Введение в ООП. Классы и объекты

№ урока: 1 **Курс:** Java Essential

Средства обучения: Компьютер с установленной IntelliJ IDEA.

Обзор, цель и назначение урока

Рассмотрение понятий «Класс», «Объект». Парадигмы ООП. Модификаторы доступа. Getters, Setters. Рассмотрение Packages как части инкапсуляции.
Ключевые слова “this”, “null”.

Изучив материал данного занятия, учащийся сможет:

- Понимать принципы ООП.
- Понимать использование Пакетов.
- Понимать концепцию использования модификаторов доступа.
- Значение ключевых слов «this», «null»
- Создавать Классы, Объекты
- Вызывать методы Класса через Объекты

Содержание урока

1. Понятие класса.
2. Понятие объекта.
3. Объявление класса.
4. Понятия «Инкапсуляция», «Полиморфизм», «Наследование»
5. Модификаторах доступа «private», «default», «protected», «public»
6. Пакеты.
7. Создание Объекта. Ключевое слово «new». Устройство памяти JVM.
8. Геттеры и Сеттеры.
9. Ключевое слово «null»

Резюме

ООП (Объектно-ориентированное программирование) – парадигма программирования, в которой основными концепциями являются понятия объектов и классов.

Класс – прототип, с которого создается объект. Класс является основой для приложения; он содержит методы и переменные, которые являются его составляющими. Поля определяют состояние, а методы **–** поведение будущего объекта.

Класс – модель объекта в реальной жизни (Кошка, Машина, Дом, и т.д.). Два экземпляра данного класса могут содержать разные данные, но они всегда имеют одни и те же методы. Существует только один класс автомобилей, но приложение может создать много разных объектов автомобилей (Спорт, Грузовик, Маршрутка).

Объявление класса включает в себя название, поля, конструкторы и методы.

Конструктор – специальный блок инструкций, вызываемый при создании объекта.

Объект – это экземпляр Класса. Создается через ключевое слово “new”. Пример: Класс «Машина», а его объекты: «Спорт», «Грузовик» и т.д. Объект реализует поведение, которое заложено в Классе.

Инстанцирование (instantiation) – создание экземпляра класса. В отличие от слова «создание», применяется не к объекту, а к классу. То есть, говорят: «создать экземпляр класса или инстанцировать класс».

Инкапсуляция используется для скрытия значения или состояния объекта внутри класса, предотвращая прямой доступ несанкционированный сторон к этим значениям. Публично доступные методы, как правило, так же предоставляются в классе (так называемые методы получения(get) и

CyberBionic Systematics ® 2016
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine

t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

назначения(set)) для доступа к значениям, и другие подклассы могут вызывать эти методы для получения и изменения значения в пределах объекта.

Метод доступа get – используется для получения значения из переменной (чтение).

Метод доступа set – используется для записи значения в переменную (запись).

Модификаторы доступа – определяют видимость членов класса.

- **private –** класс, метод, поле и т.д., которые объявлены как private могут быть доступны только в Классе, где объявлены.
- **default (package-private) –** класс, метод, поле и т.д., которые Не имеют никакого модификатора доступа могут быть доступны только в рамках пакета, где класс был объявлен.
- **protected –** класс, метод, поле и т.д., которые объявлены как protected могут быть доступны только для классов-наследников (subclasses) и в рамках пакета, где класс объявлен.
- **public** – класс, метод, поле и т.д., которые объявлены как public могут быть доступны из любого другого класса.

Полиморфизм является способностью объекта принимать различные формы. Наиболее распространенное использование полиморфизма в ООП происходит, когда ссылка на суперкласс используется, чтобы обратиться к объекту сабкласса.

Shape shape = new Triangle();

Наследование – механизм языка, позволяющий описать новый класс на основе уже существующего (родительского, базового) класса или интерфейса. Потомок может добавить собственные методы и свойства, а также пользоваться родительскими методами и свойствами.

