

Міністерство освіти і науки України  
Національний технічний університет України «Київський політехнічний  
інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 4  
з дисципліни «Основи програмування –  
2. Метидології програмування»

«Перевантаження операторів»

Варіант 23

Виконав студент ІП-13 Недельчев Євген Олександрович  
(шифр, прізвище, ім'я, по батькові)

Перевірів Вєчерковська Анастасія Сергіївна  
(прізвище, ім'я, по батькові)

# Лабораторна робота 1

## Варіант 23

23. Визначити клас "Дата" для роботи із датами в межах року. Членами класу є число, місяць та рік. Реалізувати для нього декілька конструкторів, геттери, метод визначення терміну, що залишився до кінця року. Перевантажити оператори: префіксний "++" - для збільшення кількості місяців на 1, постфіксний "++" - для збільшення кількості днів на 1; ">" - для порівняння дат. Створити три об'єкти (D1, D2, D3), використовуючи різні конструктори. Збільшити дату D1 на 1 місяць, а дату D2 - на 1 день. З'ясувати, яка із цих дат (D1 чи D2) більш пізня. Для дати D3 визначити термін, що залишився до кінця року.

## Код програми

C++

main.cpp

```
#include "Date.h"

int main() {
    setlocale(LC_ALL, "rus");

    cout << "Enter the first date:" << endl;
    int day, month, year;
    cout << "day>> "; cin >> day;
    cout << "month>> "; cin >> month;
    cout << "year>> "; cin >> year;

    Date D1(day, month, year);

    cout << "Enter the second date:" << endl;
    cin.ignore();
    string line; getline(cin, line);
    Date D2(line);

    Date D3(D1);

    printf("D1 %d %d %d\n", D1.GetDay(), D1.GetMonth(), D1.GetYear());
    printf("D2 %d %d %d\n", D2.GetDay(), D2.GetMonth(), D2.GetYear());
    printf("D3 %d %d %d\n", D3.GetDay(), D3.GetMonth(), D3.GetYear());

    ++D1;
    printf("\n+1 month to D1:\n\t %d %d %d\n", D1.GetDay(), D1.GetMonth(),
D1.GetYear());
    D2++;
    printf("\n+1 day to D2:\n\t %d %d %d\n", D2.GetDay(), D2.GetMonth(), D2.GetYear());
    if (D1 > D2) {
        printf("\nДата D1 більш пізня\n");
    }
    else {
        printf("\nДата D2 більш пізня\n");
    }
}
```

```

        printf("\nТермін, що залишився до кінця року для D3: %d днів",
D3.DaysBeforeNextYear());
    }

```

## Date.h

```

#pragma once
#include <iostream>
#include <string>
#include <vector>

using namespace std;

class Date {
private:
    int day;
    int month;
    int year;
public:
    Date(int day = 1, int month = 1, int year = 1970);
    Date(string line);
    Date(Date& a);
    int GetYear();
    int GetMonth();
    int GetDay();
    int DaysBeforeNextYear();
    Date& operator++ ();
    Date& operator++ (int);
    friend bool operator> (Date& a, Date& b);
};

vector<string> split(string line);

```

## Date.cpp

```

#include "Date.h"

Date::Date(int day, int month, int year) {
    this->year = year;
    this->month = month;
    this->day = day;
}

Date::Date(string line) {
    vector<string> tmp = split(line);
    this->year = stoi(tmp[2]);
    this->month = stoi(tmp[1]);
    this->day = stoi(tmp[0]);
}

Date::Date(Date& a) {
    this->day = a.day;
    this->month = a.month;
    this->year = a.year;
}

int Date::GetDay() { return this->day; }
int Date::GetMonth() { return this->month; }
int Date::GetYear() { return this->year; }

int Date::DaysBeforeNextYear() {

```

```

    int days[12] = { 365, 334, 304, 273, 243, 212, 181, 151, 120, 90, 59, 31};
    return days[this->month - 1] - this->day;
}

Date& Date::operator++() {
    month++;
    if (month > 12) {
        year++;
        month = 1;
    }
    return *this;
}

Date& Date::operator++(int) {
    int days[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
    day++;
    if (day > days[month - 1]) {
        month++;
        day = 1;
    }
    if (month > 12) {
        year++;
        month = 1;
    }
    return *this;
}

bool operator>(Date& a, Date& b) {
    return (a.year * 365 + a.month * 30 + a.day) > (b.year * 365 + b.month * 30 + b.day);
}

vector<string> split(string line) {
    vector<string> words;
    char sep[4] = { ' ', ':', '-', '.' };
    string temp_word = "";
    line += ' ';
    for (int i = 0; i < line.length(); i++) {
        char* symbol = find(begin(sep), end(sep), line[i]);
        if (symbol != end(sep)) {
            if (temp_word.length() > 0) {
                words.push_back(temp_word);
            }
            temp_word = "";
        }
        else {
            temp_word += line[i];
        }
    }
    return words;
}

```

## Тестування:

```
Консоль отладки Microsoft Visual Studio

Enter the first date:
day>> 12
month>> 12
year>> 2003
Enter the second date:
31:12:1999
D1 12 12 2003
D2 31 12 1999
D3 12 12 2003

+1 month to D1:
      12 1 2004

+1 day to D2:
      1 1 2000

Дата D1 більш пізня

Термін, що залишився до кінця року для D3: 19 днів
```

**Висновки:**

Я вивчив та використав на практиці механізми створення класів з використання перевантаження операторів(операцій).