



AKADEMIA GÓRNICZO-HUTNICZA

Dokumentacja do projektu

Task Manager

z przedmiotu

Języki Programowania Obiektowego

Elektronika i Telekomunikacja, Rok 3

Hubert Durnaś

piątek 15:00

prowadzący: mgr inż. Jakub Zimnol

09.01.2025

1. Opis projektu

Task Manager

Task Manager to aplikacja terminalowa napisana w języku C++, która umożliwia zarządzanie listą zadań. Aplikacja pozwala na dodawanie, usuwanie, edytowanie oraz oznaczanie zadań jako wykonane. Dodatkowo aplikacja zapisuje i wczytuje listę zadań z pliku, co pozwala na zachowanie danych między sesjami.

Zastosowania

Aplikacja Task Manager może być używana do zarządzania codziennymi zadaniami, projektami lub listami zadań w pracy. Dzięki prostemu interfejsowi terminalowemu, użytkownik może łatwo i szybko zarządzać swoimi zadaniami.

2. Opis klas zaimplementowanych w projekcie

Klasa Task

Klasa Task reprezentuje pojedyncze zadanie na liście. Zawiera następujące atrybuty i metody:

- `std::string description` - opis zadania.
- `bool done` - status zadania (wykonane lub nie).
- Konstruktory:
 - `Task()` - domyślny konstruktor.
 - `Task(const std::string& description, bool done = false)` - konstruktor parametryczny.
- Przeciążone operatory:
 - `bool operator==(const Task& other) const` - operator porównania.
 - `bool operator!=(const Task& other) const` - operator nierówności.

Klasa TaskManager

Klasa TaskManager zarządza listą zadań i zawiera następujące atrybuty i metody:

- `std::vector<Task> tasks` - wektor zadań.
- `std::string fileName` - nazwa pliku do zapisywania/ladowania zadań.
- Metody:
 - `TaskManager(const std::string& fileName)` - konstruktor.
 - `~TaskManager()` - destruktor.
 - `void addTask(const std::string& description)` - dodaje nowe zadanie.
 - `void deleteTask(int index)` - usuwa zadanie po indeksie.
 - `void editTask(int index, const std::string& newDescription)` - edytuje opis zadania.
 - `void markAsDone(int index)` - oznacza zadanie jako wykonane.
 - `void displayTasks() const` - wyświetla wszystkie zadania.
 - `void menu()` - wyświetla menu zarządzania zadaniami.

- Przeciążone operatory:

Task& operator[](int index) - operator indeksowania (dostęp do zadania).

const Task& operator[](int index) const - const operator indeksowania (dostęp do zadania).

3. Opis uruchomienia/skompilowania aplikacji

Aby skompilować aplikację, należy skopiować i skompilować kody programów do kompilatora c++.

4. Dodatkowe informacje

W aplikacji użyto przestrzeni nazw `hd` do organizacji kodu. Dokumentacja w stylu Doxygen została dodana do plików kodu źródłowego, aby ułatwić zrozumienie i rozwijanie projektu.