

# Dokumentacja aplikacji do rozliczeń grupowych ŚciAppka

Łukasz Szenkiel



## Streszczenie

Niniejszy dokument zawiera opis aplikacji do rozliczeń grupowych ŚciAppka, która umożliwia użytkownikom łatwe dzielenie się kosztami podczas wspólnych wyjazdów, zakupów i innych aktywności grupowych.

## 1 Ogólny zarys programu

Aplikacja ŚciAppka została zaprojektowana z myślą o użytkownikach, którzy często dzielą się kosztami w grupach. Każdy użytkownik może szybko i łatwo dodać swoje wydatki, a aplikacja automatycznie obliczy, ile każdy powinien zapłacić lub otrzymać. Program generuje raport, który jest zapisywany do pliku PDF i na potrzeby szkoleniowe MD.

## 2 Funkcjonalność programu w diagramach UML

W niniejszym projekcie kluczową rolę w wizualizacji struktury, przepływu pracy oraz interakcji użytkownika odegrało zastosowanie języka i narzędzia PlantUML. Wybór ten podyktowany był potrzebą efektywnego i spójnego przedstawienia aspektów systemu poprzez diagramy klas, czynności i przypadków użycia.

PlantUML stanowi innowacyjne podejście do tworzenia diagramów UML oraz innych wizualizacji technicznych, odchodząc od tradycyjnych graficznych edytorów na rzecz deklaratywnego języka tekstowego. Zamiast manualnego rozmieszczania elementów i połączeń, diagram definiowany jest poprzez precyzyjny opis składniowy, który następnie jest interpretowany przez silnik PlantUML w celu wygenerowania odpowiadającej mu reprezentacji graficznej. Wynikiem tego procesu są pliki obrazów w formatach takich jak PNG czy SVG, a także dokumenty PDF. Na potrzeby tej dokumentacji diagramy były eksportowane do formatu PGN.

Decyzja o implementacji PlantUML w procesie dokumentacji projektu przyniosła wymierne korzyści. Przede wszystkim, znacząco przyspieszyło to iteracyjny proces tworzenia i modyfikacji diagramów. Wszelkie zmiany w strukturze systemu czy logice działania mogły być szybko odzwierciedlone w diagramach poprzez prostą edycję opisu tekstowego, eliminując czasochłonne manualne poprawki graficzne. Co więcej, tekstowa reprezentacja diagramów naturalnie wpisuje się w praktyki kontroli wersji, umożliwiając śledzenie ewolucji wizualizacji w czasie i ułatwiając współpracę zespołu poprzez mechanizmy porównywania i scalania zmian.

Integracja PlantUML z wykorzystywanymi w projekcie narzędziami i środowiskami programistycznymi zapewniła płynny przepływ pracy. Opisy diagramów mogły być tworzone i zarządzane bezpośrednio w edytorach tekstu, a następnie automatycznie generowane do dokumentacji. Takie podejście gwarantuje również wysoką spójność wizualną wszystkich diagramów w projekcie, co przekłada się na lepszą czytelność i profesjonalizm dokumentacji.

W kontekście tego projektu, PlantUML okazał się nie tylko narzędziem do tworzenia statycznych obrazów, ale dynamicznym elementem procesu dokumentacyjnego, umożliwiającym efektywne komunikowanie złożonych koncepcji systemowych w sposób precyzyjny i zrozumiały. Wykorzystanie tego języka i narzędzia stanowiło świadomą decyzję mającą na celu podniesienie jakości i efektywności tworzonej dokumentacji.

