Прогресс компьютерных технологий определил процесс появления новых разнообразных знаковых систем для записи алгоритмов – языков программирования. Смысл появления такого языка – оснащенный набор вычислительных формул дополнительной информации, превращает данный набор в алгоритм. Язык программирования служит двум связанным между собой целям: он дает программисту аппарат для задания действий, которые должны быть выполнены, и формирует концепции, которыми пользуется программист, размышляя о том, что делать. Первой цели идеально отвечает язык, который настолько "близок к машине", что всеми основными машинными аспектами можно легко и просто оперировать достаточно очевидным для программиста образом. Второй цели идеально отвечает язык, который настолько "близок к решаемой задаче", чтобы концепции ее решения можно было выражать прямо и коротко. (1-2 слайд)

### Язык 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.

**Понятие язык программирования**

Функция: язык программирования предназначен для написания компьютерных программ, которые применяются для передачи компьютеру инструкций по выполнению того или иного вычислительного процесса и организации управления отдельными устройствами.

Задача: язык программирования отличается от естественных языков тем, что предназначен для передачи команд и данных от человека компьютеру, в то время, как естественные языки используются для общения людей между собой. В принципе, можно обобщить определение «языков программирования» – это способ передачи команд, приказов, четкого руководства к действию; тогда как человеческие языки служат также для обмена информацией.

Исполнение: язык программирования может использовать специальные конструкции для определения и манипулирования структурами данных и управления процессом вычислений.

### Семантика языков программирования

– Семантика описывает последствия выполнения конструкций языка с помощью языка логики и задания пред- и постусловий.

### Компилируемые и интерпретируемые языки

– Языки программирования могут быть разделены на компилируемые и интерпретируемые

**Компилируемые**

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.

**Интерпретируемые**

Интерпретируемые языки обладают некоторыми специфическими дополнительными возможностями (см. выше), кроме того, программы на них можно запускать сразу же после изменения, что облегчает разработку. Программа на интерпретируемом языке может быть зачастую запущена на разных типах машин и операционных систем без дополнительных усилий. Однако интерпретируемые программы выполняются заметно медленнее, чем компилируемые, кроме того, они не могут выполняться без дополнительной программы-интерпретатора.

### Системы объектно-ориентированного программирования

Содержат программу-транслятор и позволяют работать в режиме как интерпретатора, так и компилятора. На этапе разработки и отладки проекта используется режим интерпретатора, а для получения готовой программы – режим компилятора.

### История развития языков программирования

Всю историю компьютерной индустрии и компьютерных наук с определенной точки зрения можно представить как историю развития языков программирования. Меняются времена, усложняются задачи, то, что раньше требовало человеко-лет, нынче энтузиасты делают на коленке за несколько недель; накоплена огромная масса типовых решений, типовых библиотек и типовых программистов. А создание, развитие и изменение языков программирования идет полным ходом.

**Машинный язык**

Программы на машинном языке – очень длинные последовательности единиц и нулей, являлись машинно зависимыми, т.е. для каждой ЭВМ необходимо было составлять свою программу.

**Ассемблер (начало 50-ых годов XX в.)**

Язык Ассемблера – это символическое представление машинного языка. Он облегчает процесс программирования по сравнению с программированием в машинных кодах.

Программисту не обязательно употреблять настоящие адреса ячеек памяти с размещенными в них данными, участвующими в операции, и вычисляемые результаты, а также адреса тех команд, к которым программа не обращается.

Некоторые задачи, например, обмен с нестандартными устройствами обработки данных сложных структур невозможно решить с помощью языков программирования высокого уровня. Это под силу ассемблеру.

В принципе, язык Ассемблер является машинным языком. И программист реализующий какую-либо задачу на языках высокого уровня, с помощью Ассемблера может определить осмыслено ли решение данной задачи, с точки зрения использования ЭВМ.

**Первые языки программирования высокого уровня**

С середины 50-ых гг. XX в. начали создавать первые языки программирования высокого уровня (high-level language). Эти языки были машинно независимыми (не привязаны к опред. типу ЭВМ). Но для каждого языка были разработаны собственные компиляторы.

Примеры таких языков: FORTRAN (FORmula TRANslator; 1954) предназначен для научных и технических расчетов; COBOL (1959) был предназначен в основном для коммерческих приложений (обрабатывал большие объемы нечисловых данных) – Common Business-Oriented Language); язык BASIC (Beginner’s All Purpose Instuction Code – универсальный язык символьных инструкций для начинающих) (1964 г.)

**БЕЙСИК**

Бейсик – это продукт Новой Англии. Cозданный в 1964г., как язык обучения программированию. Бейсик является общепринятым акронимом от"Beginner's All-purpose Symbolic Insruction Code" (BASIC) - Многоцелевой Символический Обучающий Код для Начинающих".

