

Programowanie obiektowe w języku Java – lista zadań nr 5

opracował: dr inż. Jakub Długosz

Celem ćwiczenia jest przypomnienie czym jest przeciążanie (ang. overload) i przesiłanie (ang. override) metod. Ponadto zadania dotyczą klas statycznych i finalnych oraz koncepcji „object calisthenics”

1. Przeciążanie/przeładowywanie (ang. overloading) metod odbywa się wówczas, gdy dana metoda ma kilka różnych sygnatur w obrębie danej klasy. Zadania na liście zadań nr 3 dotyczyły przeciążania konstruktorów klas *Osoba* i *Pracownik*.

2. Nadpisanie/przesłonięcie (ang. overriding) metod odbywa się wówczas, gdy metoda klasy potomnej ma taką samą sygnaturę jak metoda klasy bazowej lecz działa w inny sposób.

3. Na stronie <https://docs.oracle.com/javase/8/docs/api/index.html> wyszukaj **klasę Math** w pakiecie **java.lang**.

Używając klasy *Math*, bez wykonywania żadnych obliczeń, wyświetl w programie liczbę π .

4. Utwórz klasę *Matematyka* która jest klasą potomną dla klasy *Math* i spróbuj skompilować program. Jaki komunikat otrzymałeś/eś i co on oznacza?

5. Utwórz klasę *OperacjeMatematyczne* zawierającą metodę *podzielTablice*, którą można wywołać statycznie, tzn. bez konieczności utworzenia obiektu klasy *OperacjeMatematyczne*. Metoda ta dzieli każdy element *tablicy licznik* przez odpowiadający mu element *tablicy mianownik*:

```
... podzielTablice(int[] licznik, int[] mianownik){
    for(int i=0; i< licznik.length; i++){
        System.out.println("Dzielenie " + licznik[i] + " przez " + mianownik[i] + ": "
            + licznik[i]/mianownik[i]);
    }
}
```

W klasie *Main* wywołaj **statycznie** funkcję *podzielTablice* dla następujących elementów:

```
int[] licznik={-12,4,1000};
int[] mianownik={4,2,-50};
```

Rezultat powinien być taki, jak poniżej:

```
Dzielenie -12 przez 4: -3
Dzielenie 4 przez 2: 2
Dzielenie 1000 przez -50: -20
```

6. Wywołaj metodę *podzielTablice* dla:

```
int[] licznik={-12,4,1000};  
int[] mianownik={4,0,-50};
```

Jaki komunikat pojawił się w konsoli i dlaczego?

7. Wywołaj metodę *podzielTablice* dla:

```
int[] licznik={-12,4,1000,800};  
int[] mianownik={4,2,-50};
```

Jaki komunikat pojawił się w konsoli i dlaczego?

8. Otocz blok kodu `System.out.println("Dzielić "+ licznik[i]+ " przez "+ mianownik[i]+": "+licznik[i]/mianownik[i]);` z punktu 5. instrukcjami `try{} catch{} finally{}` , tak aby w przypadku wywołania w punkcie 7. w konsoli pojawiały się następujące komunikaty:

```
Dzielić -12 przez 4: -3  
*****  
Dzielić 4 przez 2: 2  
*****  
Dzielić 1000 przez -50: -20  
*****  
Tablice są różnej długości.  
*****
```

a w przypadku wywołania jak w punkcie 6:

```
Dzielić -12 przez 4: -3  
*****  
Działanie ma błędne argumenty  
*****  
Dzielić 1000 przez -50: -20  
*****
```

9. Zapoznaj się z 9. regułami tzw. **kalisteniki obiektowej (ang. object calisthenics)**. Zasady te wykorzystują m.in. programiści w firmie Asus.

Reguły te są opisane [tutaj](#) w języku angielskim. Polską prezentację na ten temat można znaleźć [tutaj](#).

Zastosuj te reguły w swoim kodzie.