

## Programowanie obiektowe w Javie (POJ)

### Laboratorium 6

**Zadanie 1.** Utwórz klasę Pracownik o polach imię, nazwisko, dzienne wynagrodzenie. A następnie:

1. Utwórz klasę Firma o polu pracownicy[] oraz metodach:
  - a. obliczMiesiecznyKosztFirmy(),
  - b. obliczRocznyKosztFirmy()
2. Rozszerz projekt o typ wyliczeniowy – Czas o wartościach DZIEŃ, MIESIĄC (20 dni), ROK
3. Dodaj metodę obliczKosztFirmy(Czas, int krotność):double do klasy Firma
  - a. Np. obliczKosztFirmy(Czas.MIESIAC, 6) – zwróci koszt działalności firmy przez 6 miesięcy
4. Przetestuj swoje rozwiązanie

**Zadanie 2. (8 pkt)** Napisać program do obsługi zamówień. Program powinien składać się z klas Zamowienie oraz Pozycja.

1. Zamowienie ma :
  - a. pole pozycje (Pozycje[]), maksRozmiar
  - b. Oraz konstruktor(maxRozmiar) i konstruktor bezargumentowy (gdzie maksRozmiar=10)
  - c. A także metody dodajPozycje(Pozycja) i obliczWartosc():double,
  - d. toString() zwraca spis pozycji zamówienia.
2. Pozycja posiada :
  - a. 3 pola: nazwaTowaru, iloscSztuk, cenaSztuki,
  - b. oraz metody: obliczWartosc() i toString()

Przykładowy wynik działania programu:

Zamówienie:

Chleb	2.00 zł	2 szt.	4.00 zł
Banany	6.00zł	1 szt.	6.00 zł

Razem: 10.00 zł

**Zadanie 3.** Utwórz własny typ daty np. MojaData, który przyjmuje 3 argumenty typu int dla określenia dni, miesięcy i lat. Nadpisz metodę toString(), który zwróci datę w następujący sposób:

- a. 1.11.2011
- b. 01.11.2011
- c. 1 lis 2011

*Dla każdej z wersji przetestuj działanie – na koniec zamieść kod z zakomentowanymi wersjami a i b i odkomentowaną wersją c.*

**Zadanie 4.** Napisać klasę Czas do przechowywania czasu w godzinach i minutach.

- a. Pola:

- i. Int godziny
- ii. Int minuty
- b. Metody:
  - i. Konstruktor(godziny, minuty)
  - ii. toString():String → np. „29h 14 min”
  - iii. dodaj(Czas t):Czas → zwróci nowy obiekt typu Czas będący sumą aktualnego czasu i podanego w argumencie metody
  - iv. odejmij(Czas t): Czas → analogicznie jak w dodawaniu
  - v. pomnóż(int krotnosc):Czas → okres czasu pomnożony przez podaną liczbę

**Zadanie 5.** Utwórz klasę Samochod o polach predkosc, kolor, marka, rocznik. Niech prędkość będzie domyślnie równa 0 dla nowo utworzonego auta (nie trzeba jej podawać w konstruktorze). Następnie :

- a. Utwórz metodę przyspiesz(), która rozpędzi auto o 20 km/h ale do max 140 km/h
- b. nadpisz metodę .toString() tak aby wyświetlić opis auta (np. „Czerwone BMW rocznik 2000”)
- c. nadpisz metodę equals(), która zwróci true jeśli kolor, marka i rocznik będą takie same
- d. Przetestuj powyższe rozwiązania
- e. Utwórz podklasę SzybkiSamochod
- f. Nadpisz metodę przyspiesz() aby osiągać max 200 km/h
- g. Utwórz obiekt SzybkiSamochod o tych samych parametrach co Samochod – czy są sobie równe ?
- h. Jeśli tak, to nadpisz .equals() dla SzybkiegoSamochodu, tak aby Samochod!=SzybkiSamochod

**Zadanie 6.** Utwórz metodę przyjmującą dowolną ilość argumentów typu String oraz pojedynczy argument typu int (pamiętaj o zachowaniu odpowiedniej kolejności w deklaracji i zwracającą wynik w postaci String.

Niech argument typu int definiuje co n-te elementy, które zostaną usunięte np. przy użyciu argumentu int z wartością 2 spodziewamy się, że metoda usunie co drugie słowo z zadanego tekstu, tj :

„Ala”, „nie”, „ma”, „test” , kota i”, „psa”, „papugę” 2 -> „Ala ma kota i papugę” // 2-gie i czwarte argumenty zostały usunięte

**Zadanie 7. (4 pkt)** Napisz prostą metodę do sumowania tablic – przyjmującą dowolną ilość argumentów typu tablic całkowitych i zwracającą sumę wprowadzonych liczb w postaci liczby całkowitej.

Rozwiązanie zadań powinno być udostępnione w na indywidualnym repozytorium github studenta maksymalnie do końca czwartego dnia po zajęciach:

- zajęcia w poniedziałek -> termin do piątku 23:59:59
- zajęcia w środę -> termin do niedzieli 23:59:59
- zajęcia w czwartek -> termin do poniedziałku 23:59:59

Zadania powinny być zrealizowane w osobnych klasach w ramach jednego projektu, a umieszczony w **prywatnym** repozytorium, które jest udostępnione prowadzącemu (<https://github.com/plucins>)