



Java Essential

Класи й об'єкти

Java Essential

Після уроку обов'язково



Повторіть цей урок у форматі відео на [ITVDN.com](http://itvdn.com)

Доступ можна отримати через керівництво вашого навчального центру



Перевірте, як Ви засвоїли цей матеріал на [TestProvider.com](http://testprovider.com)

Класи й об'єкти

Об'єктно-орієнтований підхід

Парадигми ООП

Інкапсуляція – механізм мови, що дає змогу обмежити доступ одних компонентів програми до інших.

Успадкування – механізм, що дозволяє описати новий клас на основі вже наявного (батьківського), водночас властивості та функціональність батьківського класу запозичуються новим класом.

Поліморфізм – можливість об'єктів з однаковою специфікацією мати різну реалізацію.

Абстракція в ООП – це надання об'єктові характеристик, які відрізняють його від інших об'єктів, чітко визначаючи його концептуальні межі.

Абстрагування в ООП – це метод виділити набір значущих показників об'єкта, виключаючи з розгляду незначущі. Відповідно, абстракція – це набір всіх таких характеристик.

Ключові слова

Ключове слово this

this – ключове слово, яке є посиланням на ПОТОЧНИЙ ОБ'ЄКТ.

```
public Car(String model, int maxSpeed, int speed) {  
    this.model = model;  
    this.maxSpeed = maxSpeed;  
    this.speed = speed;  
}
```

У конструкторі ініціалізуються поля поточного об'єкта значеннями, переданими під час виклику конструктора

Конструктори

Конструктори

Конструктор – спеціальний метод створення екземпляра класу (ініціалізації об'єкта).

Ім'я конструктора відповідає імені класу.

```
public Car(String model, int maxSpeed, int speed) {  
    this.model = model;  
    this.maxSpeed = maxSpeed;  
    this.speed = speed;  
}
```

Конструктори

Конструктор за замовчуванням

Якщо програміст явно не задав конструктор у класі, JVM під час компіляції створить конструктор за замовчуванням.

```
public Car() { }
```

Якщо створено хоча б один конструктор у класі, конструктор за замовчуванням **НЕ** СТВОРЮЄТЬСЯ.

Конструктор **не може** бути `static`, `abstract` або `final`.

Конструктори

Перевантаження конструкторів

Конструктор, як і будь-який інший метод, можна перевантажувати. Під час створення об'єкта буде викликаний конструктор залежно від параметрів.

```
public Animal() {  
    this.age = 14;  
    this.height = 60;  
}
```

```
public Animal(int age) {  
    this.age = age;  
    this.height = 66;  
}
```

```
public Animal(int age, int height) {  
    this.age = age;  
    this.height = height;  
}
```

```
Animal animal = new Animal();
```

```
Animal animal = new Animal(15);
```

```
Animal animal = new Animal(20, 180);
```


Ключові слова

Статичний контекст

Модифікатор **static** визначає поле чи метод, що належить безпосередньо КЛАСУ.

Для виклику статичного поля чи методу не потрібно створювати екземпляр класу.

Виклик **нестатичного** методу

```
class A {  
    public int method(){...};  
}
```

```
public static void main(String[] args ) {  
    A a = new A();  
    a.method();  
}
```

Виклик **статичного** методу

```
class A {  
    static int method() {...};  
}
```

```
public static void main(String[] args ) {  
    A.method();  
}
```

Java Essential

Q&A

Інформаційний відеосервіс для розробників програмного забезпечення

