

Bootcamp Core Services

zadanie rekrutacyjne

Wstęp

Poniższe zadanie ma na celu sprawdzenie Twojej znajomości podstaw programowania, narzędzi oraz umiejętności ich wykorzystania.

Jednak, żeby ocena Twojej pracy była możliwa musisz przestrzegać kilku reguł:

1. Rozwiązanie piszemy w Javie 8 i nowszych (stabilnych).
2. Możesz pisać w Kotlinie, Scali, Clojure, albo innym języku na JVM.
3. Rozwiązanie dostarcz w postaci pliku zip, który będzie zawierać kod źródłowy, skrypt budujący (pliki `compile.bat` i `compile.sh`) oraz skrypt uruchomieniowy (plik `run.bat` i `run.sh`)
4. Budujemy za pomocą jednego, wskazanego przez ciebie, z narzędzi (najnowsze stabilne dystrybucje):
 - a. Maven,
 - b. Sbt,
 - c. Gradle.
5. Kompilacja będzie odbywać się na najnowszych wersjach JVM.

Przykładowe uruchomienie projektu w środowisku Linux:

```
$ compile.sh && ./run.sh [argumenty]
```

I windows

```
> compile.bat && run.bat [argumenty]
```

Możesz też dostarczyć nam rozwiązanie jako link do taga o nazwie solution w repozytorium git:

```
$ git clone <URL> && cd <REPO> && git checkout tags/solution &&  
compile.sh && ./run.sh
```

Zasady oceny są następujące:

1. Poprawność działania programu
2. Jakość kodu, która będzie mierzona w następujący sposób:
 - a. Pokrycie testami
 - b. Checkstyle z domyślnymi ustawieniami
3. Dokumentacja

Co dyskwalifikuje twoje rozwiązanie:

1. Program się nie kompiluje
2. Program nie uruchamia się

Zadanie

Program do obsługi zamówień.

1. Program na wejściu przyjmuje jako argument listę plików csv i xml.
2. Każdy plik zawiera jedno lub więcej zamówień (format w załączniku).
3. Każde zamówienie należy zapisać w „bazie danych” (użyj bazy in memory).
4. Zamówienie zawiera obowiązkowe pola:
 - a. ClientId – alfanumeryczne, bez spacji nie dłuższe niż 6 znaków,
 - b. RequestId – numeryczne long,
 - c. Name – alfanumeryczne ze spacjami nie dłuższe niż 255 znaków,
 - d. Quantity – numeryczne int,
 - e. Price – numeryczne zmiennoprzecinkowe podwójnej precyzji.
5. Program pozwala na wygenerowanie następujących raportów:
 - a. Ilość zamówień łącznie,
 - b. Ilość zamówień do klienta o wskazanym identyfikatorze,
 - c. Łączna kwota zamówień,
 - d. Łączna kwota zamówień do klienta o wskazanym identyfikatorze,
 - e. Lista wszystkich zamówień,
 - f. Lista zamówień do klienta o wskazanym identyfikatorze,
 - g. Średnia wartość zamówienia ,
 - h. Średnia wartość zamówienia do klienta o wskazanym identyfikatorze.
6. Każdy raport można wyświetlić na ekranie, albo zapisać do pliku csv (format nie jest narzucony).
7. Baza danych nie jest dzielona pomiędzy uruchomieniami.
8. Nieprawidłowe linie w zamówieniu są ignorowane, ale informacja o złym formacie jest wypisywana na ekran.

Załącznik – formaty

Plik CSV

Pierwsza linia zawiera nazwy kolumn i zawsze występuje, kolumny są oddzielone przecinkiem:

Client_Id,Request_id,Name,Quantity,Price

1,1,Bułka,1,10.00

1,1,Chleb,2,15.00

1,2,Chleb,5,15.00

2,1,Chleb,1,10.00

Plik XML

Plik zawsze jest poprawnym dokumentem xml. Nie zawiera dodatkowych tagów. Może brakować niektórych tagów w elemencie <request>.

```
<requests>
  <request>
    <clientId>1</clientId>
    <requestId>1</requestId>
    <name>Bułka</name>
    <quantity>1</quantity>
    <price>10.00</price>
  </request>
  <request>
    <clientId>1</clientId>
    <requestId>2</requestId>
    <name>Chleb</name>
    <quantity>2</quantity>
    <price>15.00</price>
  </request>
  <request>
    <clientId>1</clientId>
    <requestId>2</requestId>
    <name>Chleb</name>
    <quantity>5</quantity>
    <price>15.00</price>
  </request>
  <request>
    <clientId>2</clientId>
    <requestId>1</requestId>
    <name>Chleb</name>
    <quantity>1</quantity>
    <price>10.00</price>
  </request>
</requests>
```