**4 сл. Компью́терная програ́мма** — 1) комбинация компьютерных [инструкций](https://ru.wikipedia.org/wiki/Машинный_код) и [данных](https://ru.wikipedia.org/wiki/Данные_(вычислительная_техника)), позволяющая [аппаратному обеспечению вычислительной системы](https://ru.wikipedia.org/wiki/Аппаратное_обеспечение) выполнять [вычисления](https://ru.wikipedia.org/wiki/Вычисление) или функции управления (стандарт [ISO](https://ru.wikipedia.org/wiki/Международная_организация_по_стандартизации)/[IEC](https://ru.wikipedia.org/wiki/IEC)/[IEEE](https://ru.wikipedia.org/wiki/IEEE) 24765:2010)[[1]](https://ru.wikipedia.org/wiki/Компьютерная_программа" \l "cite_note-ISO24765-1);

2) [синтаксическая](https://ru.wikipedia.org/wiki/Синтаксис_(программирование)) единица, которая соответствует правилам определённого [языка программирования](https://ru.wikipedia.org/wiki/Язык_программирования), состоящая из определений и [операторов или инструкций](https://ru.wikipedia.org/wiki/Оператор_(программирование)), необходимых для определённой функции, задачи или решения проблемы (стандарт ISO/IEC 2382-1:1993)[[2]](https://ru.wikipedia.org/wiki/Компьютерная_программа" \l "cite_note-ISO2382-2).

5 сл. Физическая часть компьютера, которая выполняет эти инструкции, называется «железом» или аппаратной частью (например, процессор, материнская плата и т.д.). Данный урок является началом серии уроков по программированию на языке С++ для начинающих.

Процессор компьютера не способен понимать напрямую языки программирования, такие как C++, Java, Python и т.д. Очень ограниченный набор инструкций, которые изначально понимает процессор, называется машинным кодом (или «машинным языком»). То, как эти инструкции организованы, выходит за рамки данного введения, но стоит отметить две вещи.

6 сл. Во-первых, каждая команда (инструкция) состоит только из определенной последовательности (набора) цифр: 0 и 1. Эти числа называются битами (сокр. от «binary digit») или двоичным кодом.

Например, одна команда машинного кода архитектуры ×86 выглядит следующим образом:

10110000 01100001

Во-вторых, каждый набор бит переводится процессором в инструкции для выполнения определенного задания (например, сравнить два числа или переместить число в определенную ячейку памяти). Разные типы процессоров обычно имеют разные наборы инструкций, поэтому инструкции, которые будут работать на процессорах Intel (используются в персональных компьютерах), с большей долей вероятности, не будут работать на процессорах Xenon (используются в игровых приставках Xbox). Раньше, когда компьютеры только начинали массово распространяться, программисты должны были писать программы непосредственно на машинном языке, что было очень неудобно, сложно и занимало намного больше.

7 сл.

*Так как программировать на машинном языке — удовольствие специфическое, то программисты изобрели язык ассемблера. В этом языке каждая команда идентифицируется коротким именем (а не набором единиц с нулями), и переменными можно управлять через их имена. Таким образом, писать/читать код стало гораздо легче. Тем не менее, процессор все равно не понимает язык ассемблера напрямую. Его также нужно переводить, с помощью ассемблера, в машинный код.*

Ассемблер — это транслятор (переводчик), который переводит код, написанный на языке ассемблера, в машинный язык. В Интернете язык ассемблера часто называют просто «Ассемблер».

8 сл. Преимуществом Ассемблера является его производительность (точнее скорость выполнения) и он до сих пор используется, когда это имеет решающее значение. Тем не менее, причина подобного преимущества заключается в том, что программирование на этом языке адаптируется к конкретному процессору. Программы, адаптированные под один процессор, не будут работать с другим. Кроме того, чтобы программировать на Ассемблере, по-прежнему нужно знать очень много не очень читабельных инструкций для выполнения даже простого задания.

Например, вот вышеприведенная команда, но уже на языке ассемблера:

mov al, 061h

## 9 сл Высокоуровневые языки программирования

**\*\*Высокоуровневый язык программирования\*\*** — язык программирования, разработанный для быстроты и удобства использования программистом.

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.

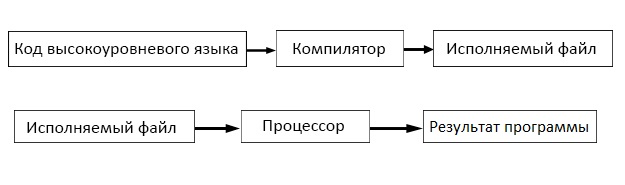
10 сл. Для решения проблем читабельности кода и чрезмерной сложности были разработаны высокоуровневые языки программирования. C, C++, Pascal, Java, JavaScript и Perl — это всё языки высокого уровня. Они позволяют писать и выполнять программы, не переживая о совместимости кода с разными архитектурами процессоров. Программы, написанные на языках высокого уровня, также должны быть переведены в машинный код перед выполнением. Есть два варианта:

   компиляция, которая выполняется компилятором;

   интерпретация, которая выполняется интерпретатором.

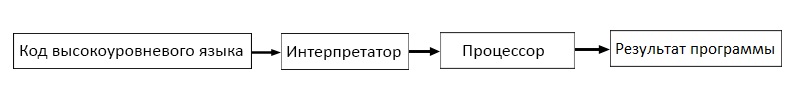
11 сл. Компилятор — это программа, которая читает код и создает автономную (способную работать независимо от другого аппаратного или программного обеспечения) исполняемую программу, которую процессор понимает напрямую. При запуске программы весь код компилируется целиком, а затем создается исполняемый файл и уже при повторном запуске программы компиляция не выполняется.