## 2.1 Diagram klas

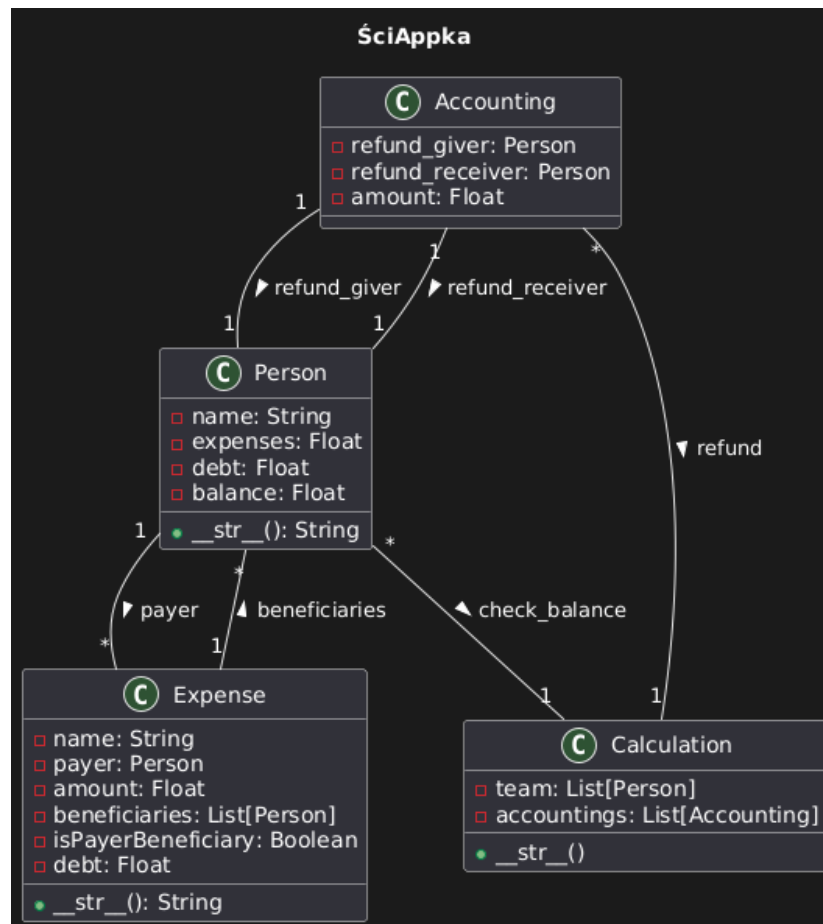


Diagram klas przedstawia kluczowe klasy aplikacji ŚciAppka i ich relacje. Klasa **Person** reprezentuje uczestnika z informacjami o imieniu, wydatkach, długu i saldzie. Klasa **Expense** opisuje pojedynczy wydatek, wskazując nazwę, płatcego, kwotę, beneficjentów oraz informację, czy płatcy jest również beneficjentem. Klasa **Accounting** reprezentuje pojedynczą transakcję rozliczeniową między dłużnikiem a wierzycielem. Klasa **Calculation** agreguje uczestników i wygenerowane rozliczenia. Relacje między klasami wskazują powiązania, takie jak płatcy wydatek, beneficjenci wydatku oraz transakcje rozliczeniowe między osobami w ramach rozliczenia.

## 2.2 Diagram przypadków użycia

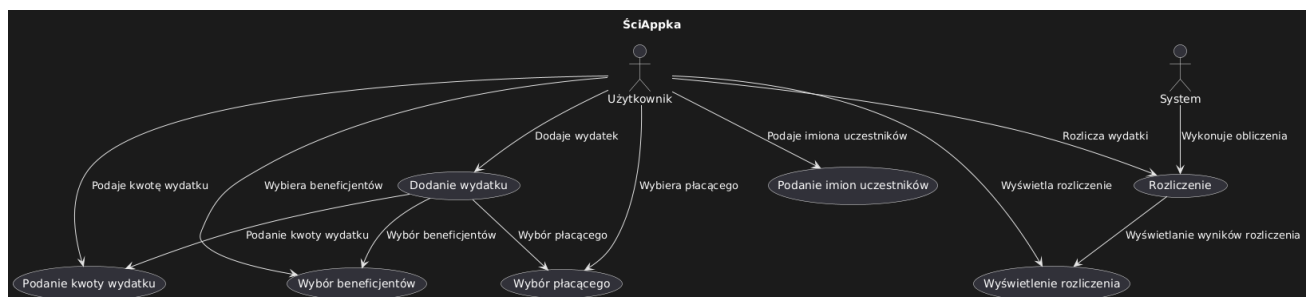


Diagram przypadków użycia przedstawia interakcje między użytkownikiem a systemem ŚciAppka. Użytkownik może podać imiona uczestników, dodawać wydatki (z wyborem płacącego, kwoty i beneficjentów), rozliczać wydatki oraz wyświetlać wynik rozliczenia. System wykonuje obliczenia rozliczeniowe.

