

Programowanie 2

Zadanie 4

Piotr Błaszyński

23 marca 2019

Do klas przechowujących drużyny i zawodników dodać funkcjonalność odczytu i zapisu danych z i do plików tekstowych. Program na starcie powinien zapytać o nazwę pliku (pliku o takiej nazwie szukamy w **bieżącym** katalogu (w przypadku projektów w VS może to być podkatalog bin/Debug w katalogu projektu)).

BARDZO WAŻNE:

Klasa opisująca drużynę odczytuje z pliku o wskazanej nazwie pojedyncze linie, przetwarzaniem linii na dane zawodnika zajmuje się klasa zawodnik (na wejściu np. jednego z konstruktorów może otrzymać linię w postaci łańcucha znaków).

Dane jednego zawodnika są przechowywane jako jedna linia w pliku, elementy są od siebie oddzielone średnikiem. Przykładowy kod demonstracyjny:

- zachować (z poprzednich zajęć) kod tworzący 3 drużyny po 5 zawodników, 2 drużyny unikalne, trzecia będąca kopią drugiej, dołożyć kolejną (czwartą) - wczytaną z pliku,
- zmodyfikować dane jednego zawodnika z wczytanej drużyny,
- wyświetlić zawartość wszystkich drużyn (w tym wczytanej z pliku)
- zapisać na koniec wszystkie drużyny do plików o nazwach: "druzyna" + nazwa_druzyny + czas z dokładnością do sekundy (w formacie h:mm:ss 23:59:59 - to oznacza sekundę przed północą, zera wiodące nie są wymagane). Do budowania nazwy pliku można skorzystać z klasy `stringstream` (plik nagłówkowy `sstream`). Przykładowy kod pokazujący czas na ostatniej stronie.

Przykładowe użycie klas obsługujących pliki:

```
#include <ctime>
#include <iostream>
#include <istream>
#include <fstream>
#include <string>

using namespace std;

void file_read ()
{
    ifstream inputFile("foobar.txt");
    if (!inputFile.is_open())
        throw std::exception("Can't read foobar.txt");
    bool we_use_string = true;
    if (we_use_string)
    {
        //blok ponizej do wykorzystania gdy korzystamy z klasy
        string - zalecane
        string lineAsString;
        while (!inputFile.eof())
        {
            inputFile >> lineAsString;
            cout << lineAsString << endl;
        }
    }
    else//ta wersja tylko wtedy, gdy bardzo potrzebne jest
        char *
    {
        char *lineAsCharPointer=new char[200];
        while (inputFile.getline(lineAsCharPointer, 100))
            cout << lineAsCharPointer << endl;
    }
}

void file_write ()
{
    ofstream outputFile("fooboo.txt");
    if (!outputFile.is_open())
        throw std::exception("Can't write fooboo.txt");
    outputFile <<"Line_1" <<endl;
    //char *lineAsCharPointer="LineAsChar*";
    //outputFile <<lineAsCharPointer <<endl;
    string lineAsString="LineAsString";
    outputFile << lineAsString <<endl;
}
```

Przykładowy kod pokazujący czas:

```
#include <ctime>
#include <iostream>
using namespace std;
void print_time()
{
    time_t t = time(0);
    struct tm now;
    localtime_s(&now, &t); //wersja VS, gdzie indziej
    prawdopodobnie localtime
    cout << (now.tm_year + 1900) << '- '
        << (now.tm_mon + 1) << '- ' << now.tm_mday << endl
        << now.tm_hour << ':' << now.tm_min << ':'
        << now.tm_sec << endl;
}
```