

Dokumentacja do programu currency-diff-calc Autor: Krzysztof Hager 52687

DOKUMENTACJA

Ten dokument ma przede wszystkim pomóc w zrozumieniu zależności pomiędzy funkcjami oraz modułami. Reszta przydatnych informacji znajduje się w docstringach, komentarzach oraz nazwach zmiennych i klas poszczególnych modułów.

Tryb interaktywny

Program uruchamia się w trybie interaktywnym, gdy przy uruchamianiu nie zostanie podany żaden argument, lub zostanie podany błędny argument dodatkowy.

Po uruchomieniu trybu interaktywnego, użytkownik ma możliwość wyboru jednego z kilku menu, w celu dotarcia do bardziej zaawansowanych funkcji:

1. Menu

Funkcja menu() w module menu.py jest głównym ekranem programu. Pełni przede wszystkim rolę organizacyjną i spina ze sobą pozostałe moduły.

Użytkownik porusza się w module menu, poprzez wpisanie jednej z dostępnych opcji (1-4 lub 'back', 'quit'), aby wybrać już jeden z modułów funkcyjnych lub cofnąć się do menu albo wyjść z programu.

W przypadku wybrania błędnej opcji, program wypisze informację o błędnym wyborze i kontynuuje z miejsca, w którym ostatnio skończył użytkownik (zarówno w menu, jak i modułach funkcyjnych).

1. Zarządzanie Fakturami (moduł faktury.py)

1. Dodaj nową Fakturę

Funkcja dodaj nową fakturę działa poprzez wywołanie funkcji `wprowadzenie_faktury()` z modułu faktur. Funkcja wymaga od użytkownika wprowadzenia:

- Nazwy firmy
- Waluty w kodzie ISO 4217 (aktualnie obsługiwane są USD, GBP, EUR) - klasa zaciągnięta z modułu waluty.py
- Daty wystawienia faktury (YYYY-MM-DD)
- Kwotę należności w podanej walucie

Po wpisaniu danych, są one oczyszczane i walidowane przez metodę klasy Faktura - `is_valid()`. Jeśli zwalidowane dane są poprawne, faktura zostanie dopisana do bazy `faktury.json` z kluczem podstawowym zwróconym z funkcji `baza.id_faktury()`.

2. Wyszukaj fakturę po ID

Po wywołaniu funkcji `wyszukaj_fakture_po_id()`, użytkownik jest proszony o podanie ID szukanej faktury. Jeśli identyfikator jest poprawny, wykonywane jest na bazie zapytanie `baza.wyszukaj_fakture(id_faktury)` (Query: `id_faktury == id_szukane`). Z wyniku zaciągane

są tzw. atrybuty dynamiczne (które są szczerze mówiąc zbędnym utrudnieniem - kiedyś zostaną poprawione) i wynik zapisywany jest z pomocą biblioteki Tabulate.

3. Wyszukaj fakturę po nazwie firmy

Działanie jest takie samo jak przy wyszukiwaniu po ID, z taką różnicą, że stosowane jest zapytanie `baza.wyszukaj_faktura_nazwa(nazwa)`, której wynikiem może być wiele rekordów (ze względu na to, że może być wiele faktur na tą samą firmę). Ponownie wyniki są prezentowane z pomocą biblioteki Tabulate.

4. Wypisz wszystkie faktury

Na bazie stosowane jest zapytanie `db_faktury.all()`, zwracające wszystkie rekordy znajdujące się w bazie, które potem są prezentowane z pomocą biblioteki Tabulate.

2. Zarządzanie Wpłatami (moduł wpłaty.py)

1. Dodaj nową Wpłatę

Funkcja dodaj nową fakturę działa poprzez wywołanie funkcji `wprowadzenie_wpłaty()` z modułu wpłat. Funkcja wymaga od użytkownika wprowadzenia:

- ID opłacanej faktury
- Wartość wpłaty w opłacanej walucie
- Walutę wpłaty w kodzie ISO 4217 (aktualnie obsługiwane są USD, GBP, EUR) - klasa zaciągnięta z modułu waluty.py
- Daty wystawienia faktury (YYYY-MM-DD)

Po wpisaniu danych, są one oczyszczane i walidowane przez metodę klasy Wpłata - `is_valid()`. Jeśli zwalidowane dane są poprawne, faktura zostanie dopisana do bazy `wplaty.json` z kluczem podstawowym zwróconym z funkcji `baza.id_wpłaty()`.

Jeśli wpłata zostanie wprowadzona w tej samej walucie, co opłacana faktura, to zostanie automatycznie wyliczona różnica kursowa pomiędzy datą wystawienia faktury a datą spłaty.

! UWAGA ! - Faktury mogą być opłacane w dowolonej walucie, nie tylko w tej, w której zostały wystawione i takie są też założenia tego programu. W przypadku opłacenia jej w innej walucie, zostanie wypisany komunikat o różniących się walutach, lecz różnica kursowa nie zostanie obliczona.

