

Odczyt daty

Po emocjach związanych ze sprawdzianem kolejne zadanie będzie stosunkowo proste. Powinno (przynajmniej w założeniach) pokazać „masowe” testowanie funkcji oraz testowanie operacji wykonywanych na strumieniach.

Trzeba jednak pamiętać, że im mniejsza liczba testów, które pozwalają na wykrycie wszystkich błędów, tym lepiej. W zadaniu będziemy posługiwać się klasą `io_date` - wyjątkowo uproszczoną, bo testy, które przygotujemy będą wymagać jedynie domyślnego konstruktora oraz operatorów wprowadzania ze strumienia i wyprowadzania na strumień.

Trzeba jednak uwzględnić sprawdzanie poprawności daty.

Zarys testów

Podstawowy przebieg testów – bez uwzględnienie wyjątków, które tutaj mają prawo się pojawić – jest stosunkowo prosty:

```
string input_date = "2020-02-29";
stringstream is(input_date);

io_date dt;
is >> dt;

stringstream os;
os << dt;

if (os.str() != input_date)
    cout << "Error - output " << os.str()
        << " != " << input_date << '\n';
```

Proszę zwrócić uwagę, na wykorzystanie strumieni w pamięci (**stringstream**). Dzięki temu unikamy wpisywania danych i wpatrywania się w konsolę.

Minimalistyczna definicja klasy

```
class io_date
{
    int day, month, year;

    // void set_members(int day, int month, int year);
    // bool is_valid(int day, int month, int year);
public:
    io_date() : day(1), month(1), year(2020)
    { }

    friend std::ostream& operator<<(std::ostream& os,
                                    const io_date& dt);
    friend std::istream& operator>>(std::istream& is,
                                    io_date& dt);
};
```

Przyjmijmy, że do sygnalizacji niepoprawnego wejścia używamy klasy `runtime_error`.

Dostarczanie rozwiązania

Rozwiązaniem zadania są pliki **io_date.h** i **io_date.cpp**.

Proszę załadować w Moodle tylko te dwa pliki.

Na początku każdego pliku trzeba umieścić w komentarzu imię, nazwisko i nr albumu autora.

Rozwiązanie należy przekazać do:

7 grudnia 2022 23:59