|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Obraz zawierający tekst, logo, Czcionka, symbol  Opis wygenerowany automatycznie | **AKADEMIA GÓRNICZO-HUTNICZA**  **im. Stanisława Staszica w Krakowie**  **WYDZIAŁ ZARZĄDZANIA** | wz_znak | |
|  |
|  |
|  |
| Amelia Madej, Justyna Sarkowicz, Weronika Duda  Projekt zaliczeniowy z “Programowania Komputerów”  **Informatyka i Ekonometria**  **I rok, stacjonarnie** |
|  |

|  |
| --- |
| Kraków, 03.06.2023. |

### 1. Wstęp

W ramach projektu z przedmiotu "Programowanie komputerów” miałyśmy za zadanie stworzyć system obsługi biblioteki. Celem projektu było zaprojektowanie i zaimplementowanie programu umożliwiającego zarządzanie książkami oraz katalogowaniem o nich informacji.

### 2. Podział zadań

W trakcie realizacji projektu podzieliłyśmy się zadaniami w następujący sposób:

1. Amelia Madej:
   1. Zaimplementowała funkcję dodawania nowych książek do bazy danych.
   2. Opracowała strukturę danych przechowującą informacje o książkach.
   3. Przeprowadziła testy jednostkowe dla funkcji dodawania książek.
   4. Dokonała implementacji funkcji odczytu danych z bazy.
   5. Stworzyła mechanizm pozwalający na odczytanie informacji o książkach z pliku i umieszczenie ich w odpowiednich strukturach danych.
2. Justyna Sarkowicz:
   1. Zaimplementowała funkcję wyszukiwania książek na podstawie różnych kryteriów, takich jak tytuł, autor czy gatunek.
   2. Zaprojektowała interfejs użytkownika umożliwiający wprowadzenie kryteriów wyszukiwania.
   3. Przeprowadziła testy jednostkowe dla funkcji wyszukiwania.
   4. Odpowiedzialna była za implementację funkcji sortowania książek w kolejności rosnącej i malejącej.
   5. Zaprojektowała algorytmy sortowania oparte na tytule i autorze.
3. Weronika Duda:
   1. Zaimplementowała funkcję usuwania książek z bazy danych.
   2. Opracowała sposób usuwania książek na podstawie tytułu.
   3. Zaimplementowała funkcję potwierdzenia chęci usunięcia określonych danych z bazy.
   4. Przeprowadziła testy jednostkowe dla funkcji usuwania.
   5. Wykonała sprawozdanie do projektu.

### 3. Opis funkcjonalności projektu

W ramach trzyosobowego projektu stworzono program umożliwiający zarządzanie bazą książek domowej biblioteczki. Program został napisany w języku C i składa się z kilku modułów: dodawanie, odczytywanie, sortowanie, usuwanie oraz wyszukiwanie.

Projekt obejmuje następujące funkcjonalności:

* zapis książek do bazy i odczyt książek z bazy, gdzie baza ma być przechowywana w pliku binarnym,
* utworzona została przykładową bazę książek zawierającą 20 pozycji do testowania aplikacji,
* dodawanie nowych książek (tytuł, lista autorów, rok wydania, typ okładki (miękka, twarda), liczba stron, rodzaj książki (fantastyka, dramat, komedia, itp.),
* wyszukiwanie książek (po nazwisku autora, według słowa kluczowego w tytule),
* usuwanie książek z bazy (potwierdzenie),
* sortowanie książek (po autorze, po tytule),
* książki są wypisywane na konsole w formie tabeli.

### 4. Szczegółowy opis funkcji

Główny plik programu, który zawiera funkcję main(), inicjuje interfejs użytkownika. Użytkownik ma możliwość wyboru różnych opcji, takich jak dodawanie książek, odczytywanie danych z bazy, wyszukiwanie książek, sortowanie ich rosnąco lub malejąco oraz usuwanie książek z bazy.

Po wprowadzeniu nazwy pliku binarnego, program przechodzi do pętli, która wykonuje się, dopóki użytkownik nie wybierze opcji wyjścia. W każdym obiegu pętli, na podstawie wyboru użytkownika, program wywołuje odpowiednie funkcje zaimplementowane w innych modułach.

Zastosowane w projekcie parametry funkcji to: “nazwa”, czyli nazwa pliku bazy danych oraz “nowa” będąca wskaźnik na tablicę struktur, do której zostaną zapisane odczytane dane.

Program został podzielony na moduły w celu zachowania czytelności i łatwiejszego zarządzania kodem. Dzięki zastosowaniu modularnej struktury kodu, program jest łatwy do rozbudowy i modyfikacji. Każdy moduł zawiera odpowiednie funkcje, które wykonują określone zadania związane z zarządzaniem bazą książek.

* **Moduł "dodawanie"**

Moduł "dodawanie" zawiera funkcję "dodaj", która pozwala użytkownikowi wprowadzić dane dotyczące nowej książki i zapisuje te dane w pliku bazy danych.

Najpierw tworzy tablicę dynamiczną struktur "nowa" o rozmiarze MAX\_BOOKS, przechowującą informacje o nowych książkach. Potem otwiera plik bazy danych w trybie dołączania ("ab"). Jeśli operacja się nie powiedzie, zwraca błąd. Rozpoczyna pętlę do-while, która pozwala na dodawanie wielu książek.

W każdym przebiegu pętli pyta użytkownika o jedna z atrybutów książki i zapisuje go w odpowiednim polu struktury "nowa[i]".

- Zapisuje strukturę "nowa[i]" do pliku bazy danych.

- Pyta użytkownika, czy chce dodać kolejną pozycję. Jeśli wybór to 0, kończy pętlę.

