

II. Manipulacja obiektami typu string, generowanie danych losowych, użycie typu `string_view`.

Warunkiem wstępnym do realizacji tematów obecnego zajęcia laboratoryjnego jest kod z poprzedniego etapu (poprzedniego laboratorium). Ogólnym założeniem realizacji laboratoriów jest rozbudowa kodu albo jego modyfikacja nie powodująca redukcji jego funkcjonalności.

Zademonstrować w programie z etapu I, który ma na przykład postać

```
struct Dane_osobowe {
    int wiek;
    string godnosc;
};
struct Student {
    int id;
    Dane_osobowe osoba;
    int liczba_predmiotow_wybranych;
    Przedmiot ** predmioty_wybrane;
    int liczba_ocen;
    int** zbior_ocen_z_predmitow; // macierz (np. wierz symbolizuje przedmiot, natomiast kolumny są ocenami)
};
struct Przedmiot {
    string nazwa;
    int liczba_studentow_do_predmiotu;
    Student ** studenci_przypisane_do_predmiotu;
};
struct Wykladowca {
    Dane_osobowe osoba;
    int liczba_predmiotow;
    Przedmiot *predmioty;
};
struct Studia {
    int liczba_wykladowcow;
    int liczba_studentow;
    Wykladowca** wyklawowcy;
    Student** studenci;
};
```

użycie typów `std::random_device`, `std::default_random_engine`, `std::uniform_int_distribution<...>` do generowania liczb i danych losowych w zawartych w atrybutach struktur.

1. Generowanie danych losowych

1.1. Do wygenerowania danych tekstowych zdefiniuj stałe tablice danych typu `std::string_view` nawiązujących do danych programu, np. takich jak nazwiska i imiona, nazwy miesięcy, nazwy chorób, objawy chorób, nazwy przedmiotów, nazwy dziedzin naukowych, itd. Dla powyższych struktur wystarczy zdefiniować dwie tablice z dowolnymi imionami oraz nazwiskami, i tablicę z nazwami przedmiotów. Wówczas **godność** jest wynikiem konkatenacji losowo wybranego nazwiska oraz imienia. Dane liczbowe generować uwzględniając przedziały ich dopuszczalnych wartości (w powyższych strukturach dane liczbowe to są wartości zmiennych **id** student(a)/(ki), **wiek**, **ocena** z przedmiotów, **liczba_studentów** albo **liczba_wykladowców**).

1.2. Przypisywać generowane dane do atrybutów obiektów struktur.

1.3. W programie uwzględnić sprawdzanie i ewentualną zmianę małych liter na duże.

1.4. Uwzględnić w programie opcję menu (case'a w switch-case) w którym jest wywoływana funkcja realizująca punkty zadania 1.1, 1.2 i 1.3.

2. Wprowadzenie i parsowanie zmiennych tekstowych

Celem punktu zadania jest dynamiczne dodawanie obiektu wybranego typu do tablicy na podstawie danych tekstowych wprowadzanych przez użytkownika z klawiatury za pomocą funkcji `getline()`.

2.1. Określ dla wybranego typu danych kształt tekstu (regułę) do wprowadzenia przez użytkownika z klawiatury.

Przykład dla struktury `Student` :

Id : 123; Dane osobowe: Marek Kowal; Przedmioty: Fizyka, Programowanie; Ocenę z przedmiotów : 4, 5; 4.5, 5;

2.2. Stwórz obiekt typu *Student*, parsując dane zmiennej tekstowej (jak w punkcie 2.1. zadania) i przypisując je do atrybutów obiektu *Student*.

2.3. Wstaw obiekt do odpowiedniej tablicy (np. do tablicy *studenci* w strukturze *Studia*).

3. Zadanie domowe

Uwzględnić weryfikację poprawności wprowadzanych danych przez użytkownika.

4. Pliki źródłowe programu wgrać do zadania **Temat 2** w Teams w określonym terminie.

5. Podczas zajęcia zademonstrować działanie programu.