## 2.3 Diagram czynności

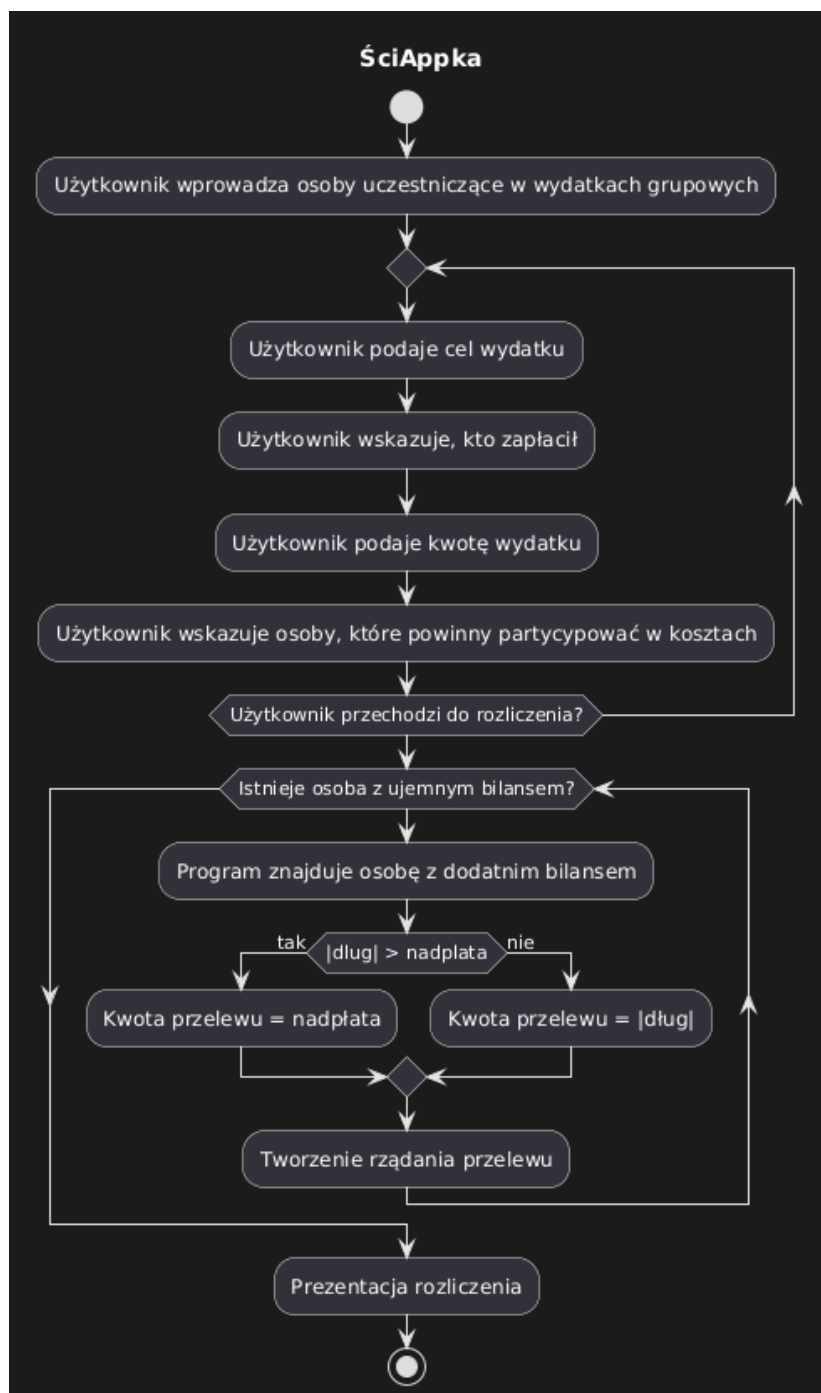


Diagram czynności przedstawia proces działania aplikacji ŚciAppka. Rozpoczyna się od wprowadze-

nia uczestników. Następnie, w pętli, użytkownik podaje szczegóły wydatków (cel, płacący, kwota, beneficjenci), aż zdecyduje się przejść do rozliczenia. Proces rozliczenia polega na iteracyjnym znajdowaniu osób z ujemnym i dodatnim bilansem oraz generowaniu przelewów wyrównujących różnice, aż wszystkie bilanse będą zerowe. Na końcu prezentowane jest rozliczenie.

## 3 Wbudowane funkcje i klasy

Aplikacja ŚciAppka wykorzystuje zarówno wbudowane funkcje i moduły języka Python, jak i zdefiniowane w ramach projektu klasy, do realizacji swojej funkcjonalności. Poniżej omówiono kluczowe elementy z obu tych kategorii.

### 3.1 Wbudowane funkcje i moduły Pythona

Aplikacja opiera się na kilku podstawowych wbudowanych funkcjach i modułach Pythona:

