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 No 4 з дисципліни
«Основи програмування 2.
Модульне програмування»

«Перевантаження операторів»

Варіант 7

Виконав студент ІП-1407 Грицина Діана Русланівна (шифр, прізвище, ім'я, по батькові)

Перевірив _____ (прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота No 4

Тема: Перевантаження операторів.

Мета: вивчити механізми створення класів з використанням перевантажених операторів(операцій).

Задача:

7. 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 пори року, що відповідає вказаній даті. Перевантажити оператори: "+=" – для збільшення дати на вказану кількість днів, "-" – для знаходження інтервалу між двома датами. Створити три об'єкта-дати (D1, D2, D3), використовуючи різні конструктори. Збільшити дату D1 на 9 днів, а дату D2 – на 14 днів. Визначити тривалість інтервалу між датами D1 і D2. Для дати D3 визначити пору року, якій відповідає ця дата.

Постановка задачі: Винесемо інтерфейс класу у окремий модуль – заголовний файл, в оголошенні класу запишемо атрибути та прототипи функцій. Визначення методів розмістимо в файлі реалізації. Використаємо операторні функції для перевантаження операторів "+=" та "-".

main.cpp

```
#include <iostream>
#include "DateOL.hpp"
using namespace std;
int main() {
    DateOL Date("12", "12", "2020");
    DateOL Date2;
    DateOL Date3("08", "07", "2015");
    Date.print();
    Date2.print();
    Date3.print();
    Date += 9;//Збільшення дати 1 на 9 днів
    cout<<"date1 after changing\n";
    Date.print();
    Date2 += 14;//Збільшення дати 2 на 14 днів
    cout<<"date2 after changing\n";
    Date2.print();
    int res = Date - Date2;//Тривалість інтервалу між датами
    cout<<"Result: "<<res<<"\n";
    string season = Date3.getSeason();//пора року, якій відповідає дата
    cout<<"date3 Season: "<<season<<"\n";
    return 0;
}
```

DateOL.hpp

```hpp
#ifndef DateOL_hpp
#define DateOL_hpp
#include <iostream>
#include <stdio.h>
using namespace std;
class DateOL{
    string day, month, year;
public:
    DateOL(string day, string month, string year);
    DateOL();
    void setDay(string day);
    void setMonth(string month);
    void setYear(string year);
    string getDay();
    string getMonth();
    string getYear();
    void print();
    int daysInMonth(int month, int year);
    int daysInYear(int year);
    string getSeason();
    DateOL& operator+=(int days);
    friend int operator-(DateOL date1, DateOL date2);

};
#endif /* DateOL_hpp */
```

