|  |  |
| --- | --- |
|  | **AKADEMIA GÓRNICZO-HUTNICZA** |

Dokumentacja do projektu

**AirControl**

z przedmiotu

**Języki programowania obiektowego**

Elektronika II rok

*Bartosz Rychlicki*

Poniedziałek 16:45

prowadzący: Rafał Frączek

29.01.2025r.

# Opis projektu

Projekt pełni funkcję systemu kontroli lotów. Umożliwia on użytkownikowi zarządzanie danymi lotami w obrębie danego lotniska. Użytkownik z pozycji programu posiada również możliwość zmiany danych lotniska oraz danych poszczególnych lotów. Po programie użytkownik porusza się za pomocą terminala, przechodząc między różnymi menu oraz submenu.

# Project description

# The project acts as a flight control system. It allows the user to manage flight data within a given airport. The user also has the option of changing airport data and individual flight data from the program. The user navigates the program using a terminal, switching between different menus and submenus..

# Instrukcja użytkownika

Poruszanie się pomiędzy interfejsami jest sygnalizowane poprzez komunikaty. Aby wybrać moduł/opcję, którą to chcemy uruchomić wystarczy wpisać numer operacji. Aby zmienić pewne nazewnictwo różnych parametrów zaleca się używanie tylko i wyłącznie liter, jeśli natomiast został uruchomiony moduł zmiany dat, godziny itp. Należy natomiast stosować się do formatu wyświetlanego w komunikatach. Pod żadnym pozorem nie zaleca się przerywania działania programu z pozycji kompilatora, można to natomiast wykonać poruszając się po menu wychodząc z poszczególnych podmenu do menu głównego , stamtąd natomiast wpisując odpowiednią cyfrę reprezentującą moduł zamknięcia programu.

# Kompilacja

Kompilacja przebiega standardowo, wystarczy uruchomić kompilację z pozycji kompilatora. Kompilacja działa jedynie na platformie Windows.

# Pliki źródłowe

Projekt składa się z następujących plików źródłowych:

* *source.cpp* – plik startowy, tutaj następuję ładowanie danych z bazy danych ,
* *Controller.cpp* – plik posiadający definicję metod, getterów i seterów oraz konstruktora klasy Controller
* *Database.cpp* – plik posiadający definicję metod oraz konstruktora klasy Database,
* *Fligh*.cpp – plik posiadający definicję metod , seterów oraz getterów i konstruktora klasy Flight,
* *Airport*.cpp – plik posiadający definicje metod, getterów i seterów oraz konstruktora klasy Airport.

# Pliki nagłówkowe

* Airport.hpp – posiada ciało klasy Airport wraz z deklaracjami jej metod i pól
* Controller.hpp - posiada ciało klasy Controller wraz z deklaracjami jej metod i pól
* Database.hpp - posiada ciało klasy Database wraz z deklaracjami jej metod i pól
* Flight.hpp - posiada ciało klasy Flight wraz z deklaracjami jej metod i pól
* IAirport.hpp – plik posiadający interfejs klasy Airport
* IController.hpp – plik posiadający interfejs klasy Controller
* IDatabase.hpp - plik posiadający interfejs klasy Database
* IFLight.hpp - plik posiadający interfejs klasy Flight
* ICompress.hpp – posiada interfejs. Metoda zawarta w tym interfejsie wykorzystywana jest w klasach: Airport oraz Flight
* IController\_Airport.hpp – posiada interfejs, którego metody są wykorzystywane w klasie Controller
* IPrint.hpp – posiada interfejs. Metoda zadeklarowana w interfejsie wykorzystywana jest w wielu klasach celem wyświetlenia danych dot. danego obiektu danej klasy
* Validation.hpp – prócz klasy zapisanej w według wzorca Singleton, w tym pliku znajdują się również funkcje pomocnicze, takie jak: getCurrentTime(). Klasa Singleton posiada metody wykorzystywane do przeprowadzania walidacji poprawności niektórych danych, ich typu lub też formatu.

# Zależności

Projekt wykorzystuje dodatkowe bilbioteki: sqlext.h, sql.h celem implementacji konekcji z bazą danych SQL.

# Zasoby

W projekcie nie wykorzystywane są dodatkowe zasoby.

# Dalszy rozwój i ulepszenia

W dalszym rozwoju programu można zaimplementować graficzny interfejs użytkownika. Można również dokonać rozszerzenia ilości informacji zawartej w bazie danych.

# Inne

Brak