Ключевые моменты по урокам курса Основы Java

### Урок 1

1. Что такое компьютерная программа.

Компьютерная программа – это последовательность команд, на понятном компьютеру языке, которые будут выполнены компьютером при запуске такой программы.

1. Зачем нужен язык программирования.

Язык программирования необходим для того что бы человек мог свои мысли переводить на машинный язык, понятный компьютеру, реализуя тем самым своего рода диалог между человеком и машиной.

1. Что такое переменная.

Переменная – это именованная область памяти компьютера в которой можно хранить данные определенного типа, получать к ним доступ и изменять их в ходе выполнения компьютерной программы, где она используется.

1. Простые типы в Java.

К просты типам в Java обносяться: int, long, double, char, boolean

1. Что такое строка в Java.

Строка в Java это экземпляр объектного типа String. Записывается как любая последовательность символов заключенная в двойные кавычки: “Abc123!%,.:?”.

### Урок 2

1. Что такое метод, зачем он нужен.

Метод это именованный блок кода, который реализует определенную логику. Метод можно объявить, описать его тело и вызвать, когда это необходимо, из другого участка кода. В зависимости от входящих данных - передаваемых в метод параметров, мы будем получать разные результаты, используя один и тот же блок кода - метод. Использование методов реализует идею повторного использования кода. То есть написанный один раз код метода, можно использовать в дальнейшем, где и сколько угодно раз, просто помня, что возвращает или выполнят, используемый метод, получив на вход те или иные параметры. При этом необязательно знать или помнить саму реализацию метода.

Синтаксис:

<return type> | void <name>(<parameter1>,... <parameterN>) {

//Body

[return <result>;]

}

1. Что такое ссылка.

Ссылка это переменная, которая хранит адрес другой переменной, в которой, в свою очередь хранятся данные.

1. Как осуществляется передача параметров в метод (простые и ссылочные типы).

Передача параметров в метод осуществляется по значению. При передаче переменных простых типов данных в метод в качестве параметров, значения этих переменных просто копируются в вызываемый метод, при этом передаваемые параметры останутся неизменными, в той части программы (напр. в main()), откуда вызывается метод.

При передаче параметров объектных или ссылочных типов в метод, ситуация аналогичная, только при этом передаются копии значений ссылок на объекты. И если внутри метода мы переопределим переданную ссылку, то внешняя ссылка не изменится, а поменяется только значение её копии внутри метода.

1. Зачем нужен оператор if.

Условный оператор if реализует логические ветвления и позволяет Java программе, в зависимости от условия, выполнить то или иное действие, основываясь на значении (true/false) булевой переменной или логического выражения.

1. Зачем нужны циклы.

Жизнь циклична. Часто бывает необходимость неоднократно выполнять те или иный действия в программе. Для этого существую циклы. В Java мы научились использовать while и for.

### Урок 3

1. Что такое массив и как он представлен в памяти компьютера.

Массив это последовательность проиндексированных элементов одного типа. Доступ к каждому элементу массива осуществляется по его индексу. Массив имеет заданную длину или размер, которая соответствует количеству всех его элементов. Индекс первого элемента массива равен 0, последнего – на единицу меньше размера массива.

Массив представлен в памяти как переменная – имя массива, которая содержит адрес первого элемента массива с индексом [0], за которым далее последовательно распложены остальные элементы массива. Вся эта цепочка занимает определенный объём памяти, который зависит от типа элементов массива, то есть числа байт, которое отводиться под каждый элемент массива и общего числа элементов массива.

1. Что из себя представляет многомерный массив.

Многомерный массив это массив, элементы которого являются массивами. Чаще всего используется двумерные массив. Такой массив можно представить в виде матрицы, где доступ к элементам осуществляется по двум индексам – номеру строки и номеру столбца ячейки в которой находиться элемент массива. Правда это условное представление. На самом деле в памяти компьютера, например в случае с двумерным массивом, мы имеем два массива. Каждый элемент первого, назовем его основным, массива хранит адрес первого элемента массива, который храниться в этом элементе. Если у нас трех-, четырех- и т.д. мерный массив данный принцип организации многомерных массивов в памяти компьютера просто раскручивается по цепочке.

