

1. Wstęp

Projekt składa się z następujących rozdziałów:

1. Wstęp - opisuje ogólny zarys projektu.
2. Opis projektu - przedstawia główne funkcje i klasy wykorzystane w projekcie.
3. Narzędzia - omawia narzędzia użyte do tworzenia projektu.
4. Opis dla użytkownika - zawiera informacje dotyczące uruchomienia projektu i prezentuje zrzuty ekranu.
5. Podsumowanie - podsumowuje projekt i prezentuje wnioski.

2. Opis projektu

Projekt to program symulujący działanie biblioteki. Po pierwsze program jest podzielony na 4 menu do wyboru: Dyrektor, Pracownik, Klient i Klient Premium. Do wyeksponowania tych 4 typów użytkowników użyłem typu Enum. Każdy z tych użytkowników ma inne menu i inne opcje do wyboru odpowiednio dostosowane do pełnionych funkcji. Program wykorzystuje kilka klas, takich jak Książka, Pracownik i Biblioteka.

Klasa Książka reprezentuje pojedynczą książkę w bibliotece i zawiera metody umożliwiające wypożyczenie, oddanie i zakup książki oraz metodę czasDooddania, która po wypożyczeniu książki przez użytkownika zaczyna odliczać czas, który mu pozostał na oddanie książki do biblioteki. Ta informacja wyświetla się w menu Klienta Premium. Dodatkowo Klasa Książka posiada takie atrybuty jak: tytuł, autor, rok wydania, wydawnictwo, liczba stron, bool wypożyczona, bool zakupiona, bool możeBycWypożyczona. Dodatkowo klasa zawiera konstruktor sparametryzowany do tworzenia instancji tej klasy, a w argumentach konstruktora są kolejno zawarte: tytuł, autor, rok wydania, wydawnictwo, liczba stron.

Klasa Pracownik reprezentuje pracownika biblioteki i przechowuje informacje o jego imieniu, nazwisku, wynagrodzeniu, stazu pracy, wykształceniu i Peselu. Klasa pracownik również zawiera konstruktor sparametryzowany, który jako argumenty przyjmuje kolejno: imię, nazwisko, wynagrodzenie, staz pracy, wykształcenie i pesel.

Klasa Biblioteka zarządza kolekcją książek i pracowników oraz udostępnia różne funkcje, takie jak dodawanie i usuwanie książek, zatrudnianie i zwalnianie pracowników, wypożyczanie i oddawanie książek.

W klasie Biblioteka są utworzone 3 wektory do przechowywania obiektów klasy Książka i klasy Pracownik. Klasa konkretnie zawiera takie metody jak: dodaj książkę, czyli można w ten sposób dodać książkę do biblioteki, dodaj pracownika-dyrektor ma opcje zatrudnienia nowego pracownika, metoda zwolnijPracownika-dyrektor ma opcje zwolnienia pracownika biblioteki, metoda wyświetl obecnych pracowników-pozwala to dyrektorowi na podgląd obecnie zatrudnionych pracowników, metoda znadzKsiążkePoAutorze- ta metoda pozwala użytkownikowi premium na znalezienie książki po nazwisku autora, metoda znajdźKsiążke odnajduje książkę po tytule, metoda wysietylKsiążki pozwala na przegląd wszystkich dostępnych książek w bibliotece, metoda zapiszDopliku pozwala na zapisanie wszystkich informacji z programu w dodatkowym pliku, metoda odczytajzpliku odczytuje informacje

znajdujące się w pliku, metoda `usunKsiazke` pozwala dyrektorowi na usunięcie zbędnych książek z biblioteki, metoda `oddajKsiazke` pozwala użytkownikom oddać wypożyczoną książkę do biblioteki, metoda `zakupKsiazke` jest dostępna tylko dla użytkowników premium i pozwala na wykupienie książki z biblioteki, metoda `wyswietlWypozyczoneKsiazki` wyświetla użytkownikowi premium wszystkie książki, które są aktualnie przez niego wypożyczone, metoda `wyswietlZakupioneKsiazki` wyświetla wszystkie książki, które zostały zakupione przez użytkownika premium.

