

# 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 od wersji C# 5 w górę;
2. Możesz użyć .NET Frameworka lub .NET Core;
3. Rozwiązanie dostarcz w postaci pliku zip lub adresu do publicznego repozytorium GIT, który będzie zawierać kod źródłowy z plikiem projektu (nie załączaj folderów obj, bin, packages);
4. Kompilacja będzie odbywać się z poziomu Visual Studio lub konsoli (dotnet build);
5. Powinna zostać załączona instrukcja w jaki sposób korzystać z programu.

## Zasady oceny są następujące:

1. Poprawność działania programu
2. Jakość kodu, która będzie mierzona w następujący sposób:
  - a. Pokrycie testami
  - b. Czystość kodu
  - c. Obsługa błędów
  - d. Zastosowane wzorce projektowe
  - e. Wykorzystane struktury danych

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, xml i json
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 wyświetla menu z listą raportów oraz generuje wybrany raport:
  - a. Ilość zamówień
  - b. Ilość zamówień dla klienta o wskazanym identyfikatorze
  - c. Łączna kwota zamówień
  - d. Łączna kwota zamówień dla klienta o wskazanym identyfikatorze
  - e. Lista wszystkich zamówień
  - f. Lista zamówień dla klienta o wskazanym identyfikatorze
  - g. Średnia wartość zamówienia
  - h. Średnia wartość zamówienia dla klienta o wskazanym identyfikatorze
  - i. Ilość zamówień pogrupowanych po nazwie
  - j. Ilość zamówień pogrupowanych po nazwie dla klienta o wskazanym identyfikatorze
  - k. Zamówienia w podanym przedziale cenowym
6. Możliwość ustalenia w dowolnym momencie sortowania dla dowolnego raportu w trakcie działania programu
7. Każdy raport można wyświetlić na ekranie, albo zapisać do pliku csv (format nie jest narzucony)
8. Baza danych nie jest dzielona pomiędzy uruchomieniami
9. Nieprawidłowe dane 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
1,2,Chleb,2,15.00
1,2,Chleb,5,15.00
2,1,Chleb,1,10.00
```

### Plik XML

Plik jest 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>
```

## Plik JSON

Plik jest dokumentem json. Może brakować niektórych pól;

```
{
  "requests":[
    {
      "clientId":"1",
      "requestId":"1",
      "name":"Bułka",
      "quantity":"1",
      "price":"10.00"
    },
    {
      "clientId":"1",
      "requestId":"2",
      "name":"Chleb",
      "quantity":"2",
      "price":"15.00"
    },
    {
      "clientId":"1",
      "requestId":"2",
      "name":"Chleb",
      "quantity":"5",
      "price":"15.00"
    },
    {
      "clientId":"2",
      "requestId":"1",
      "name":"Chleb",
      "quantity":"1",
      "price":"10.00"
    }
  ]
}
```

**Q1: Czy interfejs użytkownika może być np. w formie aplikacji konsolowej?**  
**Mam pytanie, czy ma być to aplikacja konsolowa czy okienkowa czy też jest wolny wybór :)**

Sposób realizacji interfejsu użytkownika nie jest narzucony. Akceptowane są wszystkie rozwiązania, które umożliwią realizację założonej funkcjonalności.

**Q2: Ad 1. Program na wejściu przyjmuje jako argument listę plików csv, xml i json.**  
**Jaka to lista? Lista ze stringami zawierającymi ścieżki do plików na dysku?**  
**Lista ma być wprowadzona przez użytkownika, czy też należy ją skądś pobrać?**

Forma przekazania listy plików nie jest narzucona i zależy od przyjętej formy implementacji. Może to być na przykład:

- odwołanie do pliku tekstowego zawierającego listę plików do wczytania wraz z pełną ścieżką
- okno dialogowe umożliwiające zaznaczenie wielu plików do wczytania
- funkcjonalność drag & drop.

Sposób wczytania plików musi być opisany odpowiednio w dołączanej instrukcji.

**Q3: Ad 2. Każdy plik zawiera jedno lub więcej zamówień (format w załączniku)**  
**Jaka jest relacja request (request\_Id) do zamówienia?**  
**Czy każdy wpis to oddzielne zamówienie?**  
**Czy w przypadku csv kolejność kolumn może ulegać zmianom?**

Request\_Id odpowiada pojedynczemu zamówieniu. W związku z tym jeden wpis może odpowiadać całemu zamówieniu lub jedynie pozycji z zamówienia.

