**Вопросы к защите лабораторной работы:**

1. Объектно-ориентированное программирование. Основные понятия: объекты, наследование, полиморфизм, инкапсуляция.
2. Понятие класса. Классы и объекты в Java.
3. Члены класса. Модификаторы доступа.
4. Создание и инициализация объектов. Вызов методов.
5. Области видимости переменных.
6. Модификаторы final и static.
7. Пакеты, инструкция import.

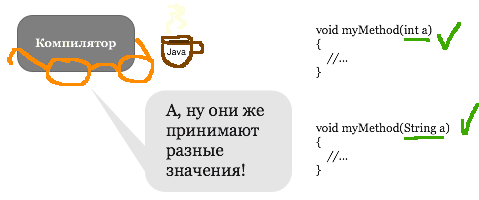
1)**Инкапсуляция**

Одним из самых важных факторов при проектировании компонентов приложения является сокрытие внутренних данных компонента и деталей его реализации от других компонентов приложения и предоставление набора методов для взаимодействия с ним (API). Этот принцип является одним из четырёх фундаментальных принципов ООП и называется **инкапсуляцией**.

коротко: скрытие реализации

**Наследование** — свойство системы, позволяющее описать новый класс на основе уже существующего с частичноили полностью заимствующейся функциональностью. Класс, от которого производится наследование, называется базовым, родительским или суперклассом. Новый класс — потомком, наследником, дочерним или производным классом.

**Полиморфизм** — свойство системы использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта



^перегрузка ф-ций

еще примерчик

Car car1 = new BMW()

Car car2 = new MERS()

void BuyCar(Car c).....

MERS, BMW наследуются от Car

**Абстракция** — означает выделение значимой информации и исключение из рассмотрения незначимой. С точки зрения программирования это правильное разделение программы на объекты. Абстракция позволяет отобрать главные характеристики и опустить второстепенные.

2)Класс – это шаблон, описывающий общие свойства группы объектов. описание объекта. Объект- экземпляр класса

Любой **объект** может обладать двумя основными характеристиками: состояние - некоторые данные, которые хранит объект, и поведение - действия, которые может совершать объект.

Для хранения состояния объекта в классе применяются поля или переменные класса. Для определения поведения объекта

в классе применяются методы.

Конструкторы вызываются при лаба по проге лаба по инфе лаба по проге лаба по инфе по проге лаба по инфе ии нового объекта данного класса. Конструкторы выполняют инициализацию объекта.

super — используется для обращения к базовому классу, а this к текущему.

3.)

public или protected, или private, - определяющие область видимости переменной;

public переменная доступна всем классам

private переменная доступна только текущему классу

default переменная доступна из данного пакета. по умолчанию

protected переменная из текцщего пакета и дочерних классов.

static - указывает, что переменная является переменной класса, а не переменной экземпляра класса;

final - запрещает присвоение значений переменной вне тела класса, в котором она объявлена; Attributes and methods cannot be overridden/modified

transient - указывает, что переменная не является частью постоянного состояния объекта; (не может быть десериализован)

volatile - определяет возможность асинхронного изменения переменной.

Результат операции записи значения в volatile переменную одним потоком, становится виден всем другим потокам, которые используют эту переменную для чтения из нее значения. для многопоточности

7)