W funkcji `main` odbywa się instancja 2 przykładowych pracowników, korzystając z utworzonego konstruktora sparametryzowanego dla klasy `Pracownik`.

Oraz odbywa się instancja 6 przykładowych książek, korzystając z utworzonego konstruktora sparametryzowanego dla klasy `Ksiazka`.

3. Narzędzia

Do tworzenia projektu został wykorzystany język C++ oraz środowisko programistyczne umożliwiające pisanie i kompilowanie kodu w tym języku, Visual Studio Code.

Visual Studio Code to darmowy edytor kodu źródłowego z kolorowaniem składni dla wielu języków, stworzony przez Microsoft, o otwartym kodzie źródłowym. Oprogramowanie ma wsparcie dla debugowania kodu, zarządzania wersjami kodu źródłowego za pośrednictwem systemu kontroli wersji Git, automatycznego uzupełniania kodu IntelliSense, zarządzania wycinkami kodu oraz jego refaktoryzacji. Funkcjonalność aplikacji można rozbudować za pomocą rozszerzeń instalowanych z dedykowanego repozytorium rozszerzeń.

Oprogramowanie zostało stworzone w oparciu o framework Electron.

C++ jest popularnym językiem programowania, który oferuje wiele narzędzi i bibliotek do tworzenia aplikacji. W projekcie zostały również wykorzystane biblioteki standardowe C++ takie jak `iostream` i `fstream` do obsługi wejścia/wyjścia oraz operacji na plikach.

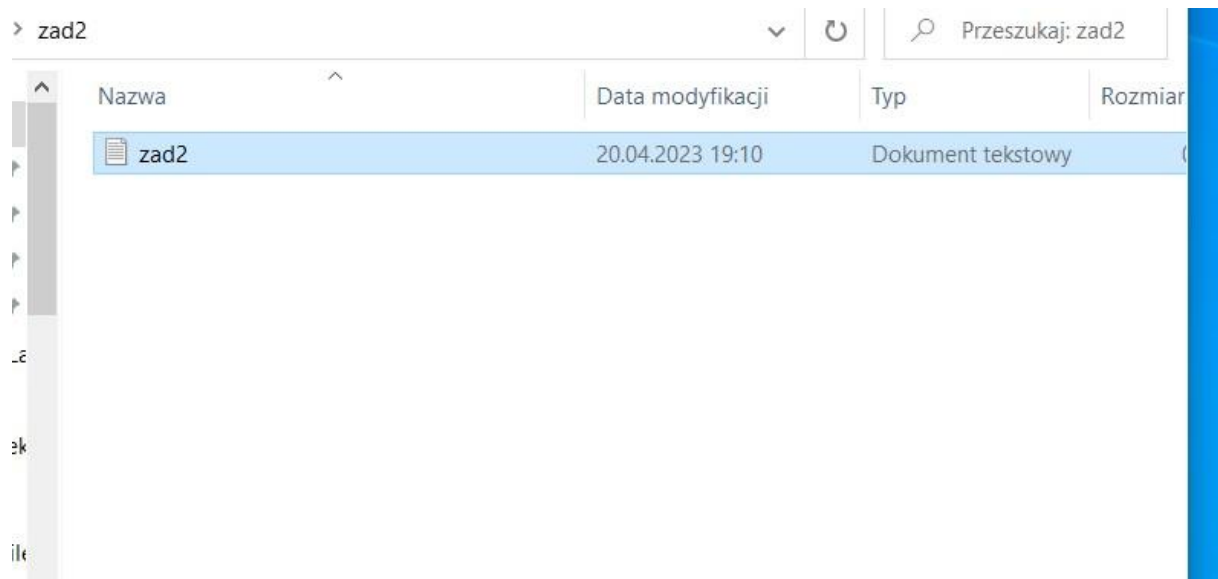
4. Opis dla użytkownika

Aby uruchomić taki kod należy kolejno postępować zgodnie z poniższą instrukcją i załączonymi screenami.

