**Дата проведения: 25.01.2022г.**

**Группа 1-7**

**Предмет: Информатика**

**Преподаватель: Амирханова А. К.**

**Тема урока: Введение в язык программирования. Синтаксис и семантика программы.**

Язык программирования — формальный язык, предназначенный для записи  
компьютерных программ. Язык программирования определяет набор лексических,  
синтаксических и семантических правил, определяющих внешний вид программы и  
действия, которые выполнит исполнитель (обычно — ЭВМ) под её управлением.

Язык программирования предназначен для написания компьютерных программ,  
которые представляют собой набор правил, позволяющих компьютеру выполнить тот  
или иной вычислительный процесс, организовать управление различными  
объектами, и т. п.  
Язык программирования отличается от естественных языков тем, что предназначен  
для управления ЭВМ, в то время как естественные языки используются, прежде  
всего, для общения людей между собой.

Со времени создания первых программируемых машин было создано более двух с  
половиной тысяч языков программирования. Ежегодно их число пополняется  
новыми.  
Некоторыми языками умеет пользоваться только небольшое число их собственных  
разработчиков, другие становятся известны миллионам людей. Профессиональные  
программисты обычно применяют в своей работе несколько языков  
программирования.

Языки программирования низкого уровня  
Первым компьютерам приходилось программировать двоичными машинными  
кодами. Однако программировать таким образом - достаточно трудоемкая и сложная  
задача.  
Для упрощения этой задачи стали появляться языки программирования низкого  
уровня, которые позволяли задавать машинные команды в более понятном для  
человека виде. Для преобразования их в двоичный код были созданы специальные  
программы - трансляторы.

Трансляторы делятся на:  
компиляторы - превращают текст программы в машинный код, который можно  
сохранить и затем использовать уже без компилятора (примером являются  
исполняемые файлы с расширением \*. exe).  
интерпретаторы - превращают часть программы в машинный код, выполняют и  
после этого переходят к следующей части. При этом каждый раз при выполнении  
программы используется интерпретатор.

Примером языка низкого уровня является ассемблер. Языки низкого уровня  
ориентированы на конкретный тип процессора и учитывают его особенности,  
поэтому для переноса программы на ассемблере на другую аппаратную платформу ее  
нужно почти полностью переписать.

Преимущества  
С помощью языков низкого уровня создаются эффективные и компактные  
программы, поскольку разработчик получает доступ ко всем возможностям  
процессора.  
Недостатки  
Программист, работающий с языками низкого уровня, должен быть высокой  
квалификации, хорошо понимать устройство микропроцессорной системы, для  
которой создается программа.  
результирующая программа не может быть перенесена на компьютер или  
устройство с другим типом процессора.  
значительное время разработки больших и сложных программ.

Языки низкого уровня, как правило, используют для написания небольших  
системных программ, драйверов устройств, модулей стыков с нестандартным  
оборудованием, программирование специализированных микропроцессоров, когда  
важнейшими требованиями являются компактность, быстродействие и возможность  
прямого доступа к аппаратным ресурсам.

Языки программирования высокого уровня  
Данный вид языков более понятен человеку, чем компьютеру. Особенности  
конкретных компьютерных архитектур в них не учитываются, поэтому созданные  
программы легко переносятся с компьютера на компьютер.  
В основном достаточно просто перекомпилировать программу под определенную  
компьютерную архитектурную и операционную систему.  
Разрабатывать программы на таких языках гораздо проще и ошибок допускается  
меньше. Значительно сокращается время разработки программы, что особенно важно  
при работе над большими программными проектами.

Достоинства языков программирования высокого уровня:  
алфавит языка значительно шире машинного, что делает его гораздо более  
выразительным и существенно повышает наглядность и понятность текста;  
набор операций, допустимых для использования, не зависит от набора машинных  
операций, а выбирается из соображений удобства формулирования алгоритмов  
решения задач определенного класса;  
конструкции операторов задаются в удобном для человека виде;  
поддерживается широкий набор типов данных.

Недостатком языков высокого уровня является больший размер программ по  
сравнению с программами на языке низкого уровня. Поэтому в основном языки  
высокого уровня используются для разработок программного обеспечения  
компьютеров и устройств, которые имеют большой объем памяти.

