PROJEKT APLIKACJI W PYTHON DO WYLICZANIA RÓŻNIC WALUTOWYCH

autor: Grzegorz Kaszuba DSW 52705

Wprowadzenie

W ramach tego projektu, chciałem stworzyć aplikację w języku Python, która stanowi trzeci mój projekt w tym języku. Aplikacja ta ma na celu przyjęcie danych na temat faktur i płatności i z wykorzystaniem zewnętrznego API, wyliczenie koniecznych dopłat wynikających z różnic walutowych.

Ponieważ od wielu lat pracuje jako programista z doświadczeniem w PHP i Javascript, możliwe jest, że w kodzie tej aplikacji mogą wystąpić pewne wpływy z pracy w tych językach, które wynikają z mojej wcześniejszej pracy i pewnych przyzwyczajeń.

Opis Aplikacji

Moja aplikacja ma za zadanie obliczenie koniecznej do zapłaty różnicy, gdy istnieje rozbieżność w terminie płatności i wystawienia faktury.

Główne funkcje obejmują:

- odczyt i zapis danych do pliku csv
- tryb interaktywny pozwalający wprowadzać dane ręcznie

W projekcie można wyróżnić kluczowe komponenty, takie jak Api, Console i Calculator, które są odpowiedzialne za:

Api: stanowi warstwę komunikującą się np. z bankiem NBP, ale nic nie stoi na przeszkodzie, by dodać własny adapter do innego API lub np. zamockować do testów.

Console: odpowiada za operacje wyjścia i wejścia oraz interakcję z użytkownikiem i systemem

Calculator: stanowi główny trzon logiki biznesowej, oblicza różnicę i zwraca wynik do późniejszego wykorzystania

Technologie i Narzędzia
W trakcie tworzenia aplikacji korzystałem z języka Python3. Środowisko deweloperskie, w

którym pracowałem, to pyCharm od JetBrains. Świetnie integruje i przyspiesza pracę programisty w systemie Ubuntu, również integrując uruchamianie testów czy debugowanie kodu za pomocą debuggera. Dodatkowo posiłkowałem się ChatGPT4, który pozwalał przyspieszyć niektóre powtarzalne operacje, a także informował o np. składni języka, która jest jeszcze dla mnie tematem świeżym i nowym.

Struktura Katalogów
Struktura katalogów w projekcie została zaplanowana w sposób, który umożliwia czytelne rozmieszczenie plików i modułów. Główne katalogi to: src (źródło programu), tests (testy jednostkowe) i docs (dokumentacja). W katalogu znajdują się moduły Api, Calculator, Console, oraz Enum do którego trafiła klasa enumeracyjna Currency.

Analiza Kodu Źródłowego
Pamiętając o dobrych praktykach (m. in. SOLID), postanowiłem rozdzielić kod na osobne klasy, co pozwala na łatwiejsze zarządzanie kodem i ewentualny rozwój aplikacji.

Z tego powodu Console bierze na barki obsługę wejścia i wyjścia, a także odpowiednio przygotowuje dane, które przekazuje do Calculator i tak samo wyniki jego obliczeń przygotowuje do zapisu. Rozpoznaje na wstępie czy podane są w konsoli argumenty dla skryptu, a jeżeli nie - przełącza się do trybu interaktywnego. Odczytuje również zmienną środowiskową CURRENCIES w celu określenia dozwolonych w aplikacji walut. Jeżeli nie są podane - używa wszystkich. Jeżeli podano nazwę pliku z danymi wejściowymi jak załączony w repozytorium przykładowy plik input.csv:

```
invoice_amount,invoice_currency,invoice_date,payment_amount,paymen
t_currency,payment_date

400,USD,2023-02-10,400,USD,2023-06-10

400,USD,2023-02-10,400,USD,2023-06-10

400,EUR,2023-02-10,400,EUR,2023-06-10

400,EUR,2023-02-10,400,EUR,2023-06-10
```

to z pomocą reader z pakietu csv będzie odczytywać ten plik i obrabiać linia po linii (dla oszczędności pamięci ładuje tylko jedną linię naraz). W formie sztuczki programistycznej zostaje sprawdzony pierwszy wiersz, czy da się zamienić jego pierwszą wartość na float - jeżeli nie, program przyjmuje, że mamy do czynienia z nagłówkiem pliku csv i pomija go. Jeżeli plik nie istnieje lub wystąpi jakiś inny błąd - skrypt rzuci odpowiednim wyjątkiem.

Klasa Calculator jest mniej skomplikowana od klasy Console, ale dzięki wydzieleniu logiki może być użyta w innych miejscach - np. w aplikacji okienkowej lub www bez zmiany swojej zawartości, dzięki temu jest przenośna. Za pomocą adaptera api pobiera przelicznik walut w stosunku do PLN dla daty płatności i daty wystawienia faktury, a następnie wylicza różnicę i zwraca dane do późniejszej obróbki.

Testowanie
Oprócz testów manualnych, dopisałem w katalogu tests/ dwa proste testy jednostkowe zarówno dla Calculator jak i Console. Ponieważ pakiet unittest ma swoje sposoby na tworzenie mocków, musiałem posiłkować się wiedzą z internetu jak rozwiązać niektóre problemy, ale ostatecznie okazało się to dość intuicyjne i zrozumiałe.

Wnioski
Projekt ten był dla mnie cennym doświadczeniem, które pozwoliło mi rozwijać umiejętności programistyczne w Python. Praca nad aplikacją pozwoliła mi również wyciągnąć wnioski na temat korzystania z ChatGPT w pracy programisty - należy opracować ogólną architekturę samemu, stworzyć cały projekt od zera, a ewentualne sugestie “sztucznej inteligencji” traktować z przymrużeniem oka, bo potrafi często się mylić lub nie rozumieć problemu. Jednak pod względem np. generowania docstrings dla klas jest bardzo pomocny, przyspiesza opisywanie kodu bardzo. Ogólnie rzecz biorąc, był to wartościowy projekt, który przyczynił się do mojego rozwoju jako programisty.

Załączniki
Link do repozytorium GitHub, gdzie znajduje się kod źródłowy projektu, można znaleźć pod adresem: https://github.com/uirapuru/Zad3-R-niceWalutowe/tree/master