2. Wyszukaj Wpłatę po ID Wpłaty

Po wywołaniu funkcji `wyszukaj_wplate_po_id()`, użytkownik jest proszony o podanie ID szukanej wpłaty. Jeśli identyfikator jest poprawny, wykonywane jest na bazie zapytanie `baza.wyszukaj_wplate(id_wpłaty_szukana)` (Query: `id_wpłaty == id_wpłaty_szukana`). Z wyniku zaciągane są tzw. atrybuty dynamiczne (które są szczerze mówiąc zbędnym utrudnieniem - kiedyś zostaną poprawione) i wynik zapisywany jest z pomocą biblioteki Tabulate.

3. Wyszukaj Wpłatę po ID Faktury

Działanie jest takie samo jak przy wyszukiwaniu po ID, z taką różnicą, że stosowane jest zapytanie *wyszukaj_wplate_id_faktury(id_faktury_szukana)*, której wynikiem może być wiele rekordów (ze względu na to, że może być wiele wpłat na tę samą fakturę). Ponownie wyniki są prezentowane z pomocą biblioteki *Tabulate*.

4. Usuń Wpłatę

Przy wywołaniu funkcji *usun_wplate()*, użytkownik będzie poproszony o wskazanie ID usuwanej wpłaty. Po wpisaniu identyfikatora, jest on walidowany i w przypadku istnienia faktury o podanym ID, zostanie ona usunięta z bazy *wplaty.json* przy użyciu *usun_wplate(id_wplaty_do_usuniecia)*. Funkcja ta, poza usunięciem samej wpłaty, wyszukuje przypisaną do niej fakturę oraz przywraca do niej wartość usuniętego wpisu, aby faktura nie została błędnie opłacona.

3. Różnice kursowe (moduł waluty.py)

1. Oblicz różnicę kursową dwóch dowolnych dat

Funkcja *roznica_kursowa()* wywołuje dwa razy *wprowadzenie_waluty()*, tworząc dwa obiekty klasy *Waluta*.

Jeśli daty obu walut są poprawne (brak błędu API) oraz obie wskazują na tę samą walutę, zostanie obliczona dla nich różnica kursowa, wypisana z pomocą *tabulate*.

2. Oblicz różnicę kursową pomiędzy Fakturą a Wpłatą - modul *wplaty.py*

Funkcja *wplaty.roznice_kursowe()* wyszukuje w bazie Wpłatę w oparciu o jej ID, za pomocą *wyszukaj_wplate_po_id()* oraz odpowiadającą jej Fakturę z pola "id_faktury" oraz zapytania *baza.wyszukaj_fakture(wplata.id_oplacanej_faktury)*.

Jeśli daty obu walut są poprawne (brak błędu API) oraz obie wskazują na tę samą walutę, zostanie obliczona dla nich różnica kursowa, wypisana z pomocą *tabulate*.

3. Sprawdź kurs podanej waluty konkretnego dnia

Funkcja *konkretna_data()* tworzy jeden obiekt klasy *Waluta* z pomocą *wprowadzenie_waluty()*. Potem wypisywane są jej główne atrybuty: *kod_waluty*, *data*, *kurs* pod warunkiem, że wprowadzone dane przejdą walidację. Wynik jest wypisywany z pomocą biblioteki *tabulate*.

4. Zarządzanie Bazami Danych (moduł *baza.py*)

1. Wyczyść Bazę Faktur

Jak nazwa wskazuje, funkcja *wyczysc_baze_faktur()* usuwa wszystkie rekordy z bazy *faktury.json*.

2. Wyczyść Bazę Wpłat

Jak nazwa wskazuje, funkcja *wyczysc_baze_faktur()* usuwa wszystkie rekordy z bazy *faktury.json*.

Tryb wsadowy

Tryb wsadowy programu `currency-diff-calc`, służy przede wszystkim do masowego dodawania faktur oraz ich opłacania z poziomu komendy. **Program obsługuje jedynie pliki .csv, w strukturze:**

- Faktury: `wprowadzenie_faktury_z_pliku(nazwa_firmy, waluta, data, kwota_naleznosci)`
 - **Nazwa firmy, Waluta, Data wystawienia faktury, Kwota należności w obcej walucie**
 - np.: Biedronka, USD, 2024-01-30, 1000
 - Reszta danych jest wyliczana z pomocą modułów: `baza.py`, `waluty.py`
- Wpłaty: `wprowadzenie_wplaty_z_pliku(id_oplacanej_faktury, wartosc_wplaty, waluta, data)`
 - **ID opłacanej faktury, wartość wpłaty, waluta, data wpłaty**
 - np.: 1, 100, EUR, 2024-01-31
 - Reszta danych jest wyliczana z pomocą modułów: `baza.py`, `waluty.py`

Tryb wsadowy wywołuje się poprzez uruchomienie programu wraz z jednym z argumentów:

- "-h" (--help) - wypisanie pomocy programu
- "-fF" (--fileFaktura <plik.csv>) - masowe wprowadzenie nowych faktur z pliku .csv z podanej ścieżki
- "-fW" (--fileWpłata <plik.csv>) - masowe wprowadzenie nowych wpłat z pliku .csv z podanej ścieżki

Np.: `python currency-diff-calc.py -fF faktury.csv` lub `python currency-diff-calc.py -fW C:\Users\<username>\Desktop\wpłaty.csv`

KONIEC