- `input()`: Funkcja ta służy do pobierania danych od użytkownika, np. imion uczestników, nazw wydatków, kwot. Jest to kluczowy element interakcji z użytkownikiem w trybie tekstowym.
- `print()`: Funkcja ta jest wykorzystywana do wyświetlania informacji na konsoli, takich jak komunikaty dla użytkownika, dane o wydatkach i wynik rozliczenia.
- `float()`: Funkcja ta służy do konwersji łańcucha znaków reprezentującego kwotę wydatku na liczbę zmiennoprzecinkową, co umożliwia wykonywanie operacji matematycznych.
- `int()`: Funkcja ta służy do konwersji łańcucha znaków na liczbę całkowitą, na przykład przy pobieraniu numeru osoby z listy uczestników.
- `list()`: Funkcja ta służy do tworzenia list, np. listy uczestników (`team`), listy wydatków (`expenses`) czy listy beneficjentów wydatku.
- `map()`: Funkcja ta jest używana w połączeniu z `int()` i `split()` do konwersji listy łańcuchów znaków reprezentujących numery beneficjentów na listę liczb całkowitych.
- `datetime (moduł)`: Moduł ten jest używany do generowania znacznika czasowego w raporcie (funkcja `datetime.now().strftime()`).
- `os (moduł)`: Moduł ten jest używany do tworzenia katalogu na wyjściowe pliki (funkcja `os.makedirs(óut", exist_ok=True)`).
- `subprocess (moduł)`: Moduł ten służy do uruchamiania zewnętrznego procesu, jakim jest kompilator LaTeX (`pdflatex`).

### 3.2 Klasy aplikacji

W aplikacji ŚciAppka zdefiniowano następujące klasy, które reprezentują kluczowe elementy logiki biznesowej:

- **Person**: Klasa ta reprezentuje uczestnika rozliczenia.
  - Atrybuty:
    - \* `name (str)`: Imię uczestnika.
    - \* `expenses (float)`: Suma wydatków poniesionych przez uczestnika.
    - \* `debt (float)`: Suma należności uczestnika (ile powinien zapłacić).
    - \* `balance (float)`: Saldo uczestnika (różnica między poniesionymi wydatkami a należnościami).
  - Metody:

- \* `__init__(self, name: str)`: Konstruktor klasy, inicjalizuje obiekt `Person` z podanym imieniem.
  - \* `__str__(self)`: Metoda specjalna, zwraca łańcuch znaków reprezentujący obiekt `Person`, zawierający informacje o wydatkach, należnościach i saldzie.
- **Expense**: Klasa ta reprezentuje pojedynczy wydatek.
    - Atrybuty:
      - \* `name (str)`: Nazwa wydatku.
      - \* `payer (Person)`: Osoba, która zapłaciła za wydatek.
      - \* `amount (float)`: Kwota wydatku.
      - \* `beneficiaries (List[Person])`: Lista osób, które uczestniczą w kosztach wydatku.
      - \* `is_payer_beneficiary (bool)`: Informacja, czy płacący jest również beneficjentem wydatku.
      - \* `debt (float)`: Kwota przypadająca na jednego beneficjenta.
    - Metody:
      - \* `__init__(self, name: str, payer: Person, amount: float, beneficiaries: List[Person])`: Konstruktor klasy, inicjalizuje obiekt `Expense` z podanymi danymi. Aktualizuje również informacje o wydatkach i saldzie płacącego oraz należnościach beneficjentów.
      - \* `__str__(self)`: Metoda specjalna, zwraca łańcuch znaków reprezentujący obiekt `Expense`, zawierający informacje o nazwie wydatku, płacącym, kwocie i beneficjentach.
  - **Accounting**: Klasa ta reprezentuje pojedynczą transakcję rozliczeniową między dwoma uczestnikami.
    - Atrybuty:
      - \* `refund_giver (Person)`: Osoba, która ma dokonać zwrotu.
      - \* `refund_receiver (Person)`: Osoba, która ma otrzymać zwrot.
      - \* `amount (float)`: Kwota zwrotu.
    - Metody:
      - \* `__init__(self, refund_giver: Person, refund_receiver: Person, amount: float)`: Konstruktor klasy, inicjalizuje obiekt `Accounting` i aktualizuje salda uczestników transakcji.
  - **Calculation**: Klasa ta odpowiada za przeprowadzenie obliczeń rozliczeniowych dla grupy osób.
    - Atrybuty:
      - \* `team (List[Person])`: Lista uczestników rozliczenia. `accountings (List[Accounting])`: Lista transakcji rozliczeniowych.
    - Metody:
      - \* `__init__(self, team: List[Person])`: Konstruktor klasy, inicjalizuje obiekt `Calculation` i przeprowadza obliczenia rozliczeniowe, tworząc listę obiektów `Accounting`.
      - \* `__str__(self)`: Metoda specjalna, zwraca łańcuch znaków reprezentujący wynik obliczeń rozliczeniowych, czyli listę transakcji zwrotów.

### 3.3 Relacje między klasami

Klasy w aplikacji ŚciAppka są powiązane ze sobą w następujący sposób:

- Uczestnik (`Person`) może być płacącym (`payer`) za wiele wydatków (`Expense`).

