Java: Klasa, Konstruktor, Dziedziczenie – Przykład na Samochodzie

# 1. Klasa i pola

W Javie klasa to szablon do tworzenia obiektów. Przykładowa klasa `Samochod` zawiera dane opisujące samochód: markę i rok produkcji.

public class Samochod {  
 private String marka;  
 private int rokProdukcji;  
}

# 2. Konstruktor domyślny

Konstruktor bezparametrowy (domyślny) ustawia wartości domyślne pól.

public Samochod() {  
 marka = "nieznana";  
 rokProdukcji = 0;  
}

# 3. Konstruktor z parametrami

Pozwala tworzyć obiekt z konkretnymi wartościami przekazanymi jako argumenty.

public Samochod(String marka, int rokProdukcji) {  
 this.marka = marka;  
 this.rokProdukcji = rokProdukcji;  
}

# 4. Metoda wypiszInfo()

Metoda wypisuje na ekranie informacje o danym obiekcie.

public void wypiszInfo() {  
 System.out.println("Marka: " + marka);  
 System.out.println("Rok produkcji: " + rokProdukcji);  
}

# 5. Dziedziczenie – SamochodElektryczny

Klasa `SamochodElektryczny` dziedziczy po klasie `Samochod` i dodaje pole `pojemnoscBaterii`.

public class SamochodElektryczny extends Samochod {  
 private int pojemnoscBaterii;  
  
 public SamochodElektryczny(String marka, int rok, int pojemnoscBaterii) {  
 super(marka, rok);  
 this.pojemnoscBaterii = pojemnoscBaterii;  
 }  
  
 public void wypiszInfo() {  
 super.wypiszInfo();  
 System.out.println("Pojemność baterii: " + pojemnoscBaterii + " kWh");  
 }  
}

# 6. Przykładowe użycie – metoda main()

public class Test {  
 public static void main(String[] args) {  
 Samochod s1 = new Samochod();  
 Samochod s2 = new Samochod("Toyota", 2022);  
 SamochodElektryczny e1 = new SamochodElektryczny("Tesla", 2023, 75);  
  
 s1.wypiszInfo();  
 System.out.println();  
 s2.wypiszInfo();  
 System.out.println();  
 e1.wypiszInfo();  
 }  
}