

# Zadania z programowania w języku Java dla II roku Informatyki

## Programowanie zorientowane obiektowo

1. Utwórz klasę **RachunekBankowy**. W klasie tej zadeklaruj zmienną statyczną typu **double** o nazwie **rocznaStopaProcentowa** do zapamiętania rocznej stopy procentowej dla wszystkich posiadaczy rachunków. Każdy obiekt klasy **RachunekBankowy** powinien zawierać prywatne pole **saldo** oznaczające bieżący stan oszczędności na rachunku. Napisz metodę **obliczMiesieczneOdsetki**, która służy do obliczania miesięcznych odsetek wg wzoru  $(saldo * rocznaStopaProcentowa) / 12$ . Po obliczeniu, odsetki powinny być dodane do pola **saldo**. Napisz statyczną metodę **setRocznaStopaProcentowa**, która nadaje polu **rocznaStopaProcentowa** nową wartość.

Napisz program testujący klasę **RachunekBankowy**. W programie tym utwórz dwa obiekty klasy **RachunekBankowy**, **saver1** oraz **saver2**, z początkowymi oszczędnościami 2000.00 zł oraz 3000.00 zł. Nadaj polu **rocznaStopaProcentowa** wartość 4%, oblicz miesięczne odsetki i wypisz nowy stan oszczędności dla obu rachunków. Następnie nadaj polu **rocznaStopaProcentowa** wartość 5%, oblicz odsetki dla następnego miesiąca i wypisz stan oszczędności dla obu rachunków.

2. Utwórz klasę **IntegerSet**. Każdy obiekt klasy **IntegerSet** może przechowywać liczby z przedziału od 1 do 100. Zbiór jest reprezentowany przez tablicę elementów typu **boolean**. Element tablicy o indeksie  $k$  ma wartość **true** jeżeli liczba  $k$  należy do zbioru oraz wartość **false** jeżeli liczba  $k$  nie należy do zbioru. Bezargumentowy konstruktor klasy **IntegerSet** tworzy pusty zbiór (t.j. zbiór reprezentowany przez tablicę której wszystkie elementy mają wartość **false**). Zdefiniuj w klasie **IntegerSet** następujące metody:

- statyczną metodę **union**, która tworzy sumę mnogościową dwóch zbiorów będących argumentami tej metody.
- statyczną metodę **intersection**, która tworzy iloczyn mnogościowy dwóch zbiorów będących argumentami tej metody.
- metodę **insertElement**, która dodaje liczbę będącą jej argumentem do zbioru, na rzecz którego ta metoda została wywołana.
- metodę **deleteElement**, która usuwa liczbę będącą jej argumentem ze zbioru, na rzecz którego ta metoda została wywołana.
- bezargumentową metodę **toString**, która zwraca łańcuch zawierający oddzielone spacjami liczby ze zbioru, na rzecz którego ta metoda została wywołana.
- metodę **equals**, która sprawdza czy zbiór będący jej argumentem jest identyczny ze zbiorem na rzecz którego ta metoda została wywołana.

Zapoznaj się z dokumentacją metod **toString** oraz **equals** z klasy **Object**.

Napisz funkcję **main**, która sprawdzi czy powyższe metody poprawnie działają.

3. Zmodyfikuj program z wykładu o nazwie **PracownikDemo1.java** poprzez zamianę typu pola **dataZatrudnienia** z **Date** na **LocalDate** oraz dokonaj wynikających z tej zmiany koniecznych modyfikacji.