- Wydatek (Expense) ma jednego płacącego (payer) i wielu beneficjentów (beneficiaries), którzy są uczestnikami (Person).
- Obliczenie (Calculation) zawiera listę uczestników (Person) i listę transakcji rozliczeniowych (Accounting).
- Transakcja rozliczeniowa (Accounting) dotyczy dwóch uczestników (Person): osoby dokonującej zwrotu (refund\_giver) i osoby otrzymującej zwrot (refund\_receiver).

Te relacje pozwalają na modelowanie i przetwarzanie danych związanych z wydatkami grupowymi oraz generowanie rozliczeń między uczestnikami.

## 4 Dalsze możliwości rozwoju programu

Aplikacja ŚciAppka, w obecnej formie, realizuje podstawowe funkcje rozliczania wydatków grupowych. Istnieje jednak szereg możliwości dalszego rozwoju, które mogłyby znacząco zwiększyć jej użyteczność i funkcjonalność. Poniżej przedstawiono potencjalne kierunki rozwoju, skupiając się na prostszych do implementacji ulepszeniach oraz na zwiększeniu odporności programu na błędy.

### 4.1 Rozszerzenia funkcjonalności

- Interfejs graficzny (GUI): Opracowanie graficznego interfejsu użytkownika (np. przy użyciu biblioteki Tkinter, PyQt lub webowego frameworka Django/Flask) uczyniłoby aplikację bardziej intuicyjną i przyjazną dla użytkownika. Obecnie aplikacja działa w trybie tekstowym, co może być mniej wygodne dla niektórych użytkowników.
- Baza danych: Zamiast przechowywania danych w pamięci, można by użyć bazy danych (np. SQLite) do trwałego przechowywania danych rozliczeń. Obecnie po zakończeniu działania programu dane są tracone, a baza danych pozwoliłaby na ich zapisywanie i odczytywanie między uruchomieniami.
- Uwierzytelnianie użytkowników: Wprowadzenie prostego systemu uwierzytelniania użytkowników (np. logowanie z hasłem) umożliwiłoby oddzielenie danych różnych grup użytkowników i zwiększenie prywatności.
- Możliwość uwzględnienia płatności przez większą liczbę osób: Obecnie aplikacja zakłada, że jeden wydatek ma jednego płacącego. Rozszerzenie funkcjonalności o możliwość rozdzielenia płatności za jeden wydatek na kilka osób (np. każdy płaci część rachunku w restauracji) uczyniłoby aplikację bardziej elastyczną.
- Historia rozliczeń: Zapisywanie historii przeprowadzonych rozliczeń (np. w bazie danych lub plikach) pozwoliłoby na późniejsze przeglądanie i analizowanie wcześniejszych transakcji.

### 4.2 Zabezpieczenia przed błędnymi danymi

- Walidacja danych wejściowych: Wprowadzenie mechanizmów walidacji danych wejściowych od użytkownika jest kluczowe dla zapewnienia stabilności i poprawnego działania programu. Należy zweryfikować poprawność typów danych (np. czy kwota wydatku jest liczbą), zakresów wartości (np. czy numer osoby jest poprawny) oraz formatów (np. czy imiona uczestników są unikatowe).
- Obsługa wyjątków: Zastosowanie bloków try-except pozwoli na eleganckie obsłużenie potencjalnych błędów, które mogą wystąpić podczas przetwarzania danych wejściowych, np. ValueError przy konwersji łańcucha znaków na liczbę. Zamiast przerywać działanie programu, można wyświetlić użytkownikowi komunikat o błędzie i poprosić o ponowne wprowadzenie danych.

- Komunikaty dla użytkownika: W przypadku wykrycia niepoprawnych danych wejściowych, program powinien wyświetlać jasne i zrozumiałe komunikaty informujące użytkownika o rodzaju błędu i sposobie jego poprawienia.
- Pętle wymuszające poprawne dane: Można użyć pętli `while` do wymuszania na użytkowniku wprowadzenia poprawnych danych. Program powinien pytać użytkownika o dane tak długo, aż zostaną one wprowadzone poprawnie.

### 4.3 Usprawnienia techniczne

- Refaktoryzacja kodu: Przeprowadzenie refaktoryzacji kodu mogłoby poprawić jego czytelność, modularność i wydajność.
- Dokumentacja kodu: Uzupełnienie kodu o szczegółowe komentarze i dokumentację (np. w formacie Sphinx) ułatwiłoby jego zrozumienie i utrzymanie.

