

Zadanie 7 (Dziedziczenie)

1. Napisz klasę **Punkt3D**, będącą rozszerzeniem klasy **Punkt**. Klasa **Punkt3D** *dziedziczy* z klasy **Punkt**, czyli przejmuję wszystkie pola i metody oraz dodaje własne. Do definicji użyj słowa extends.
2. Dopisz do niej metody:
 - **ustawZ**
 - **pobierzZ()**
 - **ustawXYZ()**
3. Utwórz konstruktor
 - 3-argumentowy **Punkt3D(int wspX, int wspY, int wspZ)**
 - 1-argumentowy **Punkt3D(Punkt3D punkt)**klasy potomnej przy użyciu metody super() (wywołanie konstruktora klasy bazowej musi być pierwszą instrukcją konstruktora klasy potomnej)
4. Dopisz jeszcze jeden konstruktor, który jako argument będzie przyjmował obiekt klasy **Punkt**. Wykorzystaj metodę super().
5. W klasie **Punkt3D** napisz metodę **wyswietl**, która będzie *przesłaniać* metodę **wyswietl** z klasy **Punkt**.

PRZYKŁADY: Przeciążanie i przesłanianie a dziedziczenie

1. Przeciążanie a dziedziczenie <pre>public class A { public void f() { System.out.println("A"); } } public class B extends A { public void f(int liczba) { System.out.println("B"); } }</pre>	<pre>public class Test { public static void main(String[] args){ A a = new A(); B b = new B(); a.f(); a.f(1); b.f(); b.f(1); } }</pre> <p>// które wywołania są błędne?</p>
2. Przesłanianie metod (1) <pre>public class A { public void f() { System.out.println("A"); } } public class B extends A { public void f() { System.out.println("B"); } }</pre>	<pre>public class Test { public static void main(String[] args){ A a = new A(); B b = new B(); a.f(); b.f(); } }</pre> <p>// jaki będzie wynik działania programu</p>

3. Przesłanianie metod (2)

```
public class A
{
    public void f()
    {
        System.out.println("A");
    }
}

public class B extends A
{
    public void f()
    {
        super.f();
        System.out.println("B");
    }
    public void g()
    {
        f();
    }
    public void h()
    {
        super.f();
    }
}
```

```
public class Test {

    public static void main(String[] args){
        A a = new A();
        B b = new B();

        a.f();
        b.f();
        b.g();
        b.h();
    }

    // która metoda przesłania?
    // jaki będzie wynik działania programu
}
```

4. Przesłanianie pól

```
public class A
{
    protected int liczba;
}

public class B extends A
{
    protected int liczba;

    public void Ustaw_Liczba_A(int liczba)
    {
        super.liczba = liczba;
    }

    public void Ustaw_Liczba_B(int liczba)
    {
        this.liczba = liczba;
    }

    public int Pobierz_Liczba_A()
    {
        return super.liczba;
    }

    public int Pobierz_Liczba_B()
    {
        return liczba;
    }
}
```

```
public class Test {

    public static void main(String[] args){

        B b = new B();

        b.Ustaw_Liczba_A(0);
        b.Ustaw_Liczba_B(1);

        System.out.println("Wartosc przeslonietego pola
liczba: " + b.pobierz_Liczba_A());

        System.out.println("Wartosc pola liczba z klasy B: " +
b.pobierz_Liczba_B());

    }
}

// ile pól zawiera obiekt klasy B?
```

Zadanie 8 (Dziedziczenie)

1. Z klasy **Paczka** (plik na edux.pjwstk.edu.pl) wyprowadź klasę **PaczkaNazwa** z prywatnymi polami **nazwa** i **cel** typu String.
2. Zaprojektuj konstruktor:

public PaczkaNazwa(String nazwa, String cel, int dlugosc, int szerokosc, int wysokosc, int wartosc)

3. W klasie **Paczka** istnieje metoda **ObliczObj()** wyznaczająca objętość paczki. Przypuśćmy, że chcemy w klasie **PaczkaNazwa** sposób liczenia objętości nieco zmodyfikować. Dla paczek, w których pole podstawy przekracza 10000 przyjmijmy objętość równą 1000000 (może to być informacja o tym, że paczka się nie zmieści). Zaobserwuj przesłanianie metody.
4. Dodaj metodę **public String PodajIdent()**, która podaje w postaci jednego obiektu klasy **String** informację o nazwie paczki i miejscu przeznaczenia.