Вскоре как обучаемые, так и авторы программ обнаружили, что Бейсик может делать практически все то, что делает скучный неуклюжий Фортран. А так как Бейсику было легко обучиться и легко с ним работать, программы на нем писались обычно быстрее, чем на Фортране. Бейсик был также доступен на персональных компьютерах, обычно он встроен в ПЗУ. Так Бейсик завоевал популярность. Интересно, что спустя 20 лет после изобретения Бейсика, он и сегодня самый простой для освоения из десятков языков общецелевого программирования, имеющихся в распоряжении любителей программирования. Более того, он прекрасно справляется с работой.

Несмотря на высказывания снобов - сторонников языков Си и Паскаля, Бейсик считается деловым языком, снабженным мощными средствами решения специфических задач, которые обычно большинство пользователей решают при помощи небольших компьютеров, а именно: работая с файлами и выводя текстовое и графическое изображение на экране дисплея.

Современный Фортран (Fortran 95 и Fortran 2003) приобрёл черты, необходимые для эффективного программирования для новых вычислительных архитектур; позволяет применять современные технологии программирования, в частности, ООП.

Алго́л– название ряда языков программирования, применяемых при составлении программ для решения научно-технических задач на ЭВМ.

**Алгоритмические языки программирования**

С начала 80-ых г. XX в. начали создаваться языки программирования, которые позволили перейти к структурному программированию (использование операторов ветвления, выбора, цикла и практически отказ от частого использования операторов перехода (goto). К этим языкам относятся:

* язык Pascal (назван его создателем Никлаусом Виртом в честь великого физика Блеза Паскаля; 1970);
* язык Си, позволяющий быстро и эффективно создавать программный код (1971)ю

Язык программирования Pascal был создан Никлаусом Виртом, и назван в честь французского философа и математика XVIIв. Блеза Паскаля.

**Языки объектно-ориентированного программирования (90-ые г. XX в.).**

В основу этих языков положены программные объекты, которые объединяют данные и методы их обработки. В этих языках сохранялся алгоритмический стиль программирования. Для них были разработаны интегрированные среды программирования, позволяющие визуально конструировать графический интерфейс приложений:

* язык С++ (1983) – продолжение алгоритм. языка Си.
* язык Object Pascal (1989) был создан на основе языка Pascal. После создания среды программирования – Delphi (1995).
* язык Visual Basic(1991) был создан корпорацией Microsoft на основе языка Qbasic (1975) для разработки приложений с графическим интерфейсом в среде ОС Windows.

**Языки программирования на платформе .NET.**

Интегрированная среда программирования Visual Studio .Net, разработанная корпорацией Microsoft, позволяет создавать приложения на различных языках объектно-ориентированного программирования, в том числе:

* на языке Visual Basic .Net ( на основе Visual Basic) - 2003 г.;
* на языке Visual C# (С-шарп) – на основе языков С++ и J – 2003 г.;
* на языке Visual J# (J-шарп) – на основе Java и JavaScript – 2003 г..

### ВАЖНО!!!

Изобретение языка программирования высшего уровня позволило нам общаться с машиной, понимать её (если конечно Вам знаком используемый язык), как понимает американец немного знакомый с русским языком древнюю азбуку Кириллицы. Проще говоря, мы в нашем развитии науки программирования пока что с ЭВМ на ВЫ. Поверьте мне это не сарказм вы только посмотрите как развилась наука программирования с того времени, как появились языки программирования, а ведь язык программирования высшего уровня, судя по всему ещё младенец. Но если мы обратим внимание на темпы роста и развития новейших технологий в области программирования, то можно предположить, что в ближайшем будущем, человеческие познания в этой сфере, помогут произвести на свет языки, умеющие принимать, обрабатывать и передавать информации в виде мысли, слова, звука или жеста. Так и хочется назвать это детище компьютеризированного будущего: «языки программирования "высочайшего" уровня». Возможно, концепция решения этого вопроса проста, а ближайшее будущее этого проекта уже не за горами, и в этот момент, где нибудь в Запорожье, Амстердаме, Токио или Иерусалиме, перед стареньким 133MHz горбится молодой, никем не признанный специалист и разрабатывает новейшую систему искусственного интеллекта, которая наконец-то позволит человеку, с помощью своих машинных языков, вести диалог с машиной на ТЫ.

Размышляя над этим, хочется верить в прогресс науки и техники, в высоко - компьютеризированное будущее человечества, как единственного существа на планете, пусть и не использующего один, определенный разговорный язык, но способного так быстро прогрессировать и развивать свой интеллект, что и перехода от многоязыковой системы к всеобщему пониманию долго ждать не придется.

Завершить свой проект хорошо бы на такой оптимистичной ноте, но нет, напоследок хочется сказать:

Единственный способ изучать новый язык программирования – писать на нём программы.