DateOL.cpp

```cpp
#include "DateOL.hpp"
#include <iostream>
using namespace std;
DateOL::DateOL(string day, string month, string year){
    this->day = day;
    this->month = month;
    this->year = year;
}
DateOL::DateOL(){
    cout<<"Enter day ";
    cin>>day;
    cout<<"Enter month ";
    cin>>month;
    cout<<"Enter year ";
    cin>>year;
}
void DateOL::setDay(string day){
    this->day = day;
}
void DateOL::setMonth(string month){
    this->month = month;
}
void DateOL::setYear(string year){
    this->year = year;
}
string DateOL::getDay(){
    return day;
}
string DateOL::getMonth(){
    return month;
}
string DateOL::getYear(){
    return year;
```

```
void DateOL::print(){
    cout<<"Date: "<<day<<"."<<month<<"."<<year<<"\n";
}
int DateOL::daysInMonth(int month, int year){
    int month30[] = {4, 6, 9, 11};//місяці які мають 30 днів
    for(int i=0; i<4; i++){
        if(month==month30[i]) return 30;//перевірка чи місяць має 30 днів
    }
    if(month==2){
        if(year%4==0)return 29;//перевірка на високосний рік
        else return 28;
    }else return 31;//всі інші місяці мають 31 день
}
int DateOL::daysInYear(int year){
    if(year%4==0) return 366;//перевірка на високосний рік
    else return 365;
}
DateOL& DateOL::operator+=(int days){
    int dayInt = stoi(this->day);
    int monthInt = stoi(this->month);
    int yearInt = stoi(this->year);
    if(dayInt+days>this->daysInMonth(monthInt, yearInt)){//додаємо більше днів чим є в місяці
        do{
            days -= this->daysInMonth(monthInt, yearInt)-dayInt;//віднімаємо місяць від числа
                днів
            dayInt = 0;
            monthInt++;//додаємо місяць який відняли
            if(monthInt==13){//якщо місяців більше чим в році
                monthInt = 1;//перший місяць в році
                yearInt++;//додаємо рік
            }
        }while(dayInt+days>this->daysInMonth(monthInt, yearInt));//перевірка чм більше днів
            чим є в місяці
    }
    dayInt += days;//додаємо залишок
    string day2 = to_string(dayInt);
    string month2 = to_string(monthInt);
    string year2 = to_string(yearInt);
    if(dayInt<10) day2 = "0"+to_string(dayInt);
    if(monthInt<10) month2 = "0"+to_string(monthInt);
    this->setDay(day2);//встановлення нової дати
    this->setMonth(month2);
    this->setYear(year2);
    return (*this);
}
int operator-(DateOL date1, DateOL date2){
    int res = 0;
    int dayInt1 = stoi(date1.day);
    int monthInt1 = stoi(date1.month);
    int yearInt1 = stoi(date1.year);
    int dayInt2 = stoi(date2.day);
    int monthInt2 = stoi(date2.month);
    int yearInt2 = stoi(date2.year);
    if(yearInt2>yearInt1+1){//якщо різниця між датами більше за 1 рік
        int yearChecker = yearInt2 - 1;
        do{
            res += date2.daysInYear(yearChecker);//підрахунок днів в цілих роках
            yearChecker--;
        }while(yearChecker!=yearInt1);
        int monthCheker1 = monthInt1;
        do{
            res += date1.daysInMonth(monthCheker1, yearInt1);//підрахунок днів в місяцях
                в першому році
            monthCheker1++;
        }while(monthCheker1<13);
        res -= dayInt1;
        int monthCheker2 = 1;
        while(monthCheker2<monthInt2){
            res += date2.daysInMonth(monthCheker2, yearInt2);//підрахунок днів в місяцях
                в останньому році
            monthCheker2++;
        };
```

```
        res += dayInt2;
    }
    if(yearInt2==yearInt1){//підрахунок в 1 році
        if(monthInt1!=monthInt2){
        int monthChecker = monthInt1;
        do{
            res += date1.daysInMonth(monthChecker, yearInt1);//підрахунок днів в місяцях
            monthChecker++;
        }while(monthChecker<monthInt2);
        res -= dayInt1;
        res += dayInt2;

        }else res = dayInt2 - dayInt1;//різниця між днями, якщо в одному місяці
    }
    if(yearInt1+1==yearInt2){//підрахунок між сусідніми роками
        int monthCheker1 = monthInt1;
        do{
            res += date1.daysInMonth(monthCheker1, yearInt1);//підрахунок днів в місяцях в
                першому році
            monthCheker1++;
        }while(monthCheker1<13);
        res -= dayInt1;
        int monthCheker2 = 1;
        while(monthCheker2<monthInt2){
            res += date2.daysInMonth(monthCheker2, yearInt2);//підрахунок днів в місяцях в
                останньому році
            monthCheker2++;
        };
        res += dayInt2;
    }
    return res;

}
string DateOL::getSeason(){
    int monthInt = stoi(month);
    //визначення пори року за місяцем
    if(monthInt<=2||monthInt==12)return "winter";
    else if(monthInt<=5) return "spring";
    else if(monthInt <=8) return "summer";
    else return "autumn";
}
```

Результат

```
Enter day 07
Enter month 08
Enter year 2021
Date: 12.12.2020
Date: 07.08.2021
Date: 08.07.2015
date1 after changing
Date: 21.12.2020
date2 after changing
Date: 21.08.2021
Result: 243
date3 Season: summer
Program ended with exit code: 0
```

Висновок: у лабораторній роботі досліджено механізми створення класів з використанням перевантажених операторів(операцій). Інтерфейс класу DateOL винесено у окремий модуль – заголовний файл DateOL.hpp, в оголошенні класу записано атрибути та прототипи функцій. Визначення методів розміщено в файлі реалізації DateOL.cpp. Перевантажено унарний оператор “+=” з використанням операторної функції-члена, де аргумент передається неявно через покажчик this. Для перевизначення бінарного оператора “-” використано дружню функцію з явною передачею двох операндів.