1. Należy utworzyć folder o dowolnej nazwie w dowolnym miejscu w pamięci komputera.



2. Wewnątrz tego folderu należy utworzyć nowy notatnik o dowolnej nazwie.



3. Następnie należy wkleić cały napisany kod do wnętrza naszego notatnika.

```
kod (2) — Notatnik
Plik Edycja Format Widok Pomoc
#include <iostream>
#include <string>
#include <vector>
#include <fstream>
#include <exception>
#include <chrono>

using namespace std::chrono;
using namespace std;

class Ksiazka {
    string tytul;
    string autor;
    int rok_wydania;
    string wydawnictwo;
    int liczba_stron;
    bool wypożyczona;
    chrono::system_clock::time_point czas_wypożyczenia;

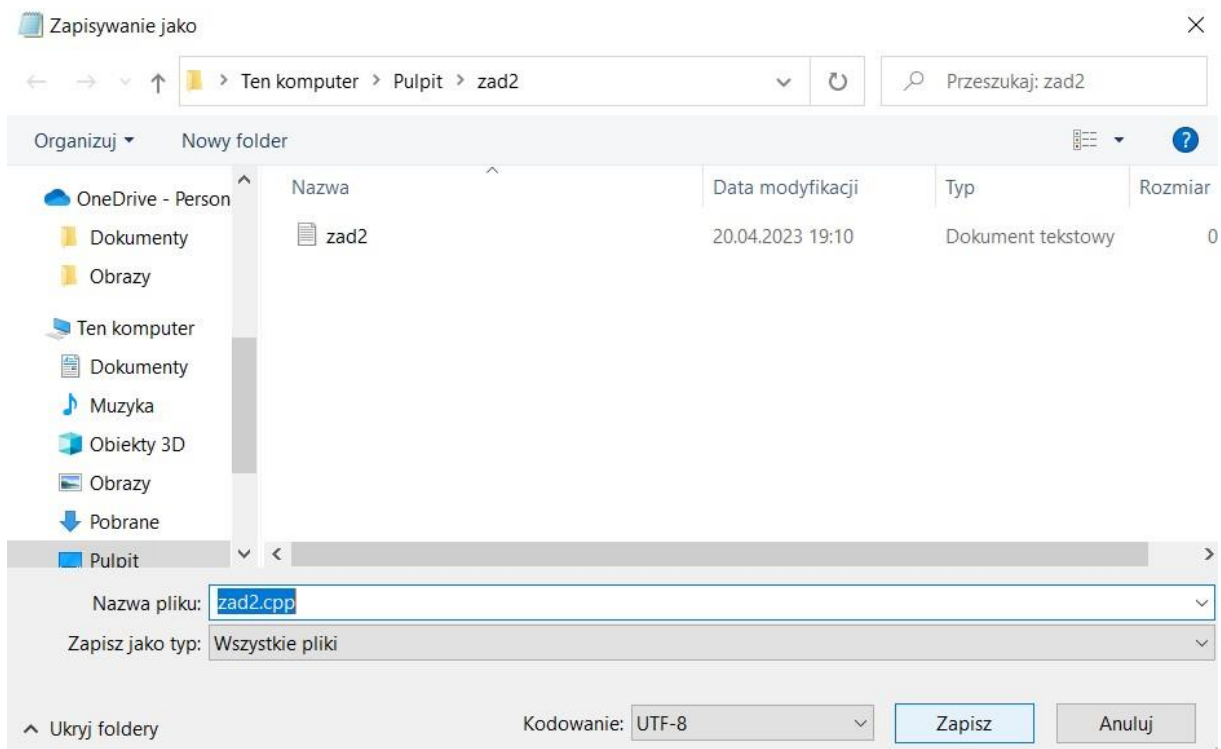
public:
    Ksiazka(string t, string a, int r, string w, int ls)
        : tytul(t), autor(a), rok_wydania(r), wydawnictwo(w), liczba_stron(ls) {}

    void wypożycz() {
        wypożyczona = true;
        czas_wypożyczenia = chrono::system_clock::now();
    }

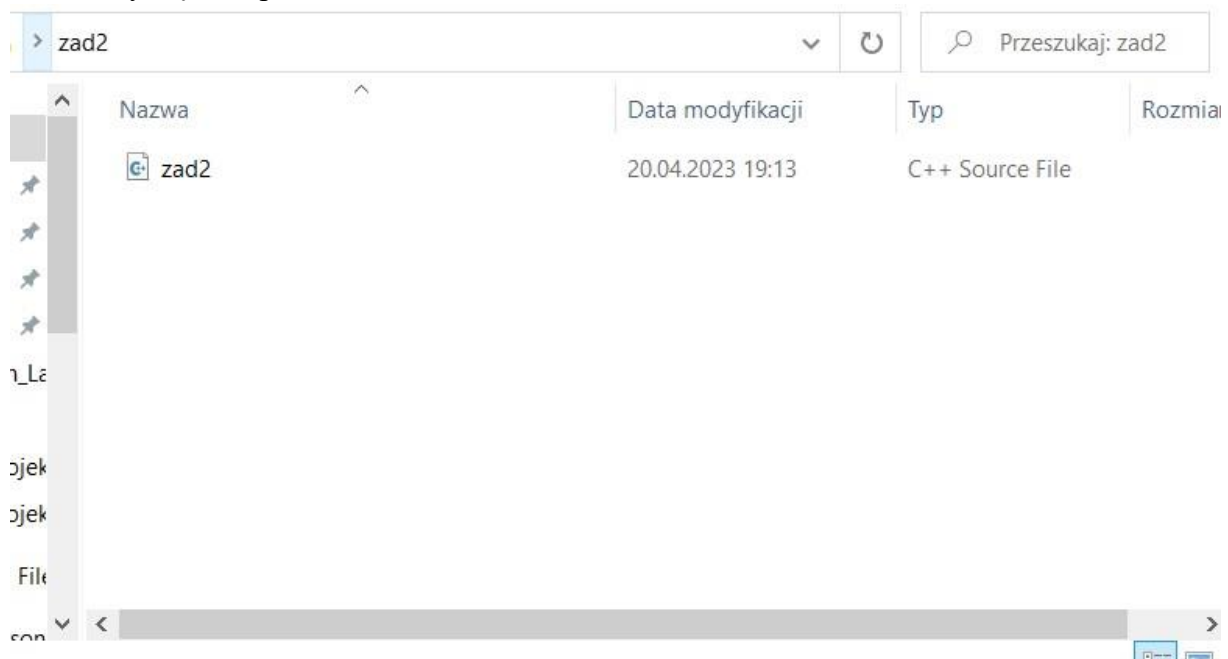
    void oddaj() {
        wypożyczona = false;
    }

    bool czyWypożyczona() const {
        return wypożyczona;
    }
}
```

