

1 Dziedziczenie

Dziedziczenie pozwala na tworzenie hierarchii klas i współdzielenia składowych

1.1 Przykład

```
class Base{  
    }  
  
public class Klasa extends Base{  
    }  
}
```

Klasa (Klasa) pochodna posiada wszystkie składowe klasy bazowej (Base). W języku java nie występuje wielodziedziczenie – dozwolone jest tylko dziedziczenie po jednej klasie.

2 Rzutowanie

Klasy mogą zostać rzutowane zarówno na swoje klasy bazowe (bezpiecznie) jak również na klasy potomne (niebezpiecznie).

2.1 Rzutowanie w górę

```
Klasa k = new Klasa();  
Bazowa b = k;
```

2.2 Rzutowanie w dół

```
Klasa k1 = (Klasa)k;
```

Konieczne podanie klasy na jaką rzutujemy

3 Instanceof

Operator *instanceof* pozwala na określenie czy dany obiekt jest instancją danej klasy.

```
if (k1 instanceof Klasa)
    System.out.println("inna jest Klasa");
```

4 this i super

this wskazuje na aktualny obiekt

super wskazuje na obiekt klasy bazowej

5 Zadania do wykonania

1. Stworzyć strukturę klas zadaną przez prowadzącego
2. Wykorzystać w klasie pochodnej metody i pola klasy bazowej
3. Stworzyć w klasie potomnej pola i metody o takich samych nazwach jak w klasie bazowej
4. Stworzyć w klasie potomnej statyczne pola i metody o takich samych nazwach jak w klasie bazowej
5. Sprawdzić możliwość rzutowania między poszczególnymi klasami
6. Sprawdzić możliwość dziedziczenia po klasie finalnej
7. Sprawdzić efekt zastosowania operatora *instanceof* na różnych obiektach
8. Wywołać konstruktory (parametrowe i domyślne) w klasach pochodnych
9. Sprawdzić możliwość odwołania się do elementów o różnych widocznościach z klas pochodnych
10. Jaki jest efekt gdy w klasie potomnej w konstruktorze brakuje *super*
11. Po czym dziedziczy klasa, która nie wykorzystuje *extends*