Zadanie 8

Pieniądze... wielkie pieniądze

CYBERCOM POLAND

Opis

Jeden z klientów - dalekowschodni *Kupakasi Bank*, zwrócił się do nas z prośbą o dostarczenie oprogramowania do bankomatów. Ich dotychczasowy dostawca *Mihapalecimharata Engineering* dostarczył większość komponentów, jednak zawiedli przy module obliczającym w jakich banknotach należy wypłacić żądaną kwotę. Dostarczone oprogramowanie nie przeszło testów akceptacyjnych banku. Ponieważ termin wdrożenia narzucony przez prezesa *Tukosi-takasa* zbliża się nieubłaganie, zwrócili się do nas o dostarczenie brakującego modułu. I dlatego my zwracamy się do Ciebie o pomoc. Do dyspozycji dostaliśmy API, jakiego będzie używał brakujący komponent. Na szczęście szef działu IT banku *Nowego-malaptoka* dostarczył nam je w kilku językach oprogramowania, tak więc masz swobodę wyboru.

Opis zadania

Cel

Twoim zadaniem jest zaimplementowanie API zdefiniowanego przez interfejs/klasę abstrakcyjną *AtmWithdrawalCalculationService*. Na wejściu mamy kwotę pieniędzy jaką klient chce wypłacić, na wyjściu moduł ma zwrócić informację, czy wypłata jest możliwa oraz jakimi nominałami. Przykład: na wejściu podana jest kwota 110 zł, i na wyjściu ma pojawić się informacja, że wypłata jest możliwa w banknotach 3 razy po 20 zł oraz raz 50 zł. Oczywiście, Twoja implementacja może i powinna sprawdzić jakimi zasobami dysponuje bankomat. Do tego służy serwis wstrzyknięty przez pole *atmResourcesInfoService* do Twojej implementacji zawartej w klasie *AtmWithdrawalCalculationServiceImpl*.

Jeżeli wypłata jest niemożliwa, chcielibyśmy, byś zasugerował kwoty najbliższe żądanej (**jedna większa, jedna mniejsza**), przy których wypłata jest możliwa. Szczegóły w dokumentacji kodu.

Wejście

Kod w jednym z języków Java/C++/C# .

- C++

  mastercoder.pl/download/zad8/c/atm.zip

- C#

  mastercoder.pl/download/zad8/csharp/atm.zip

- Java

  mastercoder.pl/download/zad8/java/atm.zip

Mały przewodnik po kodzie, czyli krótki opis odpowiedzialności poszczególnych typów (staraliśmy się również te informacje zawrzeć w komentarzach):

AtmWithdrawalCalculationService	Definiuje API serwisu, którego zaimplementowanie jest celem zadania. Jego odpowiedzialnością jest odpowiedź na pytanie, czy wypłata danej kwoty jest możliwa przy użyciu zasobów bankomatu, w jakich banknotach a jeśli nie, to jakie są najbliższe kwoty, w których wypłata jest możliwa.
Money	Reprezentuje informację o ilości pieniędzy.
WithdrawalInfo	Reprezentuje informację czy wypłata jest możliwa, w jakiej kwocie (jeśli możliwa), oraz jakie są sugerowane kwoty wypłaty (jeśli niemożliwa).
AtmResourcesInfoService	Definiuje API do serwisu zewnętrznego, który zwraca informację o zasobach bankomatu. Innymi słowy, mówi ile banknotów w danym nominale posiada bankomat.
BanknoteBundle	Reprezentuje informację o ilości pieniędzy w różnych nominałach
DenominationBundle	Reprezentuje informację o ilości banknotów w danym nominale
Denomination	Reprezentuje informację o wielkości nominału
SuggestedWithdrawal	Reprezentuje informację o sugerowanych kwotach wypłaty (mniejsza i większa kwota).

Rozwiązanie

Oczekujemy działającej implementacji oraz zestawu testów jednostkowych, które ją testują. Możesz tworzyć własne klasy w ramach modułu implementacyjnego. Oceniamy nie tylko poprawność rozwiązania, ale również czystość kodu, dlatego pamiętaj o dobrych praktykach takich jak np. zasady SOLID.

W rozwiązaniu polecamy zwrócić uwagę na złożoność obliczeniową. Nie będzie to główne kryterium oceny, ale bez wątpienia będziemy najniżej punktować rozwiązania polegające na popularnie zwanej metodzie *„brute force”*. Autorów najciekawszych pomysłów nagrodzimy dodatkowymi punktami.

Zwracamy uwagę na to, że nie dostarczamy implementacji interfejsu/klasy abstrakcyjnej *AtmResourcesInfoService*. Docenimy jednak znajomość technik pozwalających mimo to przetestować Waszą implementację. Można w tym celu użyć bibliotek zewnętrznych o ile ich licencja nie ogranicza swobodnego i bezpłatnego ich użycia w naszym konkursie.

Ważne terminy

- Publikacja zadania: 15-04-2014
- Ostateczny termin nadsyłania odpowiedzi: 20-04-2014 godz.: 23:59
- Ogłoszenie wyników: 28-04-2014

Ocenianie

Zadanie będzie oceniane według poniższych kryteriów:

- Poprawna implementacja ilości zwracanych banknotów – do **150** punktów
- Poprawna implementacja sugerowanych kwot wypłaty, w przypadku braku możliwości zrealizowania pierwotnej wypłaty – do **150** punktów
- EXTRA bonus za rzeczy które nas pozytywnie zaskoczą – do **150** punktów, w tym
    - Testy jednostkowe (do **50** punktów)
    - Osiągnięcie najniższej złożoności obliczeniowej wśród wszystkich uczestników (50 punktów)
    - Premia za wyjątkowo dobrą jakość kodu (do **50** punktów)

Maksymalna ilość punktów do uzyskania w zadaniu: **450 pkt.**

Złote zasady

- przed wysłaniem sprawdź, czy kod się kompiluje
- nie zmieniaj API
- źródła przesyłaj jako załącznik do e-maila skompresowany za pomocą ZIPa, 7ZIP bądź RAR - możliwe też jest umieszczenie rozwiązania jako publicznego zasobu np.: na OneDrive, GoogleDrive
- swoje rozwiązanie prześlij na: **mastercoder.poland@cybercom.com**
- w temacie e-maila z odpowiedzią podaj język, technologię za pomocą której problem został rozwiązany, przykładowo: [Re: MasterCoder Zadanie 8 Pieniądze - rozwiązanie C#]
- w razie pytań pisz na wyżej wymieniony adres e-mail
- dobrze się baw!!!