ТОМОВ ЩЕ НИ ЕБЕ 2.0

**1. Класове**

Класовете в Java представляват шаблони за създаване на обекти и съдържат описание на състоянието (чрез **полета**) и поведението (чрез **методи**) на обектите. Един клас дефинира:

* **Променливи** – те описват състоянието на обекта, като всеки обект има собствени стойности на тези променливи. Тези променливи могат да бъдат от различни типове – примитивни или референтни.
* **Методи** – това са действията, които обектите от този клас могат да извършват. Методите определят как обектите взаимодействат със света или с други обекти.

**Пример:**

public class Car {

private String model;

private int year;

public void drive() {

System.out.println("The car is driving.");

}

}

**2. Елементи на класовете:**

* **Декларация на клас** – класът се декларира с ключовата дума class, следвана от името на класа. Името на класа обикновено започва с главна буква.
* **Модификатори за достъп** – определят видимостта на класовете, методите и променливите. Най-често срещаните са:
  + **public** – достъпът е възможен от всяка точка на кода.
  + **protected** – достъпът е ограничен до класа, пакетите му и неговите наследници.
  + **private** – достъпът е ограничен само до самия клас.

**3. Конструктори**

Конструкторът е специален метод, използван за създаване на нови обекти от даден клас. Той инициализира стойностите на полетата на обекта. Конструкторите:

* Имат същото име като класа.
* Нямат тип на връщан резултат.
* Могат да приемат параметри или да бъдат без параметри.
* Ако не бъде създаден ръчно, Java автоматично генерира **конструктор по подразбиране** без параметри.

**Пример за конструктор:**

public class Car {

private String model;

private int year;

public Car(String model, int year) {

this.model = model;

this.year = year;

}

}

**4. Методи Get и Set**

Методите „get“ и „set“ се използват за достъп до частните полета на класа и за тяхната промяна, като осигуряват **капсулация** – защита на данните от директен достъп и модификация. **Get методите** връщат стойността на дадено поле, докато **Set методите** позволяват промяна на стойността.

**Пример:**

public class Car {

private String model;

public String getModel() {

return model;

}

public void setModel(String model) {

this.model = model;

}

}

**5. Наследяване**

Наследяването е процес, при който един клас (подклас) наследява свойствата и поведението на друг клас (родител). Наследяването се използва за създаване на йерархии от класове и за повторно използване на код. В Java един клас може да наследи само един родителски клас, което е известно като **единично наследяване**.

**Ключови моменти:**

* Подкласът има достъп до всички public и protected полета и методи на родителския клас.
* Подкласът разширява родителския клас, добавяйки нови полета или методи, специфични за него.
* Наследяването се осъществява чрез ключовата дума extends.

**Пример:**

public class Vehicle {

protected String brand;

public void honk() {

System.out.println("Beep beep!");

}

}

public class Car extends Vehicle {

private String model;

public void drive() {

System.out.println("The car is driving.");

}

}

**7. Създаване на обекти**

Обектите в Java се създават чрез инстанциране на класове с помощта на ключовата дума new. Това действие извиква конструктора на класа и заделя памет за новия обект. След създаването на обектите, методите и полетата на класа могат да бъдат достъпвани чрез оператора . (точка).

**Пример за създаване на обект:**

Car myCar = new Car("Toyota", 2020);

**8. Статични полета и методи**

Статичните полета и методи принадлежат на класа, а не на конкретен обект. Статичните полета се споделят между всички обекти от класа, докато статичните методи могат да бъдат извиквани директно чрез името на класа, без да е необходимо създаване на обект.

**Пример за статично поле:**

public class Car {

public static int totalCars = 0;

public Car() {

totalCars++;

}

}

Извикване на статично поле:

System.out.println(Car.totalCars);