Вступ до ООП. Класи й об'єкти

№ уроку: 1 **Курс:** Java Essential

Засоби навчання: Комп'ютер із встановленою IntelliJ IDEA

Огляд, мета та призначення уроку

- Розгляд понять «Клас», «Об'єкт».
- Парадигми ООП.
- Модифікатори доступу.
- Getters, Setters.
- Розгляд Packages як частини інкапсуляції.
- Ключові слова «this», «null».

Вивчивши матеріал цього заняття, учень зможе:

- Розуміти принципи ООП.
- Розуміти використання пакетів.
- Розуміння концепції використання модифікаторів доступу.
- Значення ключових слів «this», «null».
- Створювати класи, об'єкти.
- Викликати методи класу через об'єкти.

Зміст уроку

1. Концепція класу.
2. Концепція об'єкта.
3. Оголошення класу.
4. Поняття «Інкапсуляція», «Поліморфізм», «Успадкування».
5. Модифікатори доступу «private», «default», «protected», «public».
6. Пакети.
7. Створення об'єкта. Ключове слово «new». Пристрій пам'яті JVM.
8. Гетери та сетери.
9. Ключове слово «null».

Резюме

ООП (Об'єктно-орієнтоване програмування) – парадигма програмування, у якій основними концепціями є поняття об'єктів та класів.

Клас – прототип, з якого створюється об'єкт. Клас є основою застосунку: він містить методи та змінні, які є його складниками. Поля визначають стан, а методи – поведінку майбутнього об'єкта.
Клас – модель об'єкта в реальному житті (Кішка, Машина, Будинок тощо). Два екземпляри певного класу можуть містити різні дані, але вони завжди мають одні й самі методи. Є лише один клас автомобілів, але застосунок може створити багато різних об'єктів автомобілів (Спорт, Вантажівка, Маршрутка).

Оголошення класу вміщує назву, поля, конструктори та методи.

Конструктор – спеціальний блок інструкцій, що викликається під час створення об'єкта.

Об'єкт – це екземпляр класу. Створюється за допомогою ключового слова «new». Приклад: Клас «Машина», а його об'єкти «Спорт», «Вантажівка» тощо. Об'єкт реалізує поведінку, яка закладена у класі.

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

t. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Інстанціювання (instantiation) – створення екземпляра класу. На відміну від слова «створення», застосовується не до об'єкта, а до класу. Тобто кажуть: «створити екземпляр класу чи інстанціювати клас».

Інкапсуляція використовується для приховання значення або стану об'єкта всередині класу, запобігаючи прямому доступу несанкціонованих сторін до цих значень. Публічно доступні методи, як правило, також надаються в класі (так звані методи отримання (get) і призначення (set)) для доступу до значень, інші підкласи можуть викликати ці методи для отримання та зміни значення в межах об'єкта.

Метод доступу get – використовується отримання значення зі змінної (читання).
Метод доступу set – використовується для запису значення змінну (запис).
Модифікатори доступу визначають видимість членів класу.

- **private** – клас, метод, поле тощо, які оголошені як private можуть бути доступні лише у класі, де оголошені.
- **default (package-private)** – клас, метод, поле тощо, які не мають ніякого модифікатора доступу, можуть бути доступні тільки в межах пакета, де клас був оголошений.
- **protected** – клас, метод, поле тощо, які оголошені як protected, можуть бути доступні тільки для класів-спадкоємців (subclasses) та в межах пакета, де клас оголошений.
- **public** – клас, метод, поле тощо, які оголошені як public, можуть бути доступні з будь-якого іншого класу.

Поліморфізм є спроможністю об'єкта набувати різних форм. Найбільш поширене використання поліморфізму в ООП відбувається, коли посилання на суперклас використовується, щоб звернутися до об'єкта сабкласу.
Shape shape = new Triangle();

Успадкування – механізм мови, що дає змогу описати новий клас на основі вже наявного (батьківського, базового) класу чи інтерфейсу. Нащадок може додати власні методи та властивості, а також користуватися батьківськими методами та властивостями.
Superclass – батьківський клас.
Subclass – клас спадкоємець.

