

Opis Programu

Klasy:

1. **kl (klasa lekcyjna):**

- Atrybuty:
 - `klasa` (string): oznaczenie klasy lekcyjnej.
 - `group` (string): oznaczenie grupy w klasie.
 - `subject` (string): nazwa przedmiotu.
 - `teacher` (string): imię i nazwisko nauczyciela.
 - `hForWeek` (int): liczba godzin lekcyjnych w tygodniu dla danego przedmiotu.
- Konstruktor inicjalizuje obiekt klasy `kl` przyjmując parametry `klasa`, `group`, `subject`, `teacher`, `hForWeek`.

2. **classes (klasa reprezentująca sale lekcyjne):**

- Atrybuty:
 - `className` (string): nazwa sali lekcyjnej.
 - `classNum` (float): numer klasy.
 - `classSize` (int): pojemność sali lekcyjnej.
 - `supervisor` (string): imię i nazwisko nauczyciela nadzorującego salę.
- Konstruktor inicjalizuje obiekt klasy `classes` przyjmując parametry `className`, `classNum`, `classSize`, `supervisor`.

3. **teachers (klasa reprezentująca nauczycieli):**

- Atrybuty:
 - `name` (string): imię i nazwisko nauczyciela.
 - `tachedSubjects` (vector<string>): lista przedmiotów, które nauczyciel naucza.
 - `favClass` (float): preferowana klasa przez nauczyciela.
- Konstruktor inicjalizuje obiekt klasy `teachers` przyjmując parametry `name`, `tachedSubjects`, `favClass`.

Funkcje:

1. **takeDataFromTXT:**

- Odczytuje dane z plików tekstowych `classes.txt`, `classrooms.txt`, `teachers.txt`.
- Tworzy obiekty klas `kl`, `classes`, i `teachers` na podstawie wczytanych danych.

- Zapisuje te obiekty do przekazanych wektorów `planLekcji`, `klasy`, i `nauczyciele`.
2. **displaySubjects:**
 - Wyświetla dane z pliku `classes.txt`, tj. informacje o klasach, grupach, nauczycielach, przedmiotach i liczbie godzin.
 3. **displayClassrooms:**
 - Wyświetla dane z pliku `classrooms.txt`, tj. informacje o salach lekcyjnych, ich numerach, pojemności i nauczycielach nadzorujących.
 4. **displayTeachers:**
 - Wyświetla dane z pliku `teachers.txt`, tj. informacje o nauczycielach, przedmiotach, jakie nauczają, i preferowanej klasie.
 5. **insertionSortByHours:**
 - Sortuje wektor obiektów klasy `kl` według liczby godzin lekcyjnych przy użyciu algorytmu sortowania przez wstawianie.
 6. **bubbleSortByClassNum:**
 - Sortuje wektor obiektów klasy `classes` według numerów klas przy użyciu algorytmu sortowania bąbelkowego.
 7. **partitionTeachers:**
 - Funkcja pomocnicza dla algorytmu Quick Sort, dzieli wektor nauczycieli na dwie części.
 8. **quickSortByFavClass:**
 - Sortuje wektor nauczycieli według preferowanej klasy przy użyciu algorytmu Quick Sort.

Działanie Programu:

1. Wczytuje dane z plików tekstowych, inicjalizuje obiekty klas.
2. Wyświetla informacje o klasach, salach lekcyjnych i nauczycielach.

3. Sortuje dane w wektorach według określonych kryteriów.
4. Ponownie wyświetla posortowane dane.

Uwagi:

- Program jest napisany w języku C++ i wykorzystuje obiektową strukturę danych do przechowywania informacji o klasach, salach lekcyjnych i nauczycielach.
- Zastosowano różne algorytmy sortowania dla różnych kryteriów sortowania.
- Wprowadzono odpowiednie zabezpieczenia, takie jak obsługa polskich znaków w plikach tekstowych.