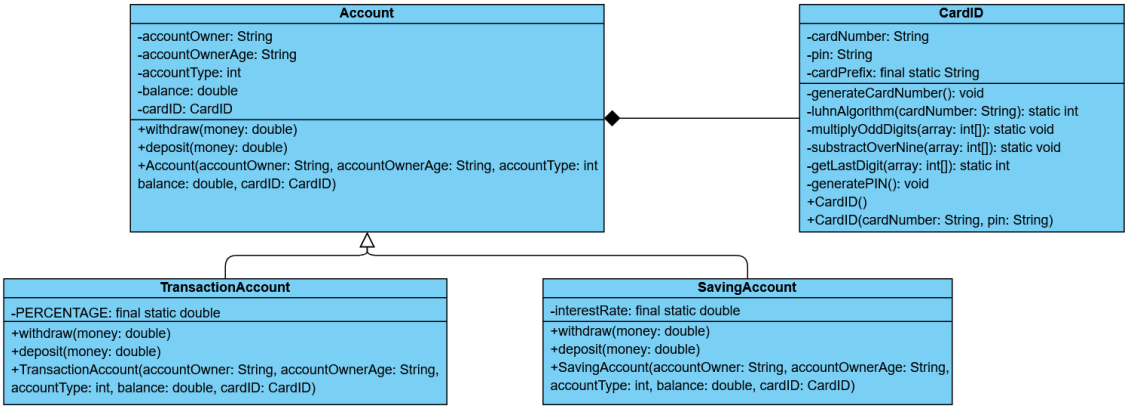
Adam Jurcz
144441; Grupa I5.1

DOKUMENTACJA DRUGIEGO PROJEKTU
PROGRAMOWANIE OBIEKTOWE

1. Zakres projektu

-Projekt ten zawiera implementację systemu bankowego, którego rdzeń został bardzo zbliżony do poprzedniego mojego projektu na przedmiot Programowanie Obiektowe. Można założyć konto, przy czym do wyboru mamy dwa rodzaje kont ***(transakcyjne oraz oszczędnościowe)***, które dziedziczą składowe oraz zarys metod po bardziej ogólnej definicji konta bankowego. W przeciwieństwie do poprzedniej implementacji, tutaj ważna jest synchronizacja pomiędzy różnymi stanami wyświetlanego obrazu (np. przejście z systemu logowania do menu głównego systemu bankowego), a także zapewnienie swobodnego dostępu do lokalnej bazy danych przechowującej informacje na temat wszystkich założonych kont. W projekcie użyłem technologii ***JavaFX*** w celu lepszej znajomości tej platformy w przyszłych projektach, które będę wykonywał. Natomiast interfejs wymiany danych lokalnie zaimplementowałem przy pomocy ***JDBC***.

2. Diagram UML



(Zawiera on tylko rdzeń modeli klas odpowiadających za przechowywanie w dynamicznej pamięci niektórych pól oraz operacji)

3. Realizacja projektu

-W celu implementacji jako pierwsze wykonałem klasy modeli pokazane na diagramie UML. Pomocne one są do akcji na bazie danych, a także na kontrolerach. Wszystkie najważniejsze akcje ukazywane na ekranie zostały przygotowane przy pomocy SceneBuildera dla ***JavaFX***, który korzysta z interfejsu znacznikowego ***FXML***. Po przypisaniu danej sceny do kontrolera, opisałem odpowiednie metody do pewnych interakcji, przykładowo po wprowadzeniu prawidłowych wartości z danego zakresu dla pożyczki (maksymalna kwota to odpowiednia wartość *PERCENTAGE* klasy *TransactionAccount*), dostajemy odpowiedź- czy operacja się udała, a jeżeli nie, to dlaczego.

4. Przyszłe usprawnienia

-zamiast dynamicznej zmiany scen na kolejne *„w locie”*, możemy zrobić strukturę przechowującą różne pliki *fxml* po załadowaniu, dzięki czemu możemy zyskać na szybkości programu,

-dokładniejsze operacje na liczbach dziesiętnych, w prawdziwym systemie bankowym każda cyfra ma znaczenie,

-synchronizacja wielowątkowa,

-synchronizacja pomiędzy ładowaniem etapów, żeby jakakolwiek awaria nie miała wpływu na dane w systemie,

-dokładniejszy system logowania,

-możliwość zmiany rozmiaru okna,

-ładniejszy *UI/UX*.