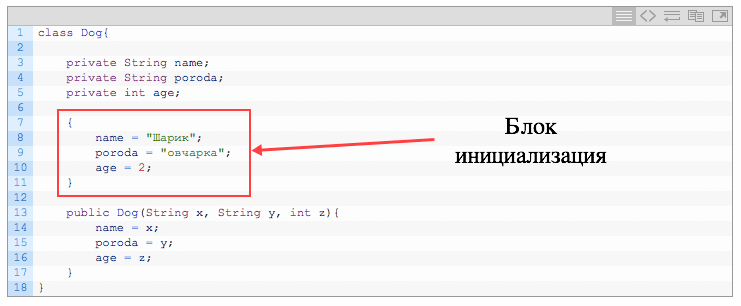
пакет - это типа папка по сути

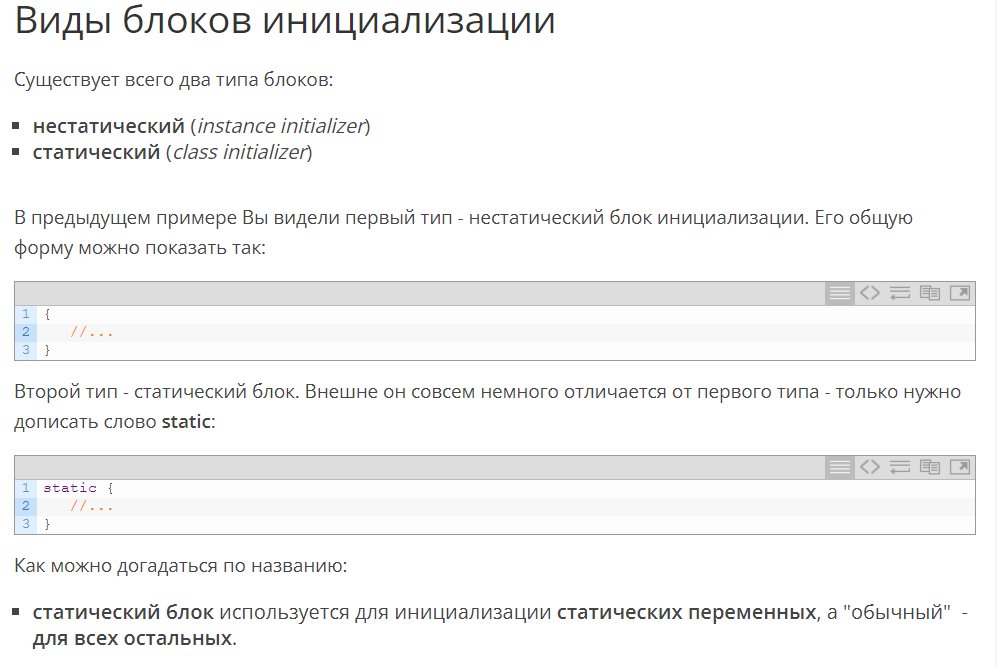
import *packagename*.*Class*; // Import a single class

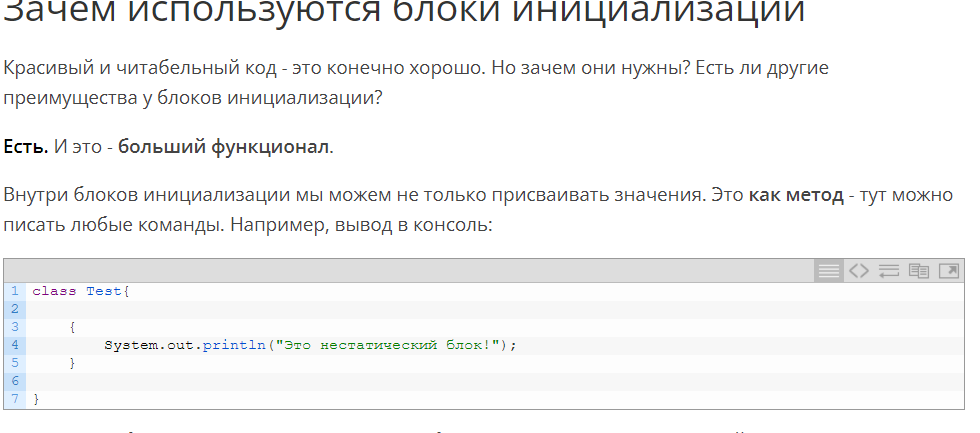
import *packagename*.\*; // Import the whole package

8)допвопрос от пред. курсов. блоки инициализации

Инициализация (от англ. initialize, от **initial** - "начальный, первоначальный") - это когда мы **впервые** задаем переменной какое-либо значение.







обычные блоки кода типа for(){} if(){} это те же блоки инициализации по сути?