для начинающих программистов, для которых построчное выполнение программы имеет неоспоримые преимущества.

Иногда для одного языка имеется и компилятор, и интерпретатор. В этом случае для разработки и тестирования программы можно воспользоваться интерпретатором, а затем откомпилировать отлаженную программу, чтобы повысить скорость ее выполнения.