

# Zaawansowane Programowanie Obiektowe i Funkcyjne

## Refleksje i adnotacje

Zadanie oceniane nr 5b

5-12-2022

Kod wstępny znajduje się w katalogu `zpoif_zadanie_5b`. Należy zsynchronizować się za pomocą polecenia `git pull`. Po zakończeniu pracy konieczne jest wgranie zmian do repozytorium (`commit + push`). Proszę pamiętać o poleceniu `add`.

## "Autobus"



W dniu dzisiejszym poeksperymentujemy nieco z mechanizmem adnotacji, za pomocą którego niczym dobrze znany, lubiany i szeroko stosowany backendowy szablon "Spring framework", będziemy oznaczać elementy klas, skanować je i coś z nimi robić aby finalnie zappełnić nasz pojazd.

Poniższe punkty zadania związane z poszukiwaniem adnotacji w kodzie i reakcją na nie należy wykonać za pomocą mechanizmu refleksji dostarczonego przez standardową bibliotekę Java. **Nie wolno** używać operatora **"new"**, **"instanceof"** ani **niczego zmieniać** w pakiecie:

`pl.edu.pw.mini.zpoif.task5` z wyjątkiem podpakietu `"solution"` (chyba że podpunkt zadania będzie stanowił inaczej). **Można stosować casting typów**. Działanie rozwiązań należy zaprezentować w klasie `Demonstrator`, wraz z odpowiednimi informacjami wypisywanymi na konsolę.

### Prace do wykonania:

1. Utworzyć następujące adnotacje widoczne podczas działania programu:
  - a) `InitiateObject`, która:
    - przeznaczona jest tylko dla klas
    - zawiera parametr o nazwie `"ilosc"` (domyślnie 1)
  - b) `MethodInvoker`
    - przeznaczona jest dla metod
    - zawiera parametr o nazwie `"licznikWywołan"`, bez wartości domyślnej
    - w/w parametrowi można przyporządkować tylko takie wartości: 1,3,5
  - c) `MyConstructor`, która:
    - przeznaczona jest dla konstruktora

- d) RandomIntInitiator
    - przeznaczona jest dla parametrów (metody, konstruktora)
    - zawiera parametry int "od" i "do", bez wartości domyślnej
  - e) AddMulti
    - przeznaczona dla pól
    - zawiera parametr typu Class bez wartości domyślnej, który mówi nam jakich klas obiekty chcemy dodawać do kolekcji (tą adnotacją będziemy oznaczać kolekcje)
2. Oznaczyć:
- a) adnotacją "InitiateObject" klasy: "Autobus" (2), "DoroslyPasazer" (30), "Czlowiek" (1), "Bagaz" (15), "Dziecko" (13), "Kierowca" (5)
  - b) adnotacją "MyConstructor" konstruktory klas:
    - Bagaz - bezparametrowy
    - Silnik z dwoma parametrami
    - Silnik z trzema parametrami
  - c) adnotacją "RandomIntInitiator" parametry konstruktorów klasy "Silnik":
    - dwuparametrowy (wszystkie parametry obstarwić adnotacją z wartością 50-100)
    - trzyparametrowy (tylko jeden parametr obstarwić adnotacją z wartością 50-100)
  - d) adnotacją "MethodInvoker" metody (należące do klasy Autobus):
    - toString() z wartością odpowiadającą jednemu wywołaniu.
    - sayHello() z wartością odpowiadającą jednemu wywołaniu
  - e) adnotacją "AddMulti" pola (należące dla klasy Autobus):
    - kierowcy (class->Kierowca.class), pasazerowie (class->DoroslyPasazer.class), bagaze (class->Bagaz.class)

Przygotować kod który wykonuje poniższe podpunkty i umieścić go wedle uznania w jakiejś klasie. Zademonstrować działanie. **Ułatwienie:** do refleksyjnego skanowania klas można się posłużyć pomocniczną kolekcją klas w klasie Helper – zakładamy że innymi nie będziemy się interesować.

3. Stworzyć kod, który automatycznie skanuje wszystkie klasy (z w/w pakietu lub z pomocą Helpera) w poszukiwaniu użytej w poprzednim punkcie adnotacji "InitiateObject". Na podstawie parametru "ilosc" tworzy tyle obiektów ile jest tam wyszczególnionych i je gdzieś zapamiętuje do późniejszego użycia. Należy mieć na uwadze że adnotacją "MyBean" ktoś może oznaczyć różne klasy więc obiekty będą różnych typów. Z oczywistych względów klasy abstrakcyjne ignorujemy. Po dodaniu wszystkich wykrytych ich kolejność pomieszać (shuffle).
4. Stworzyć kod, który poszukuje w już stworzonych obiektach adnotacji "MethodInvoker" i wywołuje je tyle razy ile jest wyszczególnione w parametrze. Uruchamiamy jeśli metoda nie przyjmuje parametrów.
5. Stworzyć kod, który poszukuje konstruktorów z adnotacją "MyConstructor". Jak już takowy znajdzie to sprawdza czy wszystkie parametry są obstarwione adnotacją RandomIntInitiator. Jeśli są, to tworzy obiekt przekazując do konstruktora losowe integerzy z zakresów podanych w poszczególnych adnotacjach parametrów. Utworzony obiekt zachowuje razem z dotychczasowymi instancjami.
6. Finał! Przeszukać już utworzone obiekty pod względem adnotacji (na polach) "AddMulti", ustalić typ tych pól i (jeśli to jest Lista lub Set) poszukać pasującego obiektów które można do nich wstawić (każdy jest instancją klasy z parametru adnotacji) wśród obiektów już utworzonych. Wstawiamy ich losową liczbę od 1 do 10. Wstawionych obiektów nie wstawiamy nigdzie indziej. W razie braku obiektu lub jeśli pole nie jest Listą lub Setem nie robimy nic.