## 5 Instrukcja obsługi wraz z najczęstszymi pytaniami

### 5.1 Instrukcja obsługi

#### 1. Uruchomienie programu Aby uruchomić program ŚciAppka:

- zainstaluj Python (wersja 3.6+). Sprawdź: `python --version` lub `python3 --version`.
- zainstaluj bibliotekę `plantuml`: `pip install plantuml`
- zainstaluj LaTeX (np. MiKTeX lub TeX Live) i dodaj go do `PATH`, jeśli chcesz generować raporty PDF.
- uruchom program: otwórz terminal, przejdź do katalogu z plikiem `main.py` i wykonaj: `python main.py` lub `python3 main.py`.

```
lukasz@szenkiel-ubuntu:~/Pulpit/Informatyka/Programowanie_objektowe/projekt/SciAppka/app$ python3 main.py
```

#### 2. Wprowadzenie uczestników rozliczenia Po uruchomieniu programu zostaniesz poproszony o podanie imion uczestników rozliczenia.

- wprowadź imiona: Wpisz imiona wszystkich osób, które brały udział w wydatkach, oddzielając każde imię przecinkiem.
- unikatowe imiona: Pamiętaj, że imiona muszą być unikatowe. Program używa ich do identyfikacji uczestników.
- format wprowadzenia: Upewnij się, że imiona są rozdzielone tylko przecinkami, bez dodatkowych spacji po przecinku (chyba że spacja jest częścią imienia).
- potwierdź wprowadzenie: Po wprowadzeniu wszystkich imion naciśnij klawisz `Enter`.

```
lukasz@szenkiel-ubuntu:~/Pulpit/Informatyka/Programowanie_objektowe/projekt/SciAppka/app$ python3 main.py
Program rozlicza wydatki grupowe i na koniec podaje, kto komu ile ma zwrócić.
Podaj imiona (unikatowe) uczestników rozliczenia (rozdzielone przecinkami):
Wojtek, Michal, Stefan, Anna, Wieslaw, Genowefa, Jerzy
```

#### 3. Wprowadzenie wydatków Po wprowadzeniu uczestników program umożliwi dodawanie informacji o poszczególnych wydatkach. Dla każdego wydatku należy podać następujące dane:



- ```

Podaj nazwę wydatku:
1 - UbuntuSoftware
Kwota wydatku (podaj numer odpowiedniej osoby)
1 - Wojtek
2 - Michał
3 - Stefan
4 - Anna
5 - Wiesław
6 - Genowefa
7 - Jerzy
8 - wszyscy
Jaka kwotę zapłacił Wiesław?
42.59
Wybierz, kto korzystał z tego wydatku: (podaj numery rozdzielone przecinkami)
1 - Wojtek
2 - Michał
3 - Stefan
4 - Anna
5 - Wiesław
6 - Genowefa
7 - Jerzy
8 - wszyscy
Podaj liczby oddzielone przecinkami: 8
1 - dodaj wydatek
2 - rozliczenie

```

- wybierz opcję rozliczenia: Po dodaniu wydatku program zapyta, czy chcesz dodać kolejny wydatek, czy przejść do rozliczenia.
- wprowadź "2": Aby przejść do rozliczenia, wprowadź cyfrę "2" i naciśnij Enter.

## 5.2 Najczęstsze pytania (FAQ)

- Program ŚciAppka służy do rozliczania wydatków grupowych. Pomaga ustalić, kto komu i ile pieniędzy powinien zwrócić po wspólnych wydatkach, takich jak wyjścia do restauracji, podróże czy zakupy.

- Program przyjmuje następujące dane wejściowe:

- 9

- kwotę wydatku.
- listę osób, które brały udział w wydatku (beneficjentów).

### 3. Jak program definiuje "beneficjenta" wydatku?

Beneficjentami wydatku są wszystkie osoby, które powinny ponieść część jego kosztów. Może to być cała grupa lub tylko jej część.

### 4. Czy osoba, która zapłaciła za wydatek, może być również jego beneficjentem?

Tak, osoba płażąca może być również beneficjentem wydatku. W takim przypadku jej udział w kosztach jest odpowiednio uwzględniany.

### 5. Jak program oblicza, kto komu powinien zwrócić pieniądze?

Program oblicza należności każdego uczestnika na podstawie wprowadzonych wydatków i beneficjentów. Następnie generuje listę transakcji zwrotów, minimalizując liczbę przelewów między uczestnikami.

### 6. W jakiej formie prezentowane są wyniki rozliczenia?

Program generuje raport w kilku formatach:

Tekstowy (wyświetlany na konsoli). Markdown (.md) LaTeX (.tex) PDF (generowany z pliku .tex) Diagram rozliczenia (w formacie .png i jako część raportu)

### 7. Czy program generuje diagram rozliczenia?

Tak, program generuje diagram rozliczenia w formacie PNG, który wizualizuje przepływ pieniędzy między uczestnikami.