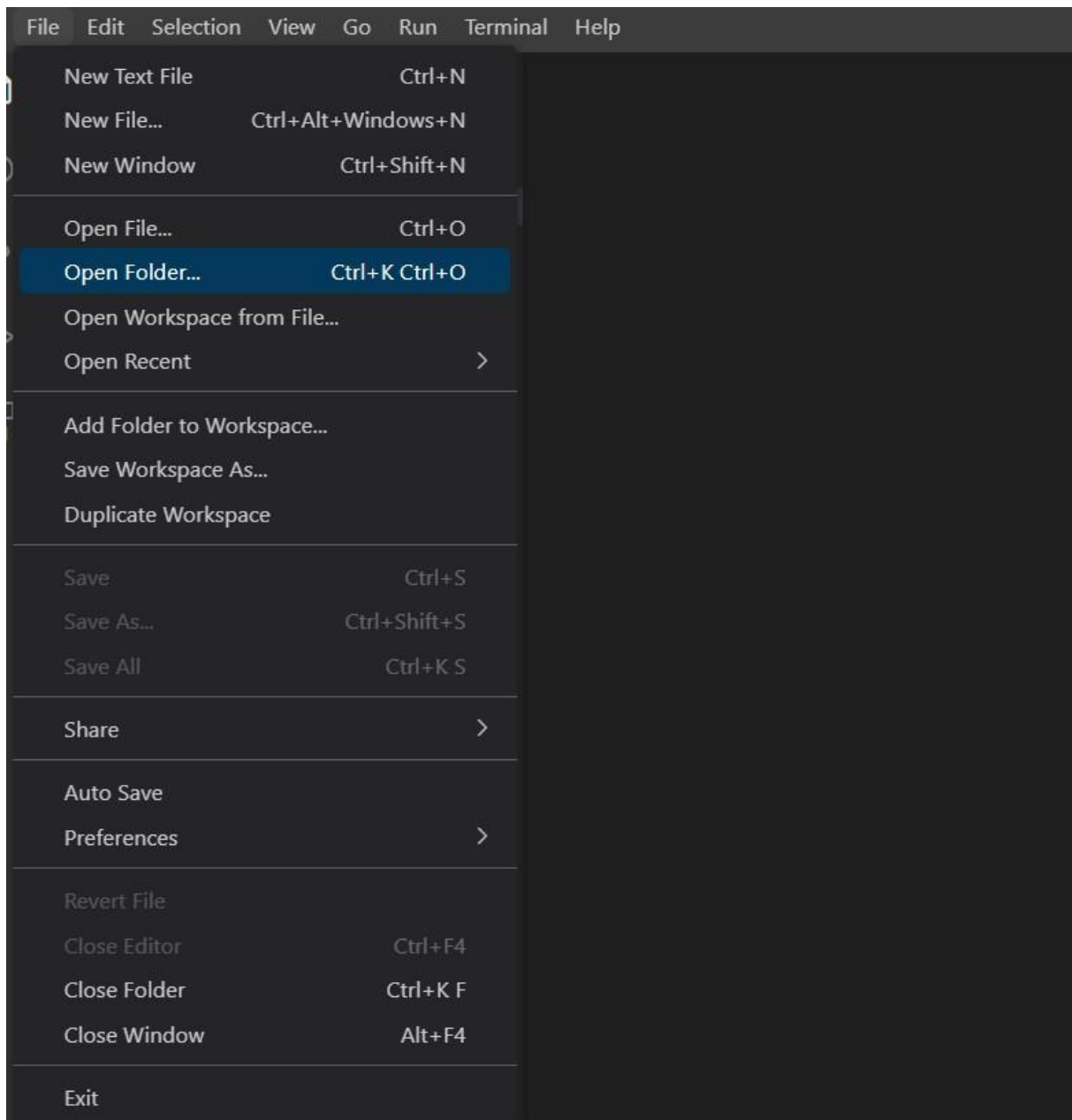
4. Następnie należy dać zapisz jako i dopisać do nazwy pliku .cpp i zmienić poniżej na opcje wszystkie pliki.