Проще говоря, процесс компиляции выглядит следующим образом:



12 сл. Интерпретатор — это программа, которая напрямую выполняет код, без его предыдущей компиляции в исполняемый файл. Интерпретаторы более гибкие, но менее эффективные, так как процесс интерпретации выполняется повторно при каждом запуске программы.

Процесс интерпретации:



Любой язык программирования может быть компилируемым или интерпретируемым, однако, такие языки, как C, C++ и Pascal — компилируются, в то время как «скриптовые» языки, такие, как Perl и JavaScript — интерпретируются. Некоторые языки программирования (например, Java) могут как компилироваться, так и интерпретироваться.

13 сл. Программное обеспечение, установленное на ПК, делится на 3 разновидности:

* прикладное;
* системное;
* инструментальное.

14 сл., 15 Это часть системы, которая помогает следить за аппаратной стороной ПК и управлять ею. Сюда входят программы, контролирующие работу оперативной памяти, центрального процессора, видеокарты, устройств ввода и вывода информации, сетевые подпрограммы.

Таким ПО считается:

* Драйверы — утилиты небольшого размера, функционирование которых заключается в обеспечении корректной работы остальных элементов оборудования;
* ОС;
* Дополнения — языковые пакеты или настройки расширения экрана.
* Основное отличие системной разновидности считается то, что она не рассчитана на выполнение конкретной поставленной задачи. Она необходима, чтобы обеспечивать бесперебойную работу остальных частей компьютера. Ее можно назвать посредником между оборудованием — «железом» и программным кодо

16,17 сл. **Прикладная программа**, или **приложение**, — [программа](https://ru.wikipedia.org/wiki/Компьютерная_программа), предназначенная для выполнения определённых задач и рассчитанная на непосредственное взаимодействие с пользователем[[*источник не указан 592 дня*](https://ru.wikipedia.org/wiki/Википедия:Ссылки_на_источники)]. В большинстве [операционных систем](https://ru.wikipedia.org/wiki/Операционная_система) прикладные программы не могут обращаться к ресурсам [компьютера](https://ru.wikipedia.org/wiki/Компьютер) напрямую, а взаимодействуют с оборудованием и другими программами посредством операционной системы.

Наиболее обширная доля классификации. Сюда относятся графические и текстовые редакторы, браузеры, базы данных и все, что люди используют в привычной работе за компьютером. Здесь же находятся антивирусные пакеты, бухгалтерия и различные архивы.

Смысл этой разновидности в выполнении четко поставленной задачи: рисовать, учитывать, открывать сетевые страницы, набирать текст. Если утилита нужна для конкретного выполнения действия, то она является прикладным ПО.

### 18 сл. Инструментальное

**Инструмента́льное програ́ммное обеспе́чение** — [программное обеспечение](https://ru.wikipedia.org/wiki/Программное_обеспечение), предназначенное для использования в ходе [проектирования](https://ru.wikipedia.org/wiki/Проектирование_программного_обеспечения), [разработки](https://ru.wikipedia.org/wiki/Разработка_программного_обеспечения) и сопровождения программ, в отличие от [прикладного](https://ru.wikipedia.org/wiki/Прикладное_программное_обеспечение) и [системного](https://ru.wikipedia.org/wiki/Системное_программное_обеспечение) программного обеспечения[[1]](https://ru.wikipedia.org/wiki/Инструментальное_программное_обеспечение" \l "cite_note-1).

Специфическое обеспечение любой компьютерной техники. Его можно было бы отнести к прикладному, но из-за специфики применения его выделили в отдельный вид. Основная функция — отладка, настройка, переписывание программного кода.

Сюда входят компиляторы, отладчики, переводчики высокого уровня, редакторы, интерпретаторы и другие средства. Они необходимы, потому что техника не понимает человеческих слов. Чтобы ей «объяснить», что надо сделать, требуется специальный «машинный язык».

Постоянно пользоваться этим кодом базовым пользователям довольно сложно, поэтому были разработаны системы, которые позволяют переводить обычную речь в двоичную, привычную для ПК.

Разница между часто используемыми компиляторами и интерпретаторами заключается в том, что первый генерирует готовый файл, который можно запускать. А второй создает архив, который функционирует только с помощью самого сервиса.

19 сл. Основные напправления разработки приложнеий

\* Веб-разработка

\* Мобильная разработка

\* Десктопные приложения

\* Разработка игр

\* Data Science

\* Программирование встроенных систем

\* Интернет вещи (IoT)

20 сл Веб-разработка

**\*\*Веб-разработка\*\*** — процесс создания *\*веб-сайта\** или *\*веб-приложения\**. Основными этапами процесса являются *\*веб-дизайн\**, *\*вёрстка страниц\**, программирование на стороне клиента и сервера, а также конфигурирование веб-сервера.

21 сл Мобильное приложение — это программное обеспечение, специально разработанное под конкретную мобильную платформу (iOS, Android и т. д.). Предназначено для использования на смартфонах, фаблетах, планшетах, умных часах и других мобильных устройствах.

22 сл. **\*\*Десктопное приложение\*\*** — программа, которая устанавливается на компьютер пользователя и работает под управлением операционной системы. Такие приложения высокопроизводительные, могут работать напрямую с принтерами, сканерами, факсами и прочей техникой.

23 сл Рзработка игр