

# Psy w Javie

**Roman Podraza**  
**Instytut Informatyki**  
**Politechniki Warszawskiej**

# Obiektowo

- **Obiekty**
  - Klasy i obiekty
  - Odnośniki
  - Automatyczne odzyskiwanie pamięci
- **Dziedziczenie**
  - Dziedziczenie jednobazowe
  - Interfejsy
- **Klasy i metody statyczne**

# Dziedziczenie

## (przykłady stosowania)

```
public class Pies {  
    private String imie;  
    private int wiek;  
    public Pies (String i, int w) { // konstruktor  
        imie = i;  
        wiek = w;  
    }  
    public boolean stary() { // metoda  
        if (wiek > 10)  
            return true;  
        return false;  
    }  
}
```

```
public String dawImie() {  
    return imie;  
}  
public int dawWiek() {  
    return wiek;  
}  
public void zmienImie(String nowe) {  
    imie = nowe;  
}  
public void zmienWiek(int nowy) {  
    wiek = nowy;  
}  
public void urodziny() {  
    wiek++;  
}  
}
```

```
public class Uruchom {  
    public static void main(String[] args) {  
        Pies burek;  
        burek = new Pies("Burek", 12);  
        Pies ares = new Pies("Ares", 1);  
        ares.urodziny();  
        if (ares.stary())  
            System.out.println(ares.dajImie() + " jest "  
                                + "stary");  
        burek.zmienImie("Azor");  
        System.out.println(burek.dajImie() + " ma " +  
                            burek.dajWiek() + " lat.");  
    }  
}
```

```
public class Jamnik extends Pies {  
    private int dlugosc; // w cm  
    public Jamnik(String i, int w, int dl) {  
        super(i, w); // konstruktor klasy bazowej  
        dlugosc = dl;  
    }  
    public void rosnie (int oIle) {  
        if (oIle <= 0)  
            return;  
        dlugosc += oIle;  
    }  
    public int dajDlugosc() {  
        return dlugosc;  
    }  
}
```

```
public class Uruchom2 {  
    public static void main(String[] args) {  
        Pies burek;  
        Jamnik bagsio;  
        burek = new Pies("Burek", 12);  
        bagsio = new Jamnik("Bagsio", 8, 113);  
        bagsio.urodziny();  
        bagsio.rosnie(12);  
        if (bagsio.stary())  
            System.out.println(bagsio.dajImie()+ " jest "  
                               + "stary");  
        System.out.println(bagsio.dajImie() + " ma " +  
                           bagsio.dajDlugosc() + " cm.");  
    }  
}
```

```
public abstract class Mysliwski extends Pies {  
    abstract void poluje();  
    public Mysliwski(String i, int w) {  
        super(i, w); // konstruktor klasy bazowej  
    }  
}
```

```
public class Pointer extends Mysliwski {  
    public Pointer(String i, int w) {  
        super(i, w); // konstruktor klasy bazowej  
    }  
    public void poluje() {  
        System.out.println("Wystawia!");  
    }  
}
```



```
public class Basset extends Mysliwski {  
    public Basset(String i, int w) {  
        super(i, w); // konstruktor klasy bazowej  
    }  
    public void poluje() {  
        System.out.println("Tropi!");  
    }  
}
```

```
public class Uruchom3 {  
    public static void main(String[] args) {  
        Pies burek;  
        // Mysliwski NN; /*niedozwolone - abstrakcja*/  
        Pointer cepp;  
        Basset amik;  
        cepp = new Pointer ("Cepp", 5);  
        amik = new Basset ("Amik", 8);  
        System.out.print(cepp.dajImie()+ ". ");  
        cepp.poluje();  
        System.out.print(amik.dajImie()+ ". ");  
        amik.poluje();  
        burek = cepp;  
        // burek.poluje(); /* zwykły pies nie poluje */  
        cepp = (Pointer)burek;  
    }  
}
```

```
public interface Pracujacy {  
    void pracuje();  
}
```

```
public class Bernardyn extends Pies  
    implements Pracujacy {  
    public Bernardyn(String i, int w) {  
        super(i, w); // konstruktor klasy bazowej  
    }  
    public void pracuje() {  
        System.out.println("Rum!");  
    }  
    public boolean stary() {  
        return false;  
    }  
}
```

```
public class Husky extends Pies
    implements Pracujacy {
    public Husky(String i, int w) {
        super(i, w); // konstruktor klasy bazowej
    }
    public void pracuje() {
        System.out.println("Ciagnie!");
    }
}
```

```
public class Uruchom4 {  
    public static void main(String[] args) {  
        Pies burek;  
        // Mysliwski NN; /*niedozwolone - abstrakcja*/  
        Bernardyn ares = new Bernardyn("Ares", 12);  
        Husky leo = new Husky("Leo", 7);;  
        System.out.print(ares.dajImie()+ ". ");  
        ares.pracuje();  
        System.out.print(leo.dajImie()+ ". ");  
        leo.pracuje();  
        burek = ares;  
        if (burek.stary() != ares.stary())  
            System.out.println("Dziwne - ten sam pies.");  
        // leo = (Husky)burek; // nie mog zrobic z  
                                // bernardyna husky'ego  
    }  
}
```