Zaawansowane Programowanie Obiektowe i Funkcyjne
Wyrażenia lambda
Zadanie oceniane nr 3
14-11-2019

Dzisiaj będzie bez kodu wstępnego. Źródła powinny się znaleźć w Państwa katalogu roboczym (git) w podkatalogu Lab6-zadanie3. Po zakończeniu pracy konieczne jest wgranie zmian.

"Radiostacje"

W dniu dzisiejszym, wiodące tło tematyczne zadania orbitowało będzie w okolicach związanych urządzaniami komunikacji radiowej.

Za ładne rozmieszczenie klas po pakietach i prawidłową konwencję nazewniczą (nazwy klas z dużej litery, notacja wielbłądzia ogólny porządek w kodzie) będzie uznaniowy extra bonus punktowy.

Prace do wykonania:

1. Stworzyć jedną hierarchię klas reprezentującą min. możliwe do instancjonowania wymienione środki masowego przekazu:
   - Ręczne radio CB (nazwa producenta, kanał (1-40), częstotliwość w MHz (zależnie od kanału), modulacja (AM lub FM), wodoodporność (Tak/Nie))
   - Samochodowe radio CB (nazwa producenta, kanał (1-40), częstotliwość w MHz (zależnie od kanału), modulacja (AM lub FM), homologacja (Tak/Nie))
   - Stacjonarne radio CB (nazwa producenta, kanał (1-40), częstotliwość w MHz (zależnie od kanału), modulacja (AM lub FM))
   - Radiostacja krótkofalowa (nazwa producenta, częstotliwość w MHz (1-100), modulacja (AM lub FM) dostępna dla podklas)

   Częstotliwość(kanał) = 26,950 + nr kanału * 0,01

   Proszę pamiętać o:
   - konstruktorach
   - typach
   - widoczności pól i metod z zewnątrz (tylko to co potrzebne)
   - umożliwieniu dalszego dziedziczenia pól i metod, również przez klasy spoza pakietów

- poprawnym skonstruowaniu hierarchii klas
- dodatkowych rzeczach jeśli będą potrzebne

2. Stworzyć klasę Generator posiadającą metodę generującą listę radiostacji różnych typów. W celu stworzenia obiektów należy zdefiniować iterfejsy funkcyjne potrzebne tylko tutaj lub ew. w podklasach klasy: HandCBRadioGenerator , CarCBRadioGenerator, StationaryCBRadioGenerator i RadioGenerator.

   - HandCBRadioGenerator, CarCBRadioGenerator – posiada metodę abstrakcyjną pobierającą numer kanału na który ustawiona jest tworzona właśnie radiostacja.
   - StationaryCBRadioGenerator – posiada metodę abstrakcyjną pobierającą numer kanału oraz modulację.
   - RadioGenerator – posiada metodę abstrakcyjną bez parametrów

   Metoda generująca powinna:
   - Utworzyć implementacje interfejsów funkcyjnych:
     - ✔ z użyciem wyrażeń lambda
     - ✔ w których reszta danych potrzebnych do utworzenia obiektu, nie uwzględnionych w parametrach jest ustawiana losowo (jedynie częstotliwość jest wyliczaba wg kanału dla radiostacji CB).
     - ✔ które korzystają z instancji obiektu Random utworzonego poza ciałem wyrażenia lambda ale w ciele metody w której je utworzono
   - Uruchomić każdą z implementacji interfejsu losową liczbę razy (1-5), a wynik dodać do kolekcji i ją zwrócić

   Nazwy urządzeń powinny być losowo wybierane spośród przedstawionych poniżej wartości:

   CB samochodowe:
   ```
   "Cobra", "Zodiac", "President", "Uniden", "Midland", "Stryker"
   ```

   CB ręczne:
   ```
   "ALAN", "Motorola"
   ```

   Pozostałe:
   ```
   "STABO", "TEAMBSG", "RadMor", "Albrecht", "PNI_ESCORT", "CRTMIKE", "Galaxy",
   "PMR", "Midland"
   ```

3. Za pomocą wyrażenia lambda lub klasy anonimowej (tam gdzie to niezbędne) stworzyć implementację obiektów interfejsu java.util.function.Consumer w jakieś metodzie obiektu stworzonego celem demonstracji (wraz z uruchomieniem), które konsumują:
   - Stringa – wypisanie na konsoli informacji, jeżeli ciąg znaków kończy się na 'A'
   - Integera – wypisanie na konsoli wartości sin()
   - CB radio stacjonarne – losowe ustawienie kanału i wynikającej z niego częstotliwości (za pomocą metody pomocniczej np. set)
   - Radiostacja krótkofalowa – ustawienie częstotliwości na losową dopuszczalną i modulacji na FM

4. Utworzyć klasę zawierającą metodę do sortowania stworzonej listy obiektów medialnych. Sortowanie ma się odbywać z użyciem stworzonych obiektów Comparator zainicjowanych oczywiście z użyciem wyrażeń lambda:
    - (pierwsze sortowanie) sortowanie po nazwie
    - (drugie sortowanie) jeśli obiekt jest instancją klasy reprezentującej ręczne radio GB, to sortujemy po kanale

5. Zdefiniować za pomocą wyrażenia lambda dwa obiekty interfejsu java.util.function. Consumer, przekazane kolejno po jednym do dwóch wywołań metod forEach na kolekcji radiostacji.
   Implementacje te mają dla każdego obiektu kolekcji:
    - (pierwsza) Wyświetlić długości nazw producenta wraz z tą nazwą
    - (druga) Jeśli jest to samochodowa radiostacja CB, ustawia kanał (i wynikającą z niego częstotliwość) na wartość 19.

6. Stworzyć klasę ComplexFunction, posiadającą metodę getComplexFunction, która to zwraca listę trzech obiektów implementujących predefiniowany interfejs funkcyjny java.util.function.Function (dla tych, którzy dokonają tego w jednej linijce kodu – extra bonus punktowy). Każdy z nich powinien być zaimplementowany w w/w metodzie za pomocą wyrażenia lambda i reprezentować kolejno po jednej z trzech funkcji: $x + \sin(x)*x^3$, cosinus pierwiastka z wartości bezwzględnej x oraz liczba PI pomnożona przez losową liczbę z przedziału 1-10. Klasa posiada metodę calculateValue przyjmującą jakiś parametr, dla którego oblicza wartość funkcji złożonej, czyli nic innego jak sumę wartości zwracanych przez kolejno zwracane implementacje interfejsów funkcyjnych (oczywiście trzeba wywołać odpowiednią metodę z tego interfejsu). Suma powinna być policzona za pomocą forEach.