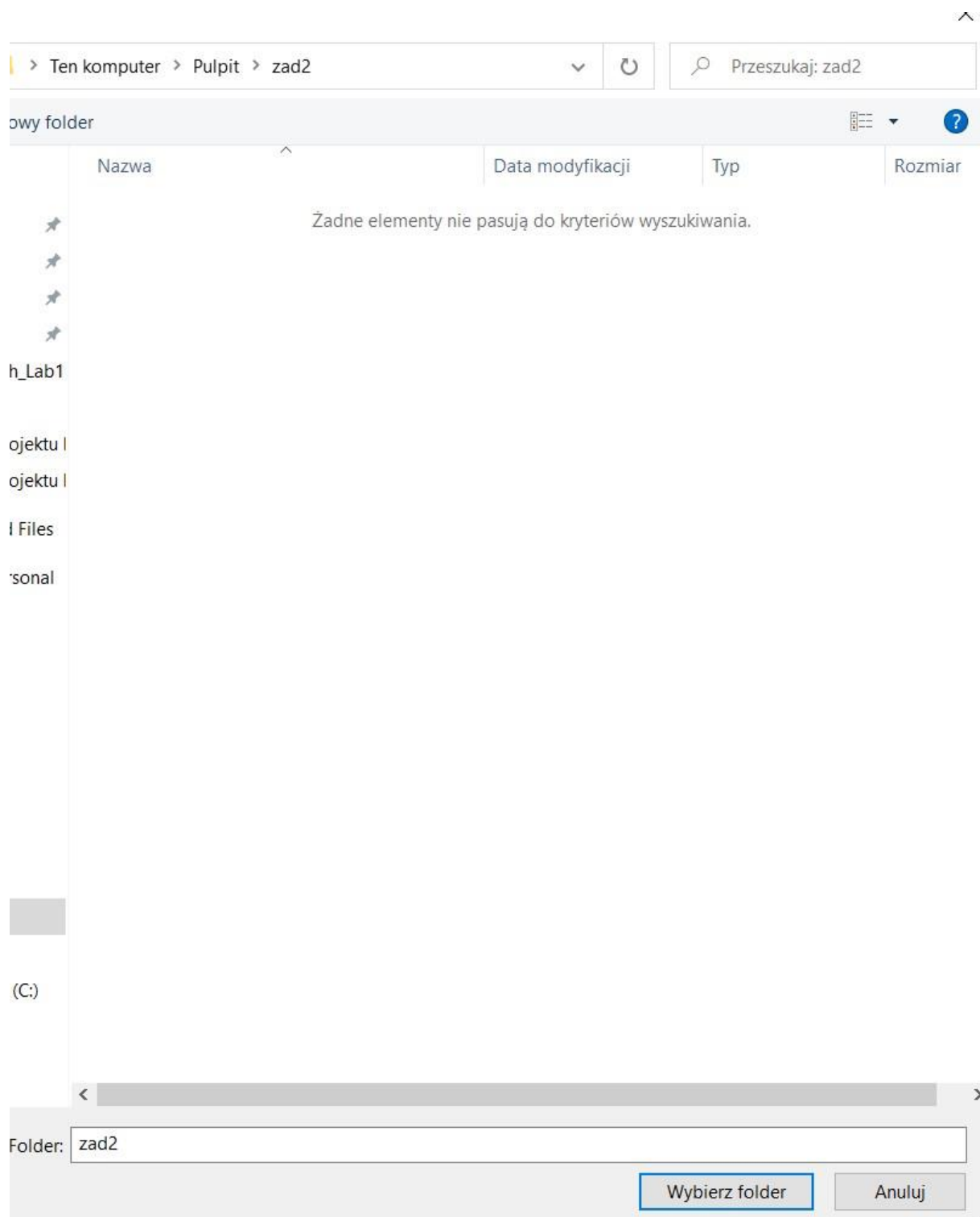
5. Utworzy się nam plik o rozszerzeniu dla C++.



