

Zaawansowane Programowanie Obiektowe i Funkcyjne

Wyrażenia lambda

Zadanie oceniane nr 3

13-11-2019

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.

"Mass media"



Zaistniała potrzeba szybkiego i sprawnego definiowania kodu związanego z przetwarzaniem kolekcji obiektów nawiązujących do tzw. środków masowego przekazu. Za ładne rozmieszczenie klas po pakietach i prawidłową konwencję nazewnictwa będzie uznaniowy extra bonus punktowy.

Prace do wykonania:

1. Stworzyć hierarchię klas reprezentującą wymienione środki masowego przekazu:
 - Radio (nazwa, renoma (1-100), ilość odbiorców (do 1 000 000 000), współczynnik prawdomówności (1-100))
 - Telewizja (nazwa, renoma (1-200), ilość odbiorców (do 500 000 000), współczynnik prawdomówności (1-100), czy jest to kanał informacyjny (tak/nie))
 - Gazeta (nazwa, renoma (1-220), nakład (do 100 000), ilość stron (do 100))

Proszę pamiętać o:

- konstruktorach
- typach
- widoczności pól i metod z zewnątrz (tylko to co potrzebne)
- umożliwieniu dalszego dziedziczenia pól i metod
- poprawnym skonstruowaniu hierarchii klas

2. Stworzyć klasę posiadającą metodę generującą listę mediów różnych typów.
W celu stworzenia obiektów należy zdefiniować interfejsy funkcyjne: `RadioGenerator`, `TVGenerator`, `NewspaperGenerator` które posiadają jakąś metodę zwracającą obiekt danego typu, stworzyć ich implementacje w metodzie generującej obiekty (za pomocą wyrażeń lambda) i użyć. Wszystkie wartości są losowane (wykorzystywany obiekt klasy `Random` powinien być pobrany spoza ciała wyrażenia lambda), natomiast nazwy mediów mają być losowo wybierane spośród takich wartości:

TV:

"CNNW", "FoxNews", "NBCC", "NHKK", "Jetix2", "Khabara TV", "UTVa"

Radio:

"Rouge FMa", "China National Radio New", "City FM 899", "Radio Hauraki2", "JFNK",
"Today FM news"

Gazeta:

"El Universal", "La Prensaco", "Tal Cualana", "Barbadian2", "Pepper PunchX"

3. Za pomocą wyrażenia lambda lub klasy anonimowej (tam gdzie to niezbędne) stworzyć implementację obiektów `java.util.function.Consumer`, które konsumują ciągi znaków (Stringi) w sposób zgodny z poniższym zamówieniem:
 - wypisanie informacji, jeżeli ciąg znaków zaczyna się na 'A'
 - wypisanie podanego ciągu znaków
 - wypisanie podanego ciągu znaków w odwrotnej kolejności
 - jeżeli ciąg znaków jest liczbą, to na konsoli pojawia się informacja o tym, czy jest parzysta, czy też nie. Powinna istnieć możliwość zapamiętania sumy liczb. Jeżeli ciągi znaków były liczbami to należy wypisać taką liczbę na konsolę za każdym razem jak liczba zostanie ona wykryta.
4. Utworzyć klasę zawierającą metodę do sortowania stworzonej listy obiektów medialnych. Sortowanie ma się odbywać z użyciem stworzonych obiektów `Comparator` zainicjowanych oczywiście z użyciem wyrażen Lambda:
 - sortowanie po typie obiektu (telewizje na końcu), reszta obojętnie
 - jeśli obiekt jest instancją klasy `Telewizja` bądź `Radio`, to sortujemy po ilości odbiorców
5. 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: $x^2 + x^3$, $\sin(x)$ oraz wartość bezwzględna z x . Klasa posiada metodę `calculateValue` przyjmującą jakiś parametr, dla którego oblicza wartość funkcji złożonej, czyli sumę wartości zwracanych przez iterowaną listę zwróconych interfejsów funkcyjnych (oczywiście trzeba wywołać odpowiednią metodę z interfejsu).