

TYTUŁ PROJEKTU

Aplikacja: **MÓJ PLAN ZAJĘĆ SZKOLNYCH**

CEL PROJEKTU

Projekt ma na celu stworzenie aplikacji konsolowej, która będzie pozwalała na odczytywanie i zapisywanie do pliku **data.txt** plan zajęć szkolnych ucznia.

WYMAGANIA FUNKCJONALNE

- Program posiada menu wyboru z opcjami:
 - **Odczytaj** – odczytuje plan zajęć z podanego dnia tygodnia
 - **Edytuj** – pozwala na dokonanie korekty istniejącego planu
 - **Zapisz** – pozwala wprowadzać nowe dane do planu
 - **Pomoc** – pozwala uzyskać informacje na temat działania programu
 - **Zakończ** – kończy działanie programu
- **Odczytaj** – program umożliwia użytkownikowi wybór dnia tygodnia, na którego podstawie wyświetlany jest plan dnia w formacie:
 - 7.10-7.55 Język polski
 - 8.00-8.45 Matematyka
- **Edytuj** - program umożliwia użytkownikowi wybór dnia tygodnia oraz numeru lekcji, a następnie na podstawie dokonanego wyboru użytkownik wprowadza zmiany na wybranej godzinie lekcyjnej
- **Zapisz** - program umożliwia użytkownikowi wybór dnia tygodnia, a następnie na podstawie wyboru numeru lekcji z dostępnych zajęć użytkownik dopisuje nowe dane w formacie:
 - 07.10-07.55 Język polski
- **Usuń** – program umożliwia użytkownikowi wybór dnia tygodnia oraz godziny i na podstawie dokonanego wyboru użytkownik usuwa wskazane zajęcia
- **Pomoc** – program umożliwia uzyskanie podstawowych informacji na temat działania aplikacji
- **Zakończ** – program umożliwia użytkownikowi zakończyć działanie programu

WYMAGANIA NIEFUNKCJONALNE

- **Interfejs użytkownika** - interfejs użytkownika powinien być prosty i intuicyjny, umożliwiając użytkownikowi łatwe korzystanie z planu zajęć
- **Wydajność** – program powinien działać sprawnie, zapewniając natychmiastowe odpowiedzi na wprowadzone dane
- **Bezpieczeństwo** - aplikacja powinna obsługiwać błędy, takie jak próba wyboru nieprzewidzianego dnia tygodnia, numeru lekcji, próba

ZALECENIA TECHNICZNE

- Język programowania: C++
- Interfejs użytkownika: konsola/ wiersz poleceń
- Struktura kodu: zastosowanie funkcji

PLAN ROZWOJU

- **Etap 1** - podstawowa struktura aplikacji: utworzenie funkcji do obsługi podstawowych operacji matematycznych
- **Etap 2** - implementacja interfejsu użytkownika
- **Etap 3** - testowanie
- **Etap 4** - optymalizacja i poprawki

HARMONOGRAM

- **Etap 1** - 1 godzina
- **Etap 2** - 1 godzina
- **Etap 3** - 1 godzina
- **Etap 4** - 1 godzina

ZESPÓŁ

- Programista
- Tester