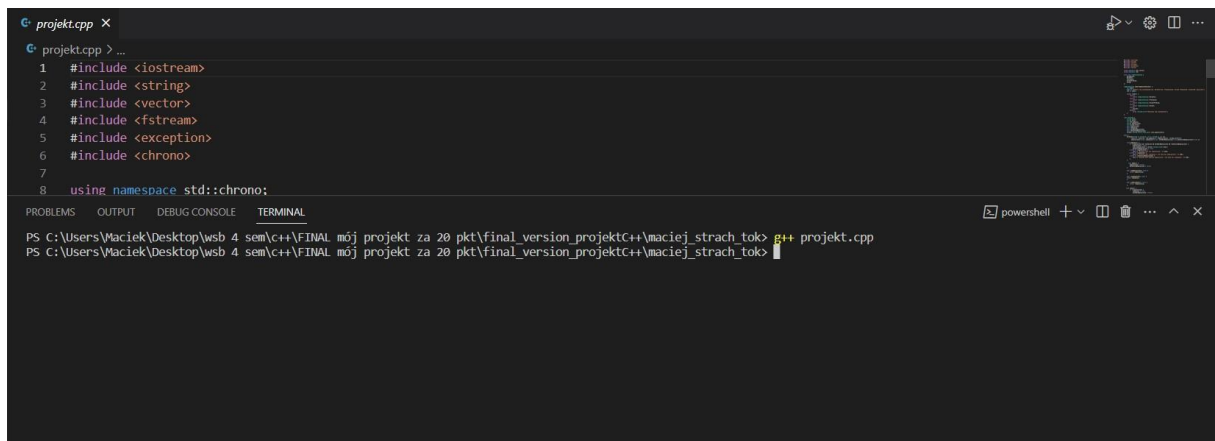
6. Następnie wchodzimy w Visual Studio Code i dajemy opcje open folder i wyszukujemy folder z naszym plikiem.



