

RAPORT ZADANIE3

Autor: Grzegorz Guz

Dsw54400

Biblioteki:

1. os: Używana do interakcji z systemem operacyjnym, w tym przypadku do pobierania zmiennych środowiskowych.
2. requests: Używana do wysyłania zapytań HTTP, w tym przypadku do pobierania danych o kursach walut z API NBP.
3. json: Używana do parsowania odpowiedzi z API NBP, które są w formacie JSON.
4. datetime: Używana do manipulacji i formatowania dat.
5. time: Używana do manipulacji czasem.
6. pandas: Używana do manipulacji danymi i analizy danych.

Opis funkcji:

1. Invoice i Payment: Są to klasy reprezentujące faktury i płatności. Każda z nich ma numer faktury, kwotę, walutę i datę.
2. pobierz_kurs: Ta funkcja pobiera kurs waluty dla danej daty z API NBP. Jeśli waluta to PLN, zwraca 1.0. Jeśli nie można znaleźć kursu waluty dla danej daty, próbuje znaleźć kurs dla poprzedniego dnia.
3. calculate_difference i calculate_remaining_amount: Te funkcje obliczają różnicę kursową i pozostałą kwotę do zapłaty dla danej faktury i jej płatności.
4. wprowadz_walute i wprowadz_date: Te funkcje służą do wprowadzania waluty i daty przez użytkownika. Sprawdzają poprawność danych i zwracają wartość None, jeśli dane są niepoprawne.
5. pytanie_tak_nie: Ta funkcja służy do zadawania pytania tak/nie użytkownikowi.

Sprawdzanie poprawności danych: Program sprawdza poprawność wprowadzanych danych na kilka sposobów:

1. **Sprawdzanie waluty:** Program prosi użytkownika o wprowadzenie waluty, a następnie sprawdza, czy wprowadzona waluta jest jedną z dostępnych walut (PLN, USD, EUR, GBP). Jeśli nie, program prosi użytkownika o ponowne wprowadzenie waluty.

2. **Sprawdzanie daty:** Program prosi użytkownika o wprowadzenie daty w formacie YYYY-MM-DD. Jeśli wprowadzona data nie jest w poprawnym formacie, program prosi użytkownika o ponowne wprowadzenie daty.
3. **Sprawdzanie kwoty:** Program prosi użytkownika o wprowadzenie kwoty, a następnie sprawdza, czy wprowadzona kwota jest liczbą. Jeśli nie, program prosi użytkownika o ponowne wprowadzenie kwoty.
4. **Sprawdzanie odpowiedzi na pytanie tak/nie:** Program prosi użytkownika o odpowiedź na pytanie tak/nie, a następnie sprawdza, czy odpowiedź jest jednym z dozwolonych wyborów (tak, nie, t, n). Jeśli nie, program prosi użytkownika o ponowne wprowadzenie odpowiedzi.
5. **Sprawdzanie, czy kwota nie jest ujemna:** Program sprawdza, czy wprowadzona kwota jest dodatnia. Jeśli nie, program prosi użytkownika o ponowne wprowadzenie kwoty.
6. **Sprawdzanie, czy data płatności nie występuje przed datą faktury:** Program sprawdza, czy wprowadzona data płatności jest późniejsza niż data wystawienia faktury. Jeśli nie, program prosi użytkownika o ponowne wprowadzenie daty płatności.

Opis kroków:

1. **Inicjalizacja:** Program zaczyna od inicjalizacji list invoices i payments, które będą przechowywać obiekty faktur i płatności.
2. **Pętla główna:** Program wchodzi w główną pętlę, która będzie działać, dopóki użytkownik nie zdecyduje się zakończyć.
3. **Wprowadzanie faktury:** Użytkownik wprowadza numer faktury, datę wystawienia faktury, kwotę faktury i walutę faktury. Program sprawdza poprawność wprowadzonych danych i tworzy nowy obiekt faktury, który jest dodawany do listy invoices.
4. **Wprowadzanie płatności:** Następnie użytkownik wprowadza numer faktury, na którą ma być wykonana płatność, datę płatności, kwotę płatności i walutę płatności. Program sprawdza poprawność wprowadzonych danych i tworzy nowy obiekt płatności, który jest dodawany do listy payments.
5. **Obliczanie różnicy kursowej i pozostałej kwoty do zapłaty:** Po wprowadzeniu szczegółów płatności, program oblicza różnicę kursową i pozostałą kwotę do zapłaty dla każdej faktury. Różnica kursowa jest obliczana jako różnica między kursem waluty na dzień wystawienia faktury a kursem waluty na dzień płatności. Pozostała kwota do zapłaty jest obliczana jako różnica między kwotą faktury a sumą wpłat.
6. **Pytanie o kolejną fakturę:** Na koniec program pyta użytkownika, czy chce wprowadzić kolejną fakturę. Jeśli użytkownik odpowie twierdząco,

program wraca do kroku 3. W przeciwnym razie program kończy działanie.