### 8. Czy program zapisuje dane rozliczenia?

Obecna wersja programu nie zapisuje danych rozliczenia trwale. Po zakończeniu działania programu dane są tracone. ( Warto zaznaczyć, że w rozdziale o dalszym rozwoju sugerujemy dodanie bazy danych, żeby to zmienić )

### 9. Jak uruchomić program?

Program jest napisany w języku Python i wymaga jego zainstalowania. Aby go uruchomić, należy uruchomić skrypt Pythona (np. `python main.py`).

### 10. Czy program wymaga zainstalowania dodatkowych bibliotek?

Tak, program wymaga zainstalowania bibliotek `plantuml` i `subprocess` (który zazwyczaj jest częścią standardowej biblioteki Pythona), a także systemu LaTeX (np. TeX Live lub MiKTeX), jeśli chcemy generować raport w formacie PDF.

## 5.3 Prezentacja wyników przykładowego użycia

### 5.3.1 Raport Markdown

| Raport rozliczeniowy wydatków grupowych                                                                                                          |                                                     |
|--------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| H1                                                                                                                                               | Uczestnicy rozliczenia                              |
| <ul style="list-style-type: none"><li>• Maciek</li><li>• Agata</li><li>• Wojtek</li><li>• Milena</li><li>• Eugeniusz</li><li>• Andrzej</li></ul> |                                                     |
| -----                                                                                                                                            |                                                     |
| Wydatki                                                                                                                                          |                                                     |
| Zakupy spożywcze                                                                                                                                 |                                                     |
| Placący:                                                                                                                                         | Maciek ↗                                            |
| Kwota:                                                                                                                                           | 120.00 zł ↗                                         |
| Beneficjenci wydatku:                                                                                                                            | Maciek, Agata, Wojtek ↗                             |
| Należność na głowę:                                                                                                                              | 40.00 zł                                            |
| Bilety do kina                                                                                                                                   |                                                     |
| Placący:                                                                                                                                         | Agata ↗                                             |
| Kwota:                                                                                                                                           | 90.00 zł ↗                                          |
| Beneficjenci wydatku:                                                                                                                            | Maciek, Agata, Wojtek, Milena ↗                     |
| Należność na głowę:                                                                                                                              | 22.50 zł                                            |
| Pizza                                                                                                                                            |                                                     |
| Placący:                                                                                                                                         | Wojtek ↗                                            |
| Kwota:                                                                                                                                           | 80.00 zł ↗                                          |
| Beneficjenci wydatku:                                                                                                                            | Maciek, Agata, Wojtek, Milena, Eugeniusz, Andrzej ↗ |
| Należność na głowę:                                                                                                                              | 13.33 zł                                            |
| Taxi                                                                                                                                             |                                                     |
| Placący:                                                                                                                                         | Eugeniusz ↗                                         |
| Kwota:                                                                                                                                           | 60.00 zł ↗                                          |
| Beneficjenci wydatku:                                                                                                                            | Eugeniusz, Andrzej ↗                                |
| Należność na głowę:                                                                                                                              | 30.00 zł                                            |
| Hotel                                                                                                                                            |                                                     |
| Placący:                                                                                                                                         | Milena ↗                                            |
| Kwota:                                                                                                                                           | 200.00 zł ↗                                         |
| Beneficjenci wydatku:                                                                                                                            | Maciek, Agata, Wojtek, Milena, Eugeniusz, Andrzej ↗ |
| Należność na głowę:                                                                                                                              | 33.33 zł                                            |

## Dyskoteka

Placący: Agata ↗

Kwota: 70.00 zł ↗

Beneficjenci wydatku: Agata, Milena ↗

Należność na głowę: 35.00 zł

## Indywidualny koszt

| Osoba     | Należność |
|-----------|-----------|
| Maciek    | 109.17 zł |
| Agata     | 144.17 zł |
| Wojtek    | 109.17 zł |
| Milena    | 104.17 zł |
| Eugeniusz | 76.67 zł  |
| Andrzej   | 76.67 zł  |