7. Wybieramy go.



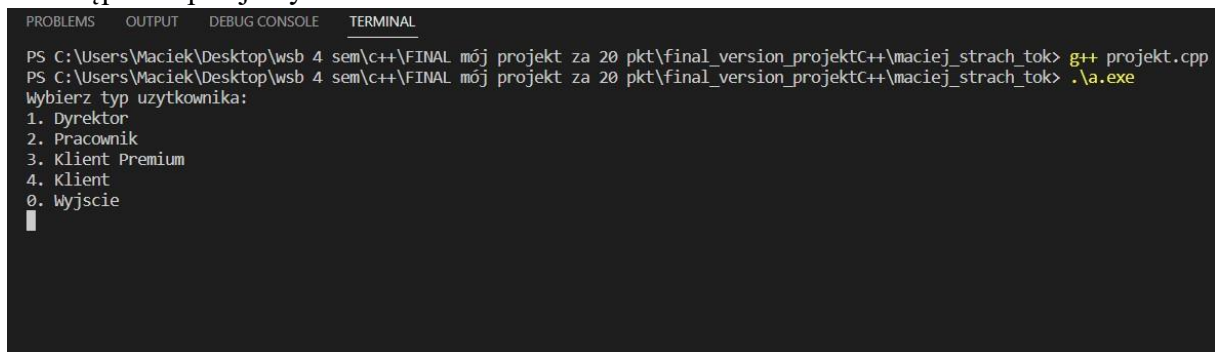
8. Wewnątrz Visual Studio Code dajemy `ctrl+`` i wpisujemy `g++ nazwanaszegopliku.cpp`.



The screenshot shows the Visual Studio IDE. The top pane displays the source code for `projekt.cpp`, which includes standard C++ headers and a `using namespace std::chrono;` declaration. The bottom pane shows the `TERMINAL` window with the following commands and output:

```
PS C:\Users\Maciek\Desktop\wsb 4 sem\c++\FINAL mój projekt za 20 pkt\final_version_projektC++\maciej_strach_tok> g++ projekt.cpp
PS C:\Users\Maciek\Desktop\wsb 4 sem\c++\FINAL mój projekt za 20 pkt\final_version_projektC++\maciej_strach_tok> .\a.exe
```

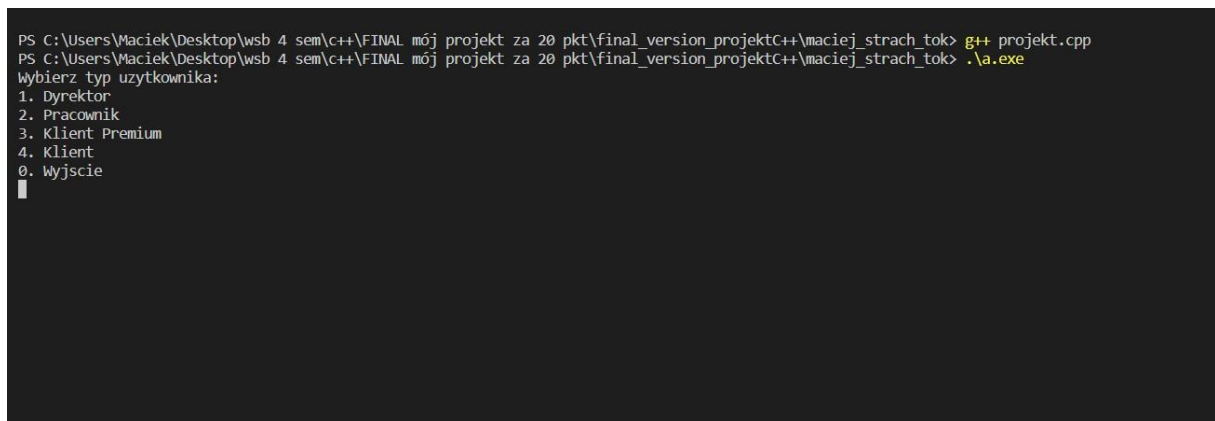
9. Następnie wpisujemy `.\a.exe`



The screenshot shows the terminal window after running the program. The output is as follows:

```
PS C:\Users\Maciek\Desktop\wsb 4 sem\c++\FINAL mój projekt za 20 pkt\final_version_projektC++\maciej_strach_tok> g++ projekt.cpp
PS C:\Users\Maciek\Desktop\wsb 4 sem\c++\FINAL mój projekt za 20 pkt\final_version_projektC++\maciej_strach_tok> .\a.exe
Wybierz typ uzytkownika:
1. Dyrektor
2. Pracownik
3. Klient Premium
4. Klient
0. Wyjscie
█
```