Struktura pliku CSV jest stała.

**Q4: Ad 6. Możliwość ustalenia w dowolnym momencie sortowania dla dowolnego raportu w trakcie działania programu.**  
**Czy na przykład użytkownik może ustalić sposób sortowania przed wygenerowaniem raportu?**  
**Sortowanie po ilościach, kwotach, średnich?**

Sposób realizacji tego wymagania nie jest narzucony, jednakże instrukcja powinna zawierać informację dla użytkownika o dostępnych sposobach sortowania danych w raporcie.

Wymaganie można zrealizować na przykład poprzez zdefiniowanie kilku rodzajów generacji raportu w aplikacji konsolowej lub umożliwienie dynamicznego sortowania wyników raportu w aplikacji „okienkowej”.

**Q5: Ad 8. Baza danych nie jest dzielona pomiędzy uruchomieniami.**  
**Czy to znaczy, że nie zapisujemy bazy danych z pamięci przed zakończeniem programu?**  
**Program rozpoczyna z pustą bazą danych dla każdego uruchomienia?**

Tak. Każdorazowo uruchomienie aplikacji wymaga załadowania danych wejściowych. Nie ma obowiązku zapisania raportów do plików na zakończenie działania aplikacji.

**Q6: Ad 9. Nieprawidłowe dane w zamówieniu są ignorowane, ale informacja o złym formacie jest wypisywana na ekran.  
Ignorowane to znaczy pomijane w obliczeniach niezbędnych do wygenerowania raportów?**

Nieprawidłowe dane w zamówieniu, ze względu na ich pominięcie w procesie wczytywania danych, nie są brane pod uwagę w części raportowej.

**Q7: Patrząc na przykładowe pliki nie do końca rozumiem strukturę "bazy danych". Wynika z nich, że konkretne zamówienie może być przypisane do dwóch klientów. Czy może kluczem w tej tabeli jest kombinacja obu pól - Client\_Id i Request\_Id?**

Kluczem w tabeli jest para Client\_Id i Request\_Id.

**Q8: 6. Możliwość ustalenia w dowolnym momencie sortowania dla dowolnego raportu w trakcie działania programu  
i. Ilość zamówień pogrupowanych po nazwie  
Prosiłbym o bardziej szczegółowe ich przedstawienie**

Dane zgrupowane po polu 'Name'

**Q9: Czy pokrycie testami i samo testowanie jest wymogiem kluczowym?**

Przygotowanie testów nie jest wymogiem kluczowym, ale zgodnie z opisem zadania pokrycie testami wpływa na ocenę wykonania zadania.

**Q10: Czy jeden plik csv, xml albo json jest jednym raportem? Czy to dopiero na ich podstawie są generowane raporty?**

Każdy plik zawiera dane, które są wykorzystywane do wygenerowania raportu końcowego.

**Q11: Czy użytkownik wprowadza listę plików czy program ma być uruchamiany z argumentem?**

Forma realizacji sposobu przekazania listy plików jest sprawą otwartą. Jednakże sposób przekazania parametrów / plików ma być dobrze opisany w załączonej dokumentacji.

**Q12: Czym jest baza in memory? Zapytania LINQ do jakiejś kolekcji? Czy należy użyć in-memory SQLite?**

Zapamiętanie informacji w ramach uruchomionej aplikacji. Nie ma konieczności zapisywania danych w typowej bazie danych czy pliku na dysku.

**Q13: Czy rozwiązanie należy utworzyć w np. Windows Forms albo WPF albo ASP.NET czy ma być to zwykła aplikacja konsolowa?**

Sposób realizacji zadania nie jest narzucony i wybór technologii i formy jego rozwiązania pozostawiamy kandydatom.

**Q14:** Czy menu z listą raportów to jest ten zestaw opcji od "a" do "k" w punkcie 5? Czyli przykładowo użytkownik wciśnie klawisz "a" na klawiaturze i dostanie informację o ilości wszystkich zamówień, tak?