## Rozliczenie

Wojtek --- 10.83 zł ----> Maciek

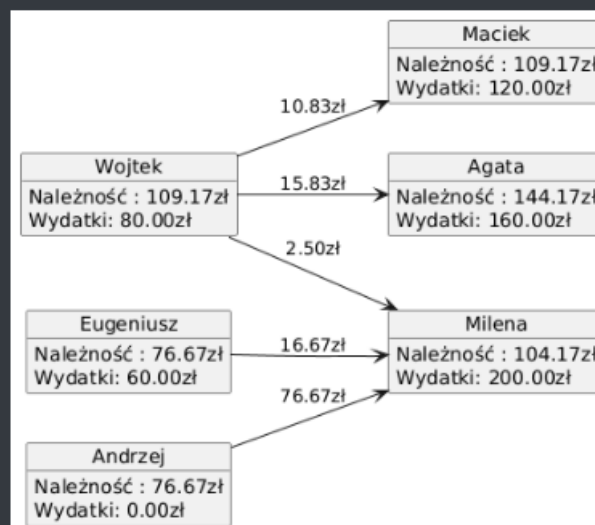
Wojtek --- 15.83 zł ----> Agata

Wojtek --- 2.50 zł ----> Milena

Eugeniusz --- 16.67 zł ----> Milena

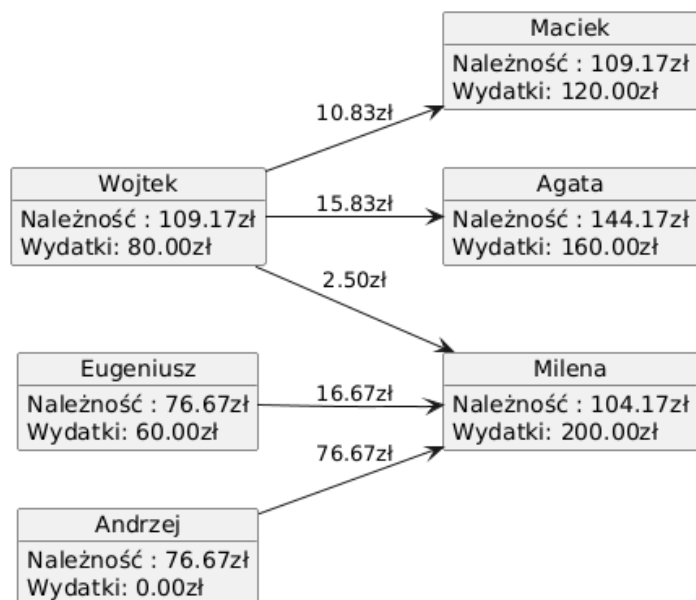
Andrzej --- 76.67 zł ----> Milena

## Diagram rozliczenia



Raport wygenerowano: 13-05-2025 23-18-29

### 5.3.2 Diagram .png



### 5.3.3 Raport PDF

|                                                                                                                                                  |  |
|--------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <b>Rozliczenie wydatków grupowych</b>                                                                                                            |  |
| <b>Uczestnicy rozliczenia</b>                                                                                                                    |  |
| <ul style="list-style-type: none"><li>• Maciek</li><li>• Agata</li><li>• Wojtek</li><li>• Milena</li><li>• Eugeniusz</li><li>• Andrzej</li></ul> |  |
| <b>Wydatki</b>                                                                                                                                   |  |
| <b>Zakupy spożywcze</b>                                                                                                                          |  |
| Placący: Maciek Kwota: 120.00zł Beneficjenci wydatku: Maciek, Agata, Wojtek Należność na głowę: 40.00zł                                          |  |
| <b>Bilety do kina</b>                                                                                                                            |  |
| Placący: Agata Kwota: 90.00zł Beneficjenci wydatku: Maciek, Agata, Wojtek, Milena Należność na głowę: 22.50zł                                    |  |
| <b>Pizza</b>                                                                                                                                     |  |
| Placący: Wojtek Kwota: 80.00zł Beneficjenci wydatku: Maciek, Agata, Wojtek, Milena, Eugeniusz, Andrzej Należność na głowę: 13.33zł               |  |
| <b>Taxi</b>                                                                                                                                      |  |
| Placący: Eugeniusz Kwota: 60.00zł Beneficjenci wydatku: Eugeniusz, Andrzej Należność na głowę: 30.00zł                                           |  |
| <b>Hotel</b>                                                                                                                                     |  |
| Placący: Milena Kwota: 200.00zł Beneficjenci wydatku: Maciek, Agata, Wojtek, Milena, Eugeniusz, Andrzej Należność na głowę: 33.33zł              |  |
| <b>Dyskoteka</b>                                                                                                                                 |  |
| Placący: Agata Kwota: 70.00zł Beneficjenci wydatku: Agata, Milena Należność na głowę: 35.00zł                                                    |  |
| <b>Indywidualny koszt</b>                                                                                                                        |  |
| Maciek -> 109.17zł                                                                                                                               |  |
| Agata -> 144.17zł                                                                                                                                |  |
| Wojtek -> 109.17zł                                                                                                                               |  |
| Milena -> 104.17zł                                                                                                                               |  |
| Eugeniusz -> 76.67zł                                                                                                                             |  |
| Andrzej -> 76.67zł                                                                                                                               |  |

1

