

# Zadanie 7

## Programowanie Java

Dzisiaj (i w następnym tygodniu) skonstruujesz prosty program, który symuluje działanie biblioteki.

Dzisiaj należy zaprojektować i przetestować wstępną hierarchię klas, która reprezentuje pozycje w bibliotece. Istnieją trzy rodzaje materiałów, które można wypożyczyć z biblioteki: zwykłe książki, czasopisma i filmy.

Zwykłe książki można wypożyczyć na dwa tygodnie, filmy na 2 dni. Czasopisma można wypożyczyć na trzy dni, jeśli jesteś studentem, i na tydzień, jeśli jesteś nauczycielem. Studenci mogą wypożyczyć maksymalnie jeden film w tym samym czasie, trzy książki i trzy czasopisma. Nie ma limitu dla nauczycieli.

Pobierana jest zaległa opłata, 0,5 PLN dziennie za książkę, 2 PLN dziennie za czasopismo i 5 PLN dziennie za film.

Wszystkie pozycje mają przypisany unikatowy identyfikator w bibliotece. Książki mają wpisy `Title`, `Author`, `Genre` i `Publisher`. Czasopisma mają wpisy `eISSN`, `Publisher`, `Latest issue` i `Journal URL`. Filmy mają wpisy `Title`, `Genre`, `Director`, `Year`, `Runtime (Minutes)`, `Rating`. Należy pamiętać, że niektóre pola mogą być niekompletne.

Zdefiniuj klasę abstrakcyjną `LibraryItem` (która jest rozszerzona o klasy `Book`, `Journal` i `Film`), zawierając pola informujące czy dana pozycja została wypożyczona, przez kogo, i kiedy, i podnieś te pola, które można podnieść do klasy bazowej.

Zaprojektuj metodę `daysOverdue`, która zwraca liczbę dni, w których ta pozycja jest zaległa. Jeśli liczba jest ujemna, przedmiot może być nadal niedostępny przez określoną liczbę dni.

Zaprojektuj metodę `isOverdue`, która generuje wartość logiczną, która informuje nas o tym, czy pozycja jest przeterminowana w danym dniu.

Zaprojektuj metodę `computeFine`, która oblicza karę dla tej pozycji, jeśli zostanie zwrócona z opóźnieniem. Wszystkie trzy powyższe metody powinny przyjmować jako parametr bieżący dzień (zdefiniowany jako liczba całkowita, liczona od początku roku.)

Korzystając z załączonych plików tekstowych, wczytaj wszystkie pozycje w bibliotece.

Wszystkie instancje reprezentujące pozycje biblioteczne powinny być przechowywane w klasie `Library`.

Zaprojektuj metodę `borrowItem` należącą do klasy `Library`, umożliwiającą wypożyczenie dowolnej pozycji w dowolnym dniu. Metoda ta będzie przydatna dla utworzenia testów jednostkowych dla metod `daysOverdue`, `isOverdue`, `computeFine`. Utwórz takie testy i uruchom dla konkretnych przykładów, korzystając ze środowiska JUnit.

Dodaj możliwość sprawdzania przez pojedynczego użytkownika stanu jego konta na dany dzień oraz sumy kar za opóźnienia zwrotu, metodę wyświetlającą użytkowników z opóźnieniami zwrotu na dany dzień, oraz metodę blokującą wypożyczenie przez użytkowników mających na koncie karę większą niż ustawiony próg.

Wywołaj odnośne metody co najmniej dla dwóch użytkowników i dwu różnych dni.