Gliwice, 05.02.2020

**LABORATORIUM PROGRAMOWANIA KOMPUTERÓW**

**Temat projektu: System zarządzania biblioteką**

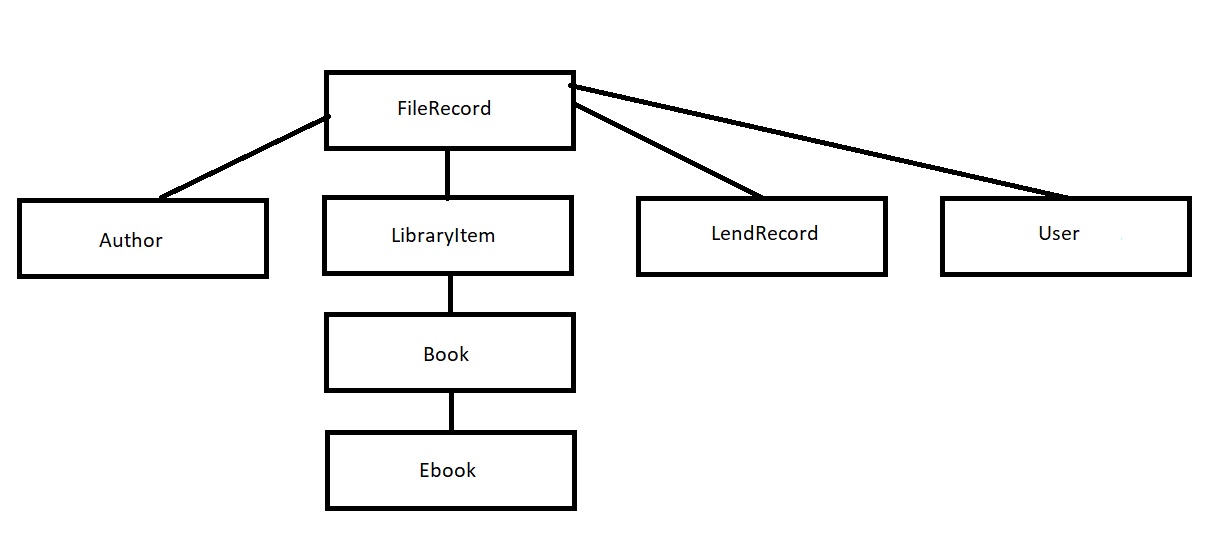
Autor: Jagoda Chmielewska

Informatyka, semestr 3, grupa 1

1. **Temat projektu:**

Należało napisać system obsługi biblioteki w obiektowym c++. Program umożliwia zarządzanie biblioteką z konta administratora (admina) oraz zwykłe wypożyczanie przedmiotów z konta użytkownika (usera). Dane przechowywane są w plikach .csv.

1. **Diagram klas:**

****

1. **Przebieg aplikacji:**

-Ładowanie danych:

Aplikacja ładuje dane z plików wprowadzonych na stałe (hardcoded) w rozwiązaniu. Różne typy obiektów są przechowywane w innych wektorach. Wektory posiadają tylko wskaźniki do obiektów

-Wiązanie danych:

Obiekty posiadają referencje do siebie nawzajem. Przechowują tylko id w plikach z danymi. Używają ID żeby znaleźć obiekt i związać go z innymi kolekcjami.

-Login:

-Konsola:

Użytkownik może wybrać różne opcje. Różni użytkownicy posiadają różne menu i funkcje   
w nich zawarte. Użytkownik może tylko zobaczyć dane. Wypozyczac musi poprzez admina biblioteki.

Admin może robić wszystko:

Autorzy: dodawanie, oglądanie, usuwanie

Użytkownicy: dodawanie, oglądanie, usuwanie

Przedmioty: dodawanie, oglądanie, usuwanie

Wypożyczenia:

-mogą być oglądane(użytkownik może zobaczyć tylko własne wypożyczenia, admin widzi wszystkie)

-mogą być zwracane. Po zwróceniu przedmiotu pojawia się on w historii wypożyczeń. Użytkownik musi zapłacić za wypożyczenie(cena wypożyczenia + 10% za każdy tydzień opóźnienia)

-każdy przedmiot może być wypożyczony raz(Jeśli chcemy mieć w bibliotece trzy te same książki to muszą zostać one dodane trzy razy)

-zakończenie programu

1. **Format danych**

Dane są przechowywane w plikach tekstowych (.csv). Nie jest to czysty plik .csv. Zachowuje się jak plik tekstowy (.txt).

Format danych liczy się tylko do zapisywania i pobierania danych.

1. **Krótkie omówienie klas i plików:**

* *LibrarySystem.cpp*

Plik zawierający main(). Znajduje się tam cała obsługa aplikacji/konsoli.

* *Library manager*

Klasa, która przechowuje całe dane i jest połączona z GUI. Cała "logika" aplikacji się tam znajduje.

* *Helpers*

Klasa z statycznymi metodami, które wykonują podstawowe rzeczy takie jak odczytywanie/zapisywanie plików, rozdzielanie stringów.

* *Structs*

Zawiera podstawowe typy używane w aplikacji (SimpleData i PersonData).

SimpleData jest prostym "wrapperem" ( funkcje lub klasy, których zadaniem jest jedynie wywołanie innych funkcji lub metod obiektów) ponieważ używanie funckji związanych  
 z datą i czasem jest bardzo skomplikowane.

PersonData umożliwia nie dołączać wszędzie name, surname, birthday.

* *FileRecord*

Abstrakcyjna klasa, która zachowuje się jak interfejs umożliwiający odczytywanie  
 i zapisywanie danych.

* *LibraryItem, Book, EBook*

Przedmioty w bibliotece. LibraryItem jest podstawowym przedmiotem, reszta jest tylko po to żeby pokazać dziedziczenie.

Book i EBook mają tylko małe różnice między sobą. Mogłoby zostać stworzone więcej przedmiotów, ale wszystkie byłyby takie same i jest to bezcelowe.

* *LendRecord*

Rezprezentuję wypożyczenia. Łączy użytkownika z przedmiotami i ustawia datę i cenę za wypożyczenie.

1. **Testowanie:**

Program został przetestowany na różnego rodzaju plikach. Pliki puste nie powodują błędów. Program działa poprawnie, został także sprawdzony pod kątem wycieków pamięci tak aby nie dopuścić do żadnego z nich.

1. **Wnioski:**

Program może wydawać się programem prostym, ale wymaga umiejętności posługiwania się wektorami, dziedziczeniem, metodami wirtualnymi czy obsługą plików. Duża trudnością było rozplanowanie działania aplikacji.

1. **Literatura:**

* <http://cpp0x.pl/>
* <https://stackoverflow.com/>
* Jerzy Grębosz „Opus magnum C++11. Programowanie w języku C++.”
* Stephen Prata „Język C++. Szkoła programowania.”

1. **Wydruk programu:**