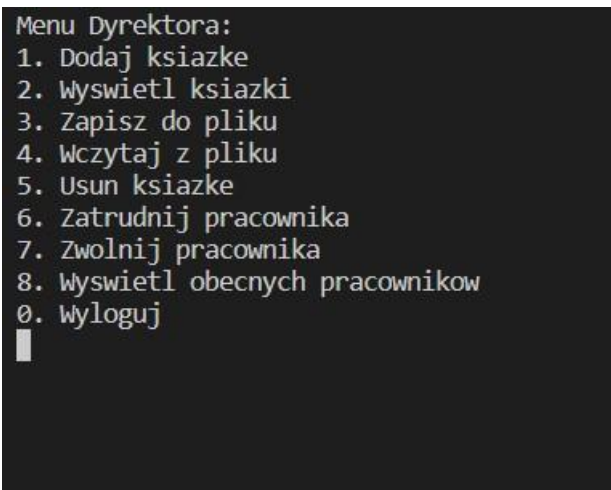
MENU GŁÓWNE PROJEKTU



The screenshot shows the terminal window displaying the main menu of the program:

```
PS C:\Users\Maciek\Desktop\wsb 4 sem\c++\FINAL mój projekt za 20 pkt\final_version_projektC++\maciej_strach_tok> g++ projekt.cpp
PS C:\Users\Maciek\Desktop\wsb 4 sem\c++\FINAL mój projekt za 20 pkt\final_version_projektC++\maciej_strach_tok> .\a.exe
Wybierz typ uzytkownika:
1. Dyrektor
2. Pracownik
3. Klient Premium
4. Klient
0. Wyjscie
█
```

MENU DYREKTORA



The screenshot shows the terminal window displaying the menu for the Director user type:

```
Menu Dyrektora:
1. Dodaj ksiazke
2. Wyswietl ksiazki
3. Zapisz do pliku
4. Wczytaj z pliku
5. Usun ksiazke
6. Zatrudnij pracownika
7. Zwolnij pracownika
8. Wyswietl obecnych pracownikow
0. Wyloguj
█
```


MENU PRACOWNIKA

```
Menu Pracownika:  
1. Dodaj książkę  
2. Wyświetl książki  
3. Zapisz do pliku  
4. Wczytaj z pliku  
0. Wyloguj
```

MENU KLIENTA PREMIUM

```
Menu Klienta Premium:  
1. Wypożycz książkę  
2. Wyświetl książki  
3. Zapisz do pliku  
4. Wczytaj z pliku  
5. Oddaj książkę  
6. Zakup książkę  
7. Wyświetl wypożyczone książki  
8. Znajdź książkę po tytule  
9. Znajdź książkę po autorze  
10. Wyświetl zakupione książki  
0. Wyloguj
```

MENU KLIENTA

```
Menu Klienta:  
1. Wypożycz książkę  
2. Wyświetl książki  
3. Zapisz do pliku  
4. Wczytaj z pliku  
5. Oddaj książkę  
6. Wyświetl wypożyczone książki  
0. Wyloguj
```

5.Podsumowanie

Celem projektu było stworzenie programu symulującego działanie biblioteki i umożliwienie użytkownikowi interakcji z różnymi funkcjonalnościami, takimi jak dodawanie i usuwanie książek, wypożyczanie i oddawanie książek, zakup książek, zatrudnianie i zwalnianie pracowników biblioteki, zapisywanie i wczytywanie danych z pliku itp. Projekt ten miał na celu zaprezentowanie podstawowych operacji bibliotecznych oraz dostarczenie środowiska do testowania różnych scenariuszy.

Podsumowując, projekt biblioteki pozwala na lepsze zrozumienie i eksplorację funkcji i operacji wykonywanych w bibliotece, co może być przydatne zarówno dla osób zainteresowanych bibliotekami, jak i programistów, którzy chcą rozwijać i doskonalić systemy zarządzania bibliotekami.