### Урок 4

1. Что такое компиляция.
2. Запуск программ из командной строки, зачем это нужно.
3. Что такое класс и объект.
4. Зачем нужны конструкторы.
5. Зачем скрывать переменные класса (делать их private) и обращаться к ним через getter, setter.
6. Что такое enum.

### Урок 5

1. Что такое наследование.

Наследование — это когда один класс дополняет или расширяет возможности другого класса.

Для того что бы определить иерархию классов, необходимо взять все классы одного типа и, выделив в них общие характеристики и свойства – методы, вынести их в отдельный класс.

У каждого класса должен быть только один класс родитель, а вот классов потомком может быть сколько угодно. При этом доступна любая вложенность.

1. Как работает method overriding.

Method overriding позволяет в классе потомке написать специфическую реализацию метода, который уже реализован в родительском классе. При переопределении метода из родительского класса в классе потомке, описывается метод с точно таким же названием, типом возвращаемого результата и принимаемыми параметрами. Вызов метода происходит по иерархической цепочке снизу вверх. Например, если мы вызываем метод, который содержится в классе родителе и, который переопределен в классе потомке, то вызывается именно реализация метода из класса потомка.

1. Отношение Наследник — Родитель.

Когда мы наследуем один объект от другого, это означает, что объект наследник сам по себе является объектом родителем (parent). Когда мы создаем экземпляр конкретного класса наследника, то экземпляр наследник имеет свойства и функции родителя, которые в него встраиваются, что позволяет обращаться к нему как к родителю, при этом он имеет такое же поведение. Например: Flute is an (extends) Instrument означает, что экземпляр класса Fluent будет иметь весь такой же функционал и такое же поведение, что и экземпляр класса Instrument.

При наследовании используется механизм upcasting – преобразование типа вверх по иерархии наследования. Upcasting позволяет передать в метод объект класса, который наследует класс, принимаемый этим методом.

1. Delegation (передача полномочий), Composition (включение).

Delegation (делегирование) – это передача полномочий выполнения методов одного класса – другому классу. При делегировании говорят о втором типе наследовании “has a”, когда один объект включает в себя другой объект. При организации делегирования необходимым условием есть реализация возможности обратиться ко всем не private методам класса, которому делегируются полномочия. В противном случае, если нет обращения ко всем не private методам класса, который включен в реализуемый класс, говорят о включении (composition) одного класса в другой. Иными словами делегация (delegation) – это частный случай включения (composition), при котором в классе, который делегирует управление, реализуются все не private методы класса, которому делегируется управление, путем написания аналогичных по названию и параметрам методов, которые просто вызывают методы класса которому делегируется управление.

Наиболее популярные случаи использования delegation: когда удобней построить иерархию классов путем включения одного класса в другой, избегая полного расширения одного класса другим; когда необходимо добавить в класс дополнительный функционал, не изменяя при этом сам класс, например, когда нет доступа на изменение, метод/класс является final или такова техническая реализация.

1. Статические данные, ключевое слово static.

Ключевое слово статик может применяться к переменным, методам и классам.

Статические переменные и методы — это переменные и методы которые принадлежат самому классу, а не его объекту/экземпляру и создаются, когда JVM загружает сам класс, при первом обращении к нему или его упоминании в программе. Статические переменные и методы можно вызывать не создавая экземпляр класса. При реализации статических методов нельзя использовать не статические переменны и методы класса.

Чаще всего статический переменные используются, когда необходимо реализовать константы, при этом после static добавляется ключевое слово final. Статические методы чаще всего применяться при реализации утилитных классов, для обработки объектов этого же класса (пример класс String), при реализации метода по созданию объектом самого себя.

1. Зачем нужны классы-обвёртки (wrappers).

Урок