

Kantor wymiany walut

Treść zadania

Celem zadania jest stworzenie symulacji działania kantoru wymiany walut. Kantor w naszym zadaniu będzie miał pewien zasób każdej z 7 walut z klasy `Currency`. Każdy nowo powstały kantor ma tyle samo każdej z nich. Ponadto, będzie miał 2 metody:

- `void showFunds()`, która wypisuje dostępne środki dla wszystkich walut.
- `void exchange(double amount, Currency from, Currency to)`, która odpowiada za faktyczną wymianę: kantor wymienia `amount` waluty `from` na odpowiednią ilość waluty `to`. Jeszcze przed samą transakcją pobiera prowizję w wysokości 20%, czyli de facto wymienia tylko $0.8 * \text{amount}$.

Jako że kursy walut są rzeczą, która zmienia się bardzo dynamicznie, będziemy łączyć się z Internetem, by pobierać możliwie aktualne. W tym celu:

1. Dodaj do projektu 2 dostarczone klasy: `CurrencyConverter` oraz `Currency`. Pamiętaj o zmianie paczki.
2. Stwórz klasę `ExchangeOffice` z funkcjonalnościami opisanymi wyżej. Sprawdzenie kursu danej waluty jest dostępne przez statyczną metodę `CurrencyConverter.convert(Currency from, Currency to)`.
3. Stwórz kantor z wartością początkową 1000 i dokonaj 3 wymian:
 - 1000 PLN → EUR
 - 250 PLN → RUB
 - 500 JPY → USD

Output w dniu 9.02.2019:

```
Exchanging 1000.0 PLN to 185.3744 EUR
Current funds: 1000.0 GBP 1000.0 JPY 814.6256 EUR 1000.0 CZK 1000.0 USD 2000.0 PLN 1000.0 RUB

Cannot exchange 250.0 PLN to 3438.7034 RUB : insufficient funds.
Current funds: 1000.0 GBP 1000.0 JPY 814.6256 EUR 1000.0 CZK 1000.0 USD 2000.0 PLN 1000.0 RUB

Exchanging 500.0 JPY to 3.6452 USD
Current funds: 1000.0 GBP 1500.0 JPY 814.6256 EUR 1000.0 CZK 996.3548 USD 2000.0 PLN 1000.0 RUB
```

Uwagi

1. Należy zadbać o podstawowe zasady programowania obiektowego - enkapsulację, gettery, settery, odpowiednią dostępność zmiennych... Te rzeczy nie są uwzględnione w treści zadania, ale są to dobre praktyki, które należy stosować.