Zgadza się. Opcje „a” do „k” to lista raportów, których wygenerowanie ma umożliwić aplikacja. Jednakże sposób implementacji nie jest narzucony i kandydat może zdecydować się na dowolną formę jego realizacji. Jednakże sposób generowania raportów ma zostać opisany w załączonej dokumentacji.

**Q15:** Po czym mają być sortowane zamówienia w danym raporcie? Nazwa produktu, ilość czy cena? Czy wszystko jednocześnie? A jeżeli raport to zgodnie z podpunktem "a" - ilość zamówień, czyli liczba - to jak ma być ta liczba posortowana, skoro jest jedna?

Jeżeli raport zawiera dane, których nie można posortować, sortowanie może nie zostać zaimplementowane.

**Q16:** Wyświetlenie raportu ma się odbywać w konsoli, czy właśnie np. w jakimś textbox w WS? I czy chodzi tu o wyświetlanie tych raportów z opcji od "a" do "k" z punktu 5 (np. dla "a" - "Liczba zamówień: 25") czy całego zestawu zamówień z jednego pliku - z nazwami produktów, cenami itd.?

Sposób prezentacji wyników nie jest narzucony i pozostaje do wyboru kandydatowi. Raport ma bazować na danych zaczytanych z plików przekazanych jako parametr wejściowy.

**Q17:** Czy przesyłamy jedynie zawartość pliku bin czyli plik exe oraz potrzebne załączniki typu plik xml, csv i json oraz plik .sln jako kod źródłowy?

Należy przesłać komplet plików koniecznych do poprawnego skompilowania i uruchomienia programu wraz z instrukcją obsługi.

**Q17:** Czy menu z listą raportów to jest ten zestaw opcji od "a" do "k" w punkcie 5? Czyli przykładowo użytkownik wciśnie klawisz "a" na klawiaturze i dostanie informację o ilości wszystkich zamówień, np. Liczba zamówień: 25. I to jest cały raport, tak?

Jeżeli z opisu raportu wynika, że ma zwrócić jeden wiersz, to jest to zachowanie poprawne.

**Q17:** Jeżeli raport to zgodnie z np. podpunktem "a" - ilość zamówień, czyli liczba - to jak ma być ta liczba posortowana, skoro jest jedna? Czy sortowanie ma się odnosić do niektórych raportów? Co nie będzie zgodne z punktem 6.

Jeżeli raport zwraca jeden wiersz, to możliwość sortowania nie wnosi wartości biznesowej, więc może być pominięty. Odpowiednia informacja może być częścią instrukcji obsługi.

**Q18:** Czy w poniższym przypadku to będzie jedno zamówienie o dwóch "Produktach", czyli w sumie zamówień liczone jako jedno?

```
<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>

```

Tak. Ponieważ clientId i requestId jest zgodne, obie pozycje liczone są jako dwie pozycje z jednego zamówienia.

**Q19: Czy całkowita wartość "Produktu" to będzie "quantity" \* "price" czy po prostu "price"?**

Wartość pojedynczego produktu prezentuje parametr 'price'.

**Q20: „g. Średnia wartość zamówienia”. Czy średnia wartość zamówienia ma dotyczyć jednego zamówienia czy raczej chodziło o średnią wszystkich zamówień?**

Średnia wartość zamówienia odnosi się do średniej obliczonej dla wszystkich załadowanych zamówień.

**Q21: "i. Ilość zamówień pogrupowanych po nazwie". Chodziło tutaj o listę zamówień? I jakiej nazwie mają być pogrupowane?**

Ilość zamówień dla danego produktu w podziale na nazwy produktów.

**Q22: Czy opisana w punkcie pierwszym lista plików to trzy pliki (o rozszerzeniach json, xml i csv), czy może być ich więcej?**

Przekazywana jako parametr lista plików to lista wszystkich plików, z których aplikacja ma zaczytać dane. Plik z danymi może być jednym z 3 podanych rodzajów CSV, XML i JSON.

**Q23: Mam pytanie odnośnie pola "price", mianowicie czy jest to cena jednostkowa którą należy przemnożyć przez quantity, czy może cena wszystkich produktów zawartych w quantity.**

Parametr 'price' jest to cena jednostkowa danego produktu.