Superclass – родительский класс

Subclass – класс наследник

Память JVM – делится на 3 части: (Stack, HEAP, PermGen) + выделяется RAM на работу самой JVM (~256 mb). В Stack хранятся примитивные типы данных и ссылки на объекты. В HEAP хранятся объекты. В PermGen хранится метаданные, используемая самой JVM (используемые классы, методы и т.п.).

new – ключевое слово в Java, через которое создается новый Объект.

null – это значение по умолчанию любых ссылочных типов, проще говоря, для всех объектов.

Package:

1) package **–** это механизм группирования классов, которые связаны друг с другом по каким-то характеристикам. Если пакет не был создан руками, то Java создает пакет по умолчанию, но это плохая практика.

2) Так же пакеты можно рассматривать как часть Инкапсуляции.

3) Ключевое слово **import** – используется для импортирования нужного нам класса с другого пакета.

4) Пакеты поддерживают иерархическую организацию, и используются для организации больших программ в логические и управляемые единицы.

java.lang.System;
java.lang.String;
java.util.List;
java.util.Map;
java.awt.Button;

5) Корневой пакет **java.** В нем содержатся 3 subpackage **lang, util, awt**.

6) Пакет **lang** всегда импортируется по умолчанию

7) Если пакет содержит много классов и все они нам нужны для работы, то что бы не импортировать каждый по отдельности, используется «*****»

java.util.*;

8) Возможно использовать класс из другого пакета без его импорта. Вы можете написать **полное** имя класса. **Полное Имя класса** состоит из **полного** пути, включая пакет, содержащий класс.

com.itvdn.oop.Shape myShape = new com.itvdn.oop.Shape.Triangle();

9) **import static** – позволяет использовать поля и методы, объявленные в классе как public static без указания на данный класс.

Закрепление материала

- Что такое Класс?
- Что такое объект?

CyberBionic systematics

CyberBionic Systematics ® 2016
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine

t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- Что такое экземпляр класса?
- Что такое ООП?
- Назовите основные парадигмы ООР.
- Что такое инкапсуляция?
- Какие модификаторы доступа вы знаете? Для чего они используются?
- Как работает память JVM.
- Что такое пакеты, зачем они нужны?
- Что такое статический импорт?

Дополнительное задание

Задание
Используя IDEA, создайте проект с пакетом.
Требуется: Создать класс с именем **Address**. В теле класса требуется создать поля**: index, country, city, street, house, apartment**.
Для каждого поля, создать **метод с двумя методами доступа** (get, set)
Создать экземпляр класса **Address**.
В поля экземпляра записать информацию о почтовом адресе.
Выведите на экран значения полей, описывающих адрес.

Самостоятельная деятельность учащегося

Задание 1
В любой из профильных **книг (Хорстман, Эккель)** найти соответствующие темы и закрепить материал. Использование **YouTube, Quizful** приветствуется.

Задание 2
Используя IDEA, создайте проект с пакетом. Требуется: Создать класс с именем **Rectangle**. В теле класса создать два поля, описывающие длины сторон **double side1, double side2**. Создать два метода, вычисляющие площадь прямоугольника – **double areaCalculator (double side1, double side2)** и периметр прямоугольника – **double perimeterCalculator (double side1, double side2)**. Написать программу, которая принимает от пользователя длины двух сторон прямоугольника и выводит на экран периметр и площадь.

Задание 3
Используя IDEA, создайте проект с пакетом.
Требуется: Создать класс **Book(Main)**. Создать классы **Title, Author** и **Content**, каждый из которых должен содержать одно строковое поле и метод **void show ()**. Реализуйте возможность добавления в книгу названия книги, имени автора и содержания. Выведите на экран при помощи метода **show()** название книги, имя автора и Содержание.

Задание 4
Используя IDEA, создайте проект с пакетом. Создать класс **Computer**, создать массив объектов **Computers** размером **5**. Далее руками создать **5** экземпляров этого класса и записать в компьютер (используя **loop)**

Рекомендуемые ресурсы

https://docs.oracle.com/javase/tutorial/java/concepts/index.html

https://docs.oracle.com/javase/tutorial/java/concepts/class.html

https://docs.oracle.com/javase/tutorial/java/javaOO/accesscontrol.html

https://www.jetbrains.com/idea/help/creating-packages.html

CyberBionic systematics
CyberBionic Systematics ® 2016
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine
t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com