Lista 9

Programowanie obiektowe II - Scala

Podczas realizacji zadań należy pamiętać o podstawowych zasadach tworzenia kodu obiektowego. Oznacza to **adekwatne** wykorzystywanie modyfikatorów dostępu, getterów oraz seterów, właściwości, klas abstrakcyjnych oraz cech.

Należy wykorzystywać fakt, że **argumenty konstruktora głównego** w Scali mogą od razu stawać się **polami** danej klasy.

Każde zadanie musi posiadać **kompletny zestaw testów**.

Do wykonania zadań należy wykorzystać mechanizmy poznane na wykładzie nr 9.

Wskazówka:

> **Domieszki** (*ang. mix-in, mixin*) mogą być **dodawane** nie tylko do klas (patrz wykład), ale także **do obiektów podczas ich tworzenia** poprzez wykorzystanie słowa kluczowego ***with*** po przekazaniu argumentów konstruktora – powstaje wtedy pomocnicza klasa anonimowa dla danego obiektu:
>
> *val objectWithMixin = new MyClass()* ***with*** *MyMixin*

1) Zdefiniuj klasę *Rectangle* reprezentującą prostokąt o wymiarach a i b. Wymiary powinny być polami tylko do odczytu. Klasa powinna umożliwiać tworzenie prostokąta o wymiarach a x b oraz kwadratu o boku długości a. Utworzyć właściwość zwracającą pole prostokąta. (Scala) (5 pkt)

2) Wykorzystując mechanizmy języka Scala zamodeluj poniższy wycinek rzeczywistości (Scala) (15 pkt)

   „W kraju pracuje wielu wolnych fachowców. Podczas rejestracji w urzędzie miejskim, fachowiec musi podać swoje imię, nazwisko oraz wiek. Każdy fachowiec posiada specyficzny zestaw umiejętności np. kładzenie kafelków, malowanie ścian, naprawa instalacji elektrycznej itp. Oczywiście fachowiec może potrafić wykonywać wiele czynności, a różni fachowcy mogą potrafić zrobić dokładnie to samo."

   Implementacja musi:
   - Zawierać klasę *Handyman* reprezentującą fachowca,
   - Imię oraz nazwisko nie mogą być puste, a wiek nie może być mniejszy niż 18 lat – zgłosić odpowiednie wyjątki,
   - Różne rodzaje fachowców np. elektrycy, murarze, hydraulicy itp. potrafią wykonywać różne czynności – w postaci metod np. paint(), fixElectricity(), … wyświetlających komunikat na ekranie,
   - Nie twórz jawnie podklas fachowców uwzględniających różne kombinacje umiejętności,
   - Każdy rodzaj fachowca powinien umożliwiać dowiedzenie się ilu fachowców danego rodzaju istnieje – poprzez wywołanie odpowiedniej metody,