1. Язык Джава

Java – это **многоплатформенный, объектно-ориентированный и сетецентрический язык, который сам по себе может использоваться как платформа**

Что такое программа?

Программа — перечень инструкций, в совокупности определяющий алгоритм решения некоторой задачи. Инструмент автоматизации повторяющихся действий или ручного труда

человека.

Программирование — процесс реализации программ.

Главная особенность Джавы: кроссплатформенность, за счёт отличного от остальных жизненного цикла самой программы. В отличчи от других программа проходит через комплияцию в байт-код и запуск jvm

1. IDE (Integrated Development Environment) — пакет программных средств, которые предоставляют программисту удобный интерфейс для компиляции, запуска и отладки программ.

JDK (Java Development Kit) — пакет программных инструментов, которые предоставляют программисту средства для компиляции, запуску, отладке своих программ. В отличии от сред разработки (IDE), JDK предоставляет только инструменты и не предоставляет средств для автоматизации процессов в жизненном цикле программного обеспечения. При использовании JDK, исходный код программы необходимо подготовить в каком-либо текстовом редакторе.JIT-компиляция, динамическая компиляция — технология увеличения производительности программных систем, использующих байт-код, путём компиляции байт-кода в машинный код или в другой формат непосредственно во время работы программы

JRE— минимальная реализация виртуальной машины, необходимая для исполнения Java-приложений, без компилятора и других средств разработки. Состоит из виртуальной машины — Java Virtual Machine — и библиотеки Java-классов.

1. Переменная — это, как уже ранее оговаривалось, некоторый именованный указатель на фрагмент памяти, содержащий значение необходимых данных

Преобразование типов — это возможность работать со значением в памяти одного типа, как со значением другого

1. Инициализация переменной означает явное (или неявное) установление некоторого значения переменной.

Присва́ивание — механизм связывания в программировании, позволяющий динамически изменять связи имён объектов данных (как правило, переменных) с их значениями

1. Программы можно (условно) разделить на две категории — линейные и нелинейные. Линейная программа предполагает последовательное выполнение набора операций. В то время, как нелинейная программа предполагает выполнение различного набора операций в зависимости от некоторых условий. Например, можно рассмотреть два алгоритма, представленных на слайде. Слева располагается алгоритм, который является линейным, в противовес ему, блок-схема справа предлагает нелинейный алгоритм.
2. Первый инструмент — оператор if. Он необходим для выполнения некоторого числа операторов в зависимости от некоторых условий. Синтаксис оператора во всех его возможных вариантах представлен на слайде. Рассмотрим некоторые элементы этого оператора подробнее. Условие — это выражение, тип результата вычисления которого является boolean, т. е. логический тип. Это говорит о том, что условием может быть любой логический оператор, операторы сравнения, а также переменные типа boolean. В примере, который представлен на слайде мы осуществляем вычисление выражение t > LIMIT , результатом которого является значение логического типа. В зависимости от полученного в условии результата будет выполнено выражение, либо будет осуществлен переход на другую ветвь else оператора if, если таковая ветвь имеется.

Тернарный оператор — это оператор, использующий три аргумента для осуществления заданной операции. Что это за операция и какие необходимы аргументы? Синтаксис тернарного оператора имеет следующий вид: условие ? выражение\_1 : выражение\_2 ;

Int delta=x>0 ? x : Math.abs(x);

Оператор ветвления switch удобен в тех случаях, когда стоит сравнить некоторое

значение с некоторым набором констант и в случае успешного сравнения, выполнить какоелибо действие. Блок схема работы оператора switch представлена на слайде.

Пример использования оператора switch представлен в правой части слайда. Давайте на этом примере рассмотрим механизм работы оператора ветвления

Цикл switch.

В круглых скобках, стоящих сразу после ключевого слова switch, указывается имя переменной, которая может быть целочисленной, строковой либо являться перечислимым типом данных (о последнем речь пойдет в лекциях по темам ООП). Затем в фигурных скобках находится тело оператора, состоящее из блоков case. После ключевого слова case в каждом блоке указывается константа с которой следует сравнить переменную x. В случае успешного сравнения будет выполнено соответствующее действие. В конце блока может находится ключевое слово break, которое в данном контексте инициирует выход из оператора switch. Если это ключевое слово опущено, то после выполнения действий в текущем блоке, управление будет передано действию, находящемуся в следующем блоке (в примере у нас таких блоков два: case 2 и case 3. В случае, если совпадения не нашлось и присутствует необязательный элемент default, то будут выполнены действия, объявленные внутри этого блока «по умолчанию».

switch(a){  
case 1: b=1;  
break;  
 case 2:b=2;  
 break;  
 case 10: b=323;}

Синтаксис определенного цикла представлен на слайде в первой строке. Разберем его

составляющие:

Блок инициализации — это выражения в которых, как правило, объявляются переменные область видимости которых ограничивается этим циклом (т. е. вне цикла этими переменными нельзя пользоваться). Условие — выражение, результат вычисления которого имеет логический тип. Блок вычисления — это выражения в которых, как правило, происходит модификация объявленных в блоке инициализации переменных. Цикл продолжает выполняться, пока условие возвращает true.