

# Kluczowe klasy

## 1. User

dziedziczy po klasie *LogInData*

Klasa odpowiedzialna za przechowywanie unikalnych danych użytkownika takich jak lista kont czy lista przyjaciół (lista obiektów typu *PaymentRetriever*).

## 2. Account

klasa, bazowa dla wszystkich rodzajów kont. Obiekty tej klasy będą przechowywane w mapie w klasie *Bank* oraz w liście w klasie *User*.

## 3. Bank

klasa nie dziedziczy i nie jest dziedziczona. Zawiera pole *User* oraz przechowuje mapę wszystkich kont z pliku JSON.

## 4. JsonManager

klasa odpowiedzialna za obsługę plików JSON. Współpracuje z klasą *Bank*.

## 5. Payment

klasa obsługuje wszystkie przelewy w banku.

## 6. Card

klasa bazowa dla wszystkich rodzajów kart.

## 7. Fund

klasa bazowa dla wszystkich rodzajów funduszy.

## 8. History

klasa przechowująca listę wszystkich płatności. Jest kluczowa przy wyświetlaniu historii transakcji użytkownika. Klasa udostępnia publiczne metody *Sort()*, pozwalające posortować listę według różnych kryterium.

## 9. Config

klasa wczytuje nazwy plików oraz inne niezbędne dane z pliku *config.txt*.

# Pliki

## 1. LogInData.txt

plik przechowuje zaszyfrowane dane logowania w formacie: login haslo id np.  
*superman trudnehaslo22 235865*

## 2. AccountData.json

przechowuje dane dotyczące kont oraz kart. Głównymi obiektami w pliku są konta.

## 3. FriendsData.json

plik przechowuje dane dotyczące listy znajomych użytkownika. Pogrupowane według ID użytkownika.

## 4. FundsData.json

plik z danymi dotyczącymi funduszy. Pogrupowany funduszami.

## 5. ConfigFile.txt.

plik z nazwami i ścieżkami do wszystkich plików.

# Przykładowy przebieg programu

## 1. Logowanie

W klasie *UI* tworzymy obiekt *LogInData* na podstawie danych wprowadzonych przez użytkownika. Obiekt jest przekazywany do statycznej klasy *Authorization*, która sprawdza czy w pliku *LogInData.txt* istnieje użytkownik. Następnie jest on przekazywany do metody *LogIn()* w klasie *Bank*, która będzie tworzyć obiekt klasy *User*.

## 2. Wczytywanie danych z plików

W klasie *Bank* tworzona jest mapa przechowująca obiekty klasy *Account*, które są tworzone na podstawie danych z pliku *UserData.json*. Plik jest obsługiwany przez klasę *JsonManager*.

## 3. Przelew

Na rzecz *User* wywołana jest metoda *MakePayment()*, która tworzy obiekt klasy *Payment*. Jeżeli wprowadzone dane są poprawne, płatność zostaje zaksięgowana poprzez wywołanie metody *AddNewRecord()* w klasie *History* oraz zaktualizowane zostają konta w mapie w klasie *Bank*.

## 4. Wyświetlanie historii

Klasa *History* zwraca listę obiektów *payment* posortowaną według podanego klucza.

## 5. Wylogowanie

Przy wylogowywaniu mapa kont z klasy *Bank* zostaje przekazana do klasy *JsonManager*, która nadpisuje plik *AccountData.json*.