Następnie zamyka plik bazy danych i wyświetla informację o pomyślnym dodaniu książki.

* **Moduł "odczytywanie"**

Moduł "odczytywanie" zawiera funkcję "odczyt", która odczytuje dane z bazy książek i wyświetla je użytkownikowi.

Najpierw otwiera plik bazy danych w trybie odczytu binarnego ("rb"). Jeśli operacja się nie powiedzie, zwraca błąd. Inicjalizuje zmienne pomocnicze: i - indeks aktualnej książki, num - numer pozycji na liście. Następnie wyświetla nagłówek tabeli z informacjami o książkach.

W pętli while odczytuje kolejne struktury z pliku i wyświetla ich dane.

- Wyświetla numer pozycji, tytuł, autorów, rok wydania, rodzaj okładki, liczbę stron i gatunek książki.

- Tworzy łańcuch znakowy "autorzy\_string" zawierający imiona i nazwiska autorów, oddzielone przecinkami.

Na końcu zamyka plik bazy danych.

* **Moduł "sortowanie"**

Moduł "sortowanie" zawiera dwie funkcje: "sortuj\_rosnaco" i "sortuj\_malejaco", które sortują książki w bazie danych według określonego kryterium (np. tytułu, autora) rosnąco lub malejąco.

Najpierw otwierany jest plik bazy danych w trybie odczytu binarnego ("rb"). Jeśli operacja się nie powiedzie, zwraca błąd. Pobiera rozmiar pliku i oblicza liczbę struktur w pliku. Odczytuje struktury z pliku i zapisuje je w tablicy. Następnie zamyka plik bazy danych. Wyświetla menu wyboru opcji sortowania.

W zależności od wybranej opcji sortuje tablicę struktur.

- Opcja 1: Sortowanie po tytule

- Opcja 2: Sortowanie po autorze

- Jeśli wybrana opcja jest niepoprawna, zwraca błąd.

Potem zapisuje je do pliku i zamyka plik bazy danych.

* **Moduł "usuwanie"**

Moduł "usuwanie" zawiera funkcję "usun", która pozwala użytkownikowi usunąć wybraną książkę z bazy danych.

Na początku otwiera plik bazy danych w trybie odczytu binarnego ("rb"). Jeśli operacja się nie powiedzie, zwraca błąd. Tworzy tymczasowy plik "temp.csv" w trybie zapisu binarnego ("wb"). Jeśli operacja się nie powiedzie, zwraca błąd. Kolejno pobiera tytuł książki do usunięcia od użytkownika.

Następnie przechodzi przez plik bazowy, odczytuje struktury i sprawdza, czy tytuł jest zgodny.

- Jeśli tytuł jest zgodny, ustawia flagę "found" na 1.

- Jeśli tytuł nie jest zgodny, zapisuje strukturę do tymczasowego pliku.

Zamyka pliki bazowy i tymczasowy. Jeśli flaga "found" jest ustawiona na 1:

- Otwiera tymczasowy plik "temp.csv" w trybie odczytu binarnego ("rb"). Jeśli operacja się nie powiedzie, zwraca błąd.

- Otwiera plik bazowy "baza.csv" w trybie zapisu binarnego ("wb"). Jeśli operacja się nie powiedzie, zwraca błąd.

- Przechodzi przez tymczasowy plik, odczytuje struktury i zapisuje je do pliku bazowego.

- Zamyka pliki bazowy i tymczasowy.

- Wyświetla komunikat o usunięciu książki.

Jeśli flaga "found" jest ustawiona na 0, wyświetla komunikat o nieznalezieniu książki w bazie.

* **Moduł "wyszukiwanie"**

Moduł "wyszukiwanie" zawiera funkcję "wyszukaj", która umożliwia użytkownikowi wyszukiwanie książek w bazie danych na podstawie określonych kryteriów (np. tytułu, autora) i wyświetla pasujące wyniki.

Najpierw otwiera plik bazy danych w trybie odczytu binarnego ("rb"). Jeśli operacja się nie powiedzie, zwraca błąd. Następnie inicjalizuje zmienne pomocnicze: found - licznik znalezionych książek, num - numer pozycji na liście. Wczytuje słowo kluczowe od użytkownika. Wyświetla nagłówek tabeli z informacjami o książkach.

W pętli while odczytuje kolejne struktury z pliku i sprawdza, czy zawierają podane słowo kluczowe.

- Jeśli tytuł / rodzaj książki zawiera słowo kluczowe, wyświetla jej informacje.

- Aby sprawdzić, czy któryś z autorów zawiera słowo kluczowe, wykorzystuje funkcję pomocniczą znalezieni\_autorzy.

Na końcu zamyka plik bazy danych, a jeśli nie znaleziono żadnej książki, wyświetla odpowiedni komunikat.

### 5. Posumowanie

Podczas realizacji poszczególnych zadań w projekcie zdobyłyśmy cenne doświadczenie w programowaniu w języku C oraz w pracy z bazami danych. Testy jednostkowe odegrały kluczową rolę w zapewnieniu poprawności i niezawodności naszego systemu. W przyszłości istnieje wiele możliwości rozwoju projektu, takich jak np. implementacja interfejsu graficznego. Obecnie projekt oparty jest na wierszu poleceń, ale można rozważyć rozwinięcie go o interfejs graficzny, co zwiększyłoby jego atrakcyjność i użyteczność dla użytkowników. Jednak na ten moment uważamy, że nasz projekt spełnia założone cele i może być wykorzystany jako podstawa do rozbudowy i ulepszenia systemu obsługi prostej bazy danych.