

## Paradygmaty programowania - ćwiczenia

### Lista 11

**Wszystkie programy mają być napisane w języku Scala.**

1. Zdefiniuj funkcję `wordCounter`: `(text: String) scala.collection.mutable.Map[String,Int]`, która zlicza wystąpienia słów w zadanym tekście. Zakładamy, że słowa są oddzielone od siebie spacjami. Wynik ma być słownikiem modyfikowalnym, w którym słowa są kluczami, a liczby wystąpień słów są wartościami. Postaraj się zrobić to jak najprościej.

2. Zdefiniuj swoją klasę dla pary polimorficznej. Ma ona mieć dwa pola modyfikowalne `fst` i `snd` z odpowiednimi akcesorami, oraz metodę `toString`, zwracającą napis w formacie `(fst, snd)`.

3. Zdefiniuj drugą klasę dla modyfikowalnej pary polimorficznej z taką samą funkcjonalnością, jak w zadaniu 2. Zrób to, wykorzystując wyłącznie składowe abstrakcyjne. Utwórz przykładowy egzemplarz tej klasy.

4. Zdefiniuj rodzinę klas `Point`, `Circle`, `Cylinder`, dziedziczących po sobie. Wszystkie argumenty konstruktorów klas mają mieć wartości domyślne. Metody, zmieniające wartości pól klasy, mają zwracać zmodyfikowany obiekt, co umożliwi kaskadowe wywoływanie metod.

a) Klasa `Point` – definiuje położenie punktu na płaszczyźnie.

b) Klasa `Circle` – definiuje położenia środka koła oraz promień.

c) Klasa `Cylinder` – definiuje dodatkowo wysokość walca.

Specyfikacja celowo pozostawia pewną swobodę implementacji. Należy wybrać najlepsze rozwiązanie, ewentualnie przedyskutować alternatywy.

5. Zdefiniuj klasę `Pracownik`. Każdy pracownik ma nazwisko ( z odpowiednimi akcesorami), oraz atrybut `zwolniony` (`true/false`). Ponadto należy pamiętać liczbę pracowników jako prywatny atrybut klasy. Utworzenie pracownika ma zwiększać ten licznik, a metoda `zwolnij` zmniejsza licznik pracowników i zmienia atrybut `zwolniony` danego pracownika na `true`. Metoda `toString` zwraca nazwisko pracownika oraz informację o tym, czy został zwolniony. Metoda `liczbaPracowników` zwraca aktualną liczbę pracowników.