

Zadania Lab 2

1. Utwórz nową solucję / projekt C++
 - a. Opcjonalnie: Skonfiguruj Gita. **NIE LOGUJ SIĘ DO GITA NA KOMPUTERZE LABORATORYJNYM – Git Credential Manager zapamiętuje poświadczenia i możesz mieć potem kłopoty.**
2. Zaprojektuj prosty system zarządzania biblioteką, składający się z trzech klas:
 - a. `Book` – Książka. Posiada dane takie jak:
 - i. `title` – Tytuł (string 32 znaki)
 - ii. `author` – Autor (string 32 znaki)
 - iii. `isbn` – Numer ISBN (string 10 znaków)
 - iv. `available` – Czy jest dostępna (bool)
 - b. `Member` – Czytelnik. Posiada dane takie jak:
 - i. `name` – Nazwa (string 32 znaki)
 - ii. `booksBorrowed` – Ile książek pożyczył (unsigned int)
 - c. `Library` – Biblioteka. Posiada dane takie jak:
 - i. `books` – Tablica przechowująca książki – Załóż statyczny rozmiar 32 obiektów typu `Book`

Pamiętaj o tym żeby dane zostały zabezpieczone zgodnie z zasadą enkapsulacji – mają być `private`. Dodaj publiczne metody getterów i seterów: Getter ma pobrać wartość zwracając obiekt. Setter ma wartość ustawić zwracając `void` i przyjmując obiekt. Do obsługi stringów możesz wykorzystać tablice (jak w C) lub stringi – `std::string`. Pamiętaj też o zasadzie jedna klasa – jedna para plików.

3. Dodaj dodatkowe publiczne metody do klas:
 - a. Dla `Book`:
 - i. Konstruktor trzyargumentowy (`tytuł`, `autor`, `isbn`), który zainicjalizuje obiekt przy użyciu listy inicjalizacyjnej ustawiając flagę `available` na `true`.
 - ii. Destruktor – wypisze – „niszczę książkę”
 - iii. `getInfo()` – Wypisującą na ekran informację o książce
 - b. Dla `Member`:
 - i. Konstruktor jednoargumentowy (`name`), który zainicjalizuje obiekt bez użycia listy inicjalizacyjnej
 - ii. `borrowBook()` – Zwiększającą liczbę pożyczonych książek
 - iii. `returnBook()` – Zmniejszającą liczbę pożyczonych książek

Pamiętaj o zabezpieczeniu zakresu `unsigned int`a
 - c. Dla `Library`:
 - i. `addBook(Book b)` – Dodającą na wolny slot książkę
 - ii. `removeBook(Book b)` – Usuającą książkę
 - iii. `borrowBook(Member, isbn)` – Pożyczającą książkę (ustawia flagę)
 - iv. `returnBook(Member, isbn)` – Zwracającą książkę (usuwa flagę)
 - v. `displayBooks()` – Zwraca `getInfo()` dla wszystkich książek

Do monitorowania zajętości „slotów” książek – możesz użyć wskaźników do nich (pamiętaj o przerobieniu metod – tak aby nie mieć wycieków pamięci), albo drugiej tablicy – zajętości slotów.

W pliku main. Utwórz obiekt biblioteki, dodaj kilka książek pobierając ich dane ze standardowego wejścia. Podobnie utwórz kilka czytelników. Spróbuj pobawić się pożyczaniem / zwracaniem i metodami `displayBooks()`, `getInfo()`

Do obsługi wejścia/wyjścia użyj `<iostream>`

Zadanie bonusowe: Przerób bibliotekę tak aby korzystała z klasy `std::vector<Book>` zamiast tablicy. Obsługę wektorów znajdziesz w dokumentacji. Przyda Ci się ona pod kątem późniejszych zajęć.