# 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 № 9 з дисципліни

«Основи програмування

»

Варіант 23

Виконав студент Панченко Сергій Віталійович

(шифр, прізвище, ім'я, по батькові)

Перевірив

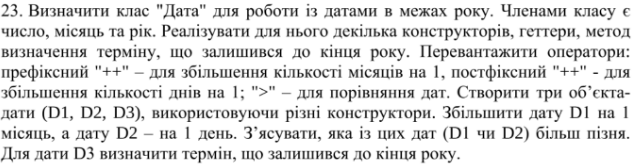
( прізвище, ім'я, по батькові)

Київ 202

## Лабораторна робота 4

**Мета:** вивчити механізми створення класів з використанням перевантажених операторів

**Постановка задачі:**

****

C:

#include <iostream>

using namespace std;

class Data {

private:

class Month {

public:

int number;

int days;

string name;

Month() {}

Month(string name, int number, int days) {

this->number = number;

this->days = days;

}

};

Month months[13] = {

Month("January", 1, 31),

Month("February", 2, 2829),

Month("March", 3, 31),

Month("April", 4, 30),

Month("May", 5, 31),

Month("June", 6, 30),

Month("July", 7, 31),

Month("August", 8, 31),

Month("September", 9, 30),

Month("October", 10, 31),

Month("November", 11, 30),

Month("December", 12, 31),

};

int day;

Month month;

int year;

void Update() {

if (this->day > this->month.days) {

this->day = 1;

this->month.number++;

}

if (this->month.number > 12) {

this->month = months[0];

this->year++;

}

this->month = months[this->month.number-1];

if (this->month.number == 2 && IsFourth(year)) {

this->month.days %= 100;

}

if (this->month.number == 2 && !IsFourth(year)) {

this->month.days /= 100;

}

}

public:

Data() {

this->day = 6;

this->month = months[0];

this->year = 1;

}

Data(int year) {

this->day = 1;

this->month = months[0];

this->year = year;

}

Data(int day, int month, int year) {

this->day = day;

this->month = months[month-1];

if (month == 2 && IsFourth(year)) {

this->month.days %= 100;

}

if (month == 2 && !IsFourth(year)) {

this->month.days /= 100;

}

this->year = year;

}

int GetDay() {

return this->day;

}

int GetMonth() {

return this->month.number;

}

int GetYear() {

return this->year;

}

static bool IsFourth(int year) {

bool res = 0;

if (year % 4 == 0 || (year % 100 == 0 && year % 400 == 0)) {

res = 1;

}

return res;

}

//// Prefix increment operator

Data& operator++() {

this->month.number++;

this->Update();

return \*this;

}

//Postfix increment operator

Data operator++(int) {

this->day++;

this->Update();

return \*this;

}

bool operator >(const Data& d) {

bool res = 0;

if (this->year > d.year) {

res = 1;

}

else if (this->year == d.year) {

if (this->month.number > d.month.number) {

res = 1;

}

else if (this->month.number == d.month.number) {

if (this->day > d.day) {

res = 1;

}

}

}

return res;

}

int RestDays() {

int total\_days = (IsFourth(this->year)) ? 366 : 365;

int days\_passed = 0;

for (int i = 0; i < this->month.number; i++){

if (i == 1) {

int feb\_days = (IsFourth(this->year)) ? 29 : 28;

days\_passed += feb\_days;

}

else {

days\_passed += months[i].days;

}

}

return total\_days - days\_passed;

}

};

int main()

{

int year1 = 0;

int month1 = 0;

int day1 = 0;

int year2 = 0;

int month2 = 0;

int day2 = 0;

int year3 = 0;

int month3 = 0;

int day3 = 0;

cout << "Enter year1: ";

cin >> year1;

cin.ignore();

cout << "Enter month1: ";

cin >> month1;

cin.ignore();

cout << "Enter day1: ";

cin >> day1;

cin.ignore();

cout << "Enter year2: ";

cin >> year2;

cin.ignore();

cout << "Enter month2: ";

cin >> month2;

cin.ignore();

cout << "Enter day2: ";

cin >> day2;

cin.ignore();

cout << "Enter year3: ";

cin >> year3;

cin.ignore();

cout << "Enter month3: ";

cin >> month3;

cin.ignore();

cout << "Enter day3: ";

cin >> day3;

cin.ignore();

Data d1 = Data(day1, month1, year1);

Data d2 = Data(day2, month2, year2);

Data d3 = Data(day3, month3, year3);

cout << "D1:" << endl;

cout << "\tYear: " << d1.GetYear()

<< "\tMonth: " << d1.GetMonth()

<< "\tDay: " << d1.GetDay() << endl;

++d1;

cout << "D1:" << endl;

cout << "\tYear: " << d1.GetYear()

<< "\tMonth: " << d1.GetMonth()

<< "\tDay: " << d1.GetDay() << endl;

cout << "D2:" << endl;

cout << "\tYear: " << d2.GetYear()

<< "\tMonth: " << d2.GetMonth()

<< "\tDay: " << d2.GetDay() << endl;

d2++;

cout << "D2:" << endl;

cout << "\tYear: " << d2.GetYear()

<< "\tMonth: " << d2.GetMonth()

<< "\tDay: " << d2.GetDay() << endl;

cout << "D1>D2?: " << (d1 > d2) << endl;

cout << "D3 Rest: " << d3.RestDays() << endl;

system("pause");

return 0;

}

**Результат:**

****

**Висновок:** під час лабораторної роботи навчилися працювати з класами та перевантажувати оператори. Оскільки програма створюює об’єкти класу та перезавантажує оператори, виконуючи умови поставленої до варіанту задачі, то лабораторна виконана правильно.