Синтаксис и семантика программы  
Алфавит — разрешенный к использованию набор символов, с помощью которого  
могут быть образованы слова и величины данного языка.  
Синтаксис — система правил, определяющих допустимые конструкции языка  
программирования из букв алфавита.  
Семантика — система правил однозначного толкования каждой языковой  
конструкции, позволяющих производить процесс обработки данных (алгоритм).

Как программировали раньше  
Язык С и Windows API  
Данный подход можно считать традиционным: программы писались на языке С с  
использованием интерфейса Windows API (Aplication Programming Interface –  
интерфейс прикладного программирования). Это проверенный временем подход  
программирования.  
Основная проблема данного подхода в том, что С лаконичный язык. Любой Спрограммист вынужден мириться с необходимостью «вручную» управлять памятью,  
иметь дело с указателями и синтаксическими конструкциями.  
И еще С – это структурный язык программирования и ему не хватает преимуществ,  
которые обеспечиваются объектно-ориентированным подходом.

Структурный язык программирования – это методология разработки  
программного обеспечения, в основе которой лежит представление программы в виде  
иерархической структуры блоков.  
Объектно-ориентированное  
программирование  
–  
это  
методология  
программирования, основанная на представлении программы в виде совокупности  
объектов, каждый из которых является экземпляром определённого класса, а классы  
образуют иерархию наследования.

Язык С++ и библиотека базовых классов  
С появлением С++ программистам стало доступно объектно-ориентированное  
программирование и его основные принципы – инкапсуляция, наследование и  
полиморфизм.  
Но не смотря на поддержку ООП, программисты вынуждены мириться с деталями  
языка С (выделение памяти вручную, указатели и т.д.). Поэтому было решено  
упростить работу – так появились платформы для программирования С++. Одна из  
самых популярных называется MFC (Microsoft Foundation Classes – библиотека базовых  
классов Microsoft).

Visual Basic 6.0  
Первым языком программирования, с которого многие начинают это язык Basic.  
Данный язык специально был предназначен для новичков т.к. очень прост. Язык был  
разработан в 1964 году, а вторую жизнь Basic получил с появлением Visual Basic от  
Microsoft.  
На Visual Basic возможно создавать сложные пользовательские интерфейсы,  
библиотеки программного кода и логику доступа к базам данных. Основной недостаток  
данного языка заключается в том, что он не является полностью объектноориентированным. Можно сказать, что он просто «объектный».  
С появлением .NET все недостатки языка устранены, но теперь он называется VB.NET.  
В этом современном языке поддерживается переопределение операций (перезагрузка),  
классическое наследование, конструкторы типов, обобщения и многое другое.

Язык Java  
Java – это объектно-ориентированный язык программирования, который по своему  
синтаксису похож на С++, но при этом Java не имеет многих неприятных  
синтаксических аспектов, которые присутствуют в C++. И самое главное что Java – это  
возможность написания кросс-платформенного кода.  
Кроссплатформенность — способность программного обеспечения работать более  
чем на одной аппаратной платформе и (или) операционной системе.

У Java есть одна очень важная проблема: Его очень сложно интегрировать с другими  
языками, поскольку он задумывался изначально как единственный язык  
программирования и единственная платформа для удовлетворения любой  
потребности.

Модель компонентных объектов  
Модель COM (Component Object Model – модель компонентных объектов) была  
предшествующей платформой для разработки приложения, которая предлагалась  
Microsoft перед .NET. Впервые COM появилась в 1993 году.  
Модель COM позволяет строить типы в соответствии с правилами COM и получать  
блок многократно используемого кода. Такие двоичные блоки кода называют  
«серверами COM». Одно из преимуществ сервера COM в том, что к нему можно  
получить доступ, используя другой язык программирования.

Модель COM можно считать успешной объектной моделью, однако её внутреннее  
устройство является очень сложной для восприятия, именно поэтому программистам  
требуется много времени на изучение этой модели.

**Ответы и вопросы отправить на эл. почту**

[asiyat.karimullaevna@yandex.ru](mailto:asiyat.karimullaevna@yandex.ru) или 8928-507-47-03