Пам'ять JVM ділиться на 3 частини: (Stack, HEAP, PermGen) + виділяється RAM на роботу самої JVM (~256 mb). У Stack зберігаються примітивні типи даних та посилання на об'єкти. У HEAP зберігаються об'єкти. У PermGen зберігаються метадані, які використовує сама JVM (використовуються класи, методи тощо).
new – ключове слово у Java, через яке створюється новий об'єкт.
null – це значення за замовчуванням будь-яких типів посилань, простіше кажучи, для всіх об'єктів.

Package.

1. package – це механізм групування класів, які пов'язані один з одним за якимись характеристиками. Якщо пакет не був створений руками, Java створює пакет за замовчуванням, але це погана практика.
2. Також пакети можна розглядати як частину інкапсуляції.
3. Ключове слово **import** використовується для імпортування потрібного нам класу з іншого пакета.
4. Пакети підтримують ієрархічну організацію та використовуються для організації великих програм у логічні та керовані одиниці.
   java.lang.System;
   java.lang.String;
   java.util.List;
   java.util.Map;
   java.awt.Button;
5. Кореневий пакет **java**. У ньому містяться 3 subpackage **lang**, **util**, **awt**.
6. Пакет **lang** завжди імпортується за замовчуванням.
7. Якщо пакет містить багато класів, і всі вони нам потрібні для роботи, то щоб не імпортувати кожен окремо використовується «*»
   java.util.*;

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

t. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

8. Можна використовувати клас з іншого пакета без його імпорту. Ви можете написати **повну** назву класу. **Повне ім'я класу** складається з **повного** шляху, вміщуючи пакет, що містить клас. **com.itvdn.oop.Shape myShape = new com.itvdn.oop.Shape.Triangle();**
9. **import static** дозволяє використовувати поля та методи, які оголошені у класі як public static, без вказівки на цей клас.

Закріплення матеріалу

- Що таке клас?
- Що таке об'єкт?
- Що таке екземпляр класу?
- Що таке ООП?
- Назвіть основні парадигми ООР.
- Що таке інкапсуляція?
- Які модифікатори доступу ви знаєте? Для чого вони використовуються?
- Як працює пам'ять JVM?
- Що таке пакети, для чого вони потрібні?
- Що таке статичний імпорт?

Додаткове завдання

Використовуючи IDEA, створіть проєкт із пакетом.
Потрібно: створити клас з ім'ям Address. У тілі класу потрібно створити поля: **index, country, city, street, house, apartment**.
Для кожного поля створити **метод із двома методами доступу** (get, set). Створити екземпляр класу **Address**. В поля екземпляра записати інформацію про поштову адресу. Виведіть на екран значення полів, що описують адресу.

Самостійна діяльність учня

Завдання 1
У будь-якій з профільних **книг (Горстманн, Еккель**) знайти відповідні теми та закріпити матеріал. Використання **YouTube, Quizful** вітається.

Завдання 2
Використовуючи IDEA, створіть проєкт із пакетом. Потрібно: створити клас з ім'ям **Rectangle**. У тілі класу створити два поля, що описують довжини сторін **double side1, double side2**. Створити два методи, що обчислюють площу прямокутника – **double areaCalculator (double side1, double side2)** і периметр прямокутника – **double perimeterCalculator (double side1, double side2)**. Написати програму, яка приймає від користувача довжини двох сторін прямокутника і виводить на екран периметр і площу.

Завдання 3
Використовуючи IDEA, створіть проєкт із пакетом.
Потрібно: створити клас **Book (Main).** Створити класи **Title**, **Author** та **Content**, кожен з яких має містити одне рядкове поле та метод **void show()**. Реалізуйте можливість додавання до книги назви книги, імені автора та змісту. Виведіть на екран за допомогою методу **show()** назву книги, ім'я автора та зміст.

Завдання 4
Використовуючи IDEA, створіть проєкт із пакетом. Створити клас **Computer**, створити масив об'єктів Computers розміром **5**. Далі руками створити **5** екземплярів цього класу та записати на комп'ютер (використовуючи **loop**).

Рекомендовані ресурси

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

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

t. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

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 ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

t. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com