Wrocław, 6 grudnia 2021

Podstawy Programowania

Laboratorium 5

Operacje na plikach

prowadzący: *dr inż. Radosław Idzikowski*

1 Wprowadzenie

Celem zajęć jest poznanie podstaw programowania w języku `C/C++`, w szczególności oprogramowanie prostej bazy danych wykorzystującej reprezentacje w postaci tablicy struktur lub tablicy wskaźników na struktury. Rozbudowanie programu o operacje archiwizacji danych w pamięci zewnętrznej w postaci plików tekstowych lub binarnych.

Zapis lub odczyt danych może zrealizować przy użyciu biblioteki `fstream`. Po załączeniu wspomnianej biblioteki pierwszym krokiem jest utworzenie uchwytu do pliku poprzez utworzenie obiektu klasy `fstream`. Domyśle utworzona zmienna nie wskazuję na żaden plik. Poprzez użycie metody `open(const char * path, mode)` można otworzyć plik do odczytu lub zapisu. Jako ścieżkę (*path*) można podać ścieżkę względną lub bezwzględną do pliku. Następnie należy wskazać w jakim trybie chcemy otworzyć plik.

`std::fstream::in`	zezwolenie na odczytywanie danych z pliku.
`std::fstream::out`	zezwolenie na zapisywanie danych do pliku.
`std::fstream::app`	zezwolenie na zapisywanie danych do pliku, ale dane mogą być zapisywane tylko i wyłącznie na końcu pliku.

```cpp
#include <iostream>
#include <fstream>

int main()
{
  std::fstream file;
  file.open("data.txt", std::fstream::in);
  if (file.is_open())
  {
    int x;
    file >> x;
    std::cout << x << " ";
    while (!file.eof()) {
      file >> x;
      std::cout << x << " ";
    }
    std::cout << "\n";
    file.close();
  }
}
```

Dane z pliku można odczytać przy użyciu strumienia `plik>>zmienna;` oraz analogicznie zapisać `plik<<zmienna;`. Wewnątrz klasy `fstream` jest wiele przydatnych metod takich jak: `eof()` do sprawdzenia czy wskaźnik pliku znajduje się na jego końcu oraz `is_open()` do sprawdzenia czy udało się poprawnie otworzyć plik. Przy użyciu strumienia z pliku odczytujemy kolejne dane oddzielone białymi znakami:

```
6 3 2
1 7
```

bez znaczenia czy jesto spacja, tabulacja czy nowa linia.

Struktury definiujemy przy użyciu słowa kluczowego `struct`. Struktury służą do przechowywania złożonych danych, mogą posiadać składowe różnego typu. Na końcu definicji struktury zawsze należy umieścić średnik.

```
#include <iostream>

struct point
{
  double x, y;
};

void print(point P) {
  std::cout << "(" << P.x << "," << P.y << ")\n";
}

int main()
{
  point A = {1.5,-1.5};
  point B;
  B.x = -0.5;
  B.y = 2.5;
  print(A);
  print(B);
}
```

2 Zadanie

1. Napisz prosty program bazodanowy. Należy zapewnić następujące funkcje:

   - wczytanie z pliku,
   - zapis do pliku,
   - wyświetlenie bazy,
   - usuwanie,
   - dodawanie,
   - losowanie,
   - wyszukiwanie,
   - filtrowanie,
   - sortowanie.

   Przykładowe struktury danych:

   - student
     - nr indeksu,
     - imię,
     - nazwisko,
     - płeć,
     - data urodzenia,
   - samochód
     - nr rejestracyjny,
     - marka,
     - model,
     - typ,
     - rocznik,
     - moc.