Pobieranie kursów z API:

1. Program wywołuje funkcję `pobierz_kurs`, przekazując datę i walutę, dla której chce uzyskać kurs.
2. Jeśli waluta to PLN, funkcja zwraca 1.0, ponieważ kurs PLN do PLN jest zawsze równy 1.
3. W przeciwnym razie, program próbuje pobrać kurs waluty z API Narodowego Banku Polskiego (NBP).
4. Jeśli nie można pobrać kursu waluty dla danej daty (na przykład, jeśli API NBP jest niedostępne lub zwraca błąd), program próbuje pobrać kurs dla poprzedniego dnia. Program próbuje pobrać kurs waluty maksymalnie trzy razy, cofając się o jeden dzień za każdym razem.

Zapis/odczyt do xlsx:

1. **Odczyt z plików Excel:** Na początku działania programu, funkcja `odczytaj_z_excel` jest wywoływana, aby odczytać dane z plików 'faktury.xlsx' i 'platnosci.xlsx'. Te pliki zawierają informacje o fakturach i płatnościach, które zostały zapisane podczas poprzednich uruchomień programu. Funkcja `odczytaj_z_excel` używa funkcji `read_excel` z biblioteki `pandas` do odczytania danych z plików Excel i przekształcenia ich w obiekty `DataFrame`. Następnie, dla każdego wiersza w `DataFrame`, tworzy obiekt `Invoice` lub `Payment` i dodaje go do odpowiedniej listy.
2. **Zapis do plików Excel:** Na końcu działania programu, po wprowadzeniu wszystkich faktur i płatności, funkcja `zapisz_do_excel` jest wywoływana, aby zapisać te dane do plików 'faktury.xlsx' i 'platnosci.xlsx'. Funkcja `zapisz_do_excel` najpierw tworzy obiekty `DataFrame` z list faktur i płatności, a następnie używa funkcji `to_excel` z biblioteki `pandas`, aby zapisać te `DataFrame` do plików Excel.
3. **Obsługa błędów:** Jeśli pliki 'faktury.xlsx' i 'platnosci.xlsx' nie istnieją w momencie odczytu, program tworzy puste listy faktur i płatności. Jeśli pliki nie istnieją w momencie zapisu, funkcja `to_excel` automatycznie tworzy te pliki.
4. **Zależności:** Aby móc pracować z plikami Excela, program musi mieć zainstalowane biblioteki `pandas` i `openpyxl`. Biblioteka `pandas` jest

używana do manipulacji danymi, a openpyxl jest potrzebna do obsługi plików Excela.

5. **Bezpieczeństwo danych:** Jeśli pliki 'faktury.xlsx' i 'platnosci.xlsx' istnieją przed uruchomieniem programu, ich zawartość zostanie nadpisana podczas zapisu. Dlatego zaleca się wykonanie kopii zapasowej tych plików przed uruchomieniem programu, jeśli chcesz zachować poprzednie dane.

```
Podaj numer faktury: 666
Podaj datę wystawienia faktury (YYYY-MM-DD): 2023-12-31
Podaj datę płatności (YYYY-MM-DD): 2024-02-07
Podaj kwotę zapłaty: 2000
Podaj walutę (PLN, USD, EUR, GBP): pln
Podaj kwotę faktury: 320
Podaj walutę (PLN, USD, EUR, GBP): usd
Błąd podczas pobierania kursu waluty, szukam dzień wcześniej
Błąd podczas pobierania kursu waluty, szukam dzień wcześniej
Kurs wymiany dla dnia wystawienia faktury: 3.8614
Kurs wymiany dla dnia płatności: 4.0061
Różnica kursowa dla nowej faktury numer 666: 0.1447
Błąd podczas pobierania kursu waluty, szukam dzień wcześniej
Błąd podczas pobierania kursu waluty, szukam dzień wcześniej
Kwota pozostała do zapłaty: -764.35
Czy chcesz wprowadzić kolejną fakturę? (tak/nie): t
Podaj numer faktury: 666
Podaj datę wystawienia faktury (YYYY-MM-DD): 2023-12-31
```

```
Podaj numer faktury: 7777
Podaj datę wystawienia faktury (YYYY-MM-DD): 2024-01-09
Podaj datę płatności (YYYY-MM-DD): 2024-02-08
Podaj kwotę zapłaty: 1200
Podaj walutę (PLN, USD, EUR, GBP): pln
Podaj kwotę faktury: 300
Podaj walutę (PLN, USD, EUR, GBP): eur
Kurs wymiany dla dnia wystawienia faktury: 4.2964
Kurs wymiany dla dnia płatności: 4.3023
Różnica kursowa dla nowej faktury numer 7777: 0.0059
Kwota pozostała do zapłaty: 88.92
Czy chcesz wprowadzić kolejną fakturę? (tak/nie): n
```