МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cт.преподаватель |  |  |  | Е.О.Шумова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 3 |
| «Перегрузка операторов» |
| по курсу: Объектно-ориентированное программирование |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134К |  |  |  | Опарин С.Н. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

1. **Задание**

**Вариант №9:**

Разработать класс «Дата». Определить в нем конструкторы и деструктор, перегрузить операцию добавления к дате заданного количества дней, вычитания из даты заданного количества дней, операцию вычитания двух дат, операции сравнения на совпадение(несовпадение).

1. **Листинг программы, реализующей алгоритм:**

#include <iostream>

#include <iomanip>

#include <cstdio>

using namespace std;

class Date{

private:

unsigned int day, month, year;

int max\_day(unsigned int m, unsigned int y);

public:

int flag;

Date(unsigned int d = 1, unsigned int m = 1, unsigned int = 1970);

Date(char \*str);

Date(const Date & copy) : day(copy.day), month(copy.month), year(copy.year){}

~Date(){}

void check\_date(unsigned int d, unsigned int m, unsigned int y, int flag);

bool leap(int y);

// int max\_day(unsigned int m, unsigned int y);

Date operator + (unsigned int days\_to\_add);

Date operator - (int days\_to\_subtract);

friend Date operator - (Date &d1, Date &d2);

friend bool operator == (Date &d1, Date &d2);

void show\_date();

};

bool Date::leap(int y){

return ((y % 4 == 0 && y % 100 != 0) || y % 400 == 0);

}

int Date::max\_day(unsigned int m, unsigned int y){

int days[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

if (leap(y)){

days[1] = 29;

}

if (m == 0){

m = 12;

}

return days[m - 1];

}

void Date::check\_date(unsigned int d, unsigned int m, unsigned int y, int flag) {

year = y;

while (m > 12){

cout << "Введите месяц <= 12! \n"<< "Месяц: ";

cin >> m;

}

month = m;

while ((d > max\_day(m, y))){

if(flag == 0){

printf("Введите день <= %d!\nДень: ", max\_day(m,y));

cin >> d;

}

else{

break;

}

}

flag = 1;

day = d;

}

Date::Date(unsigned int d, unsigned int m, unsigned int y) {

check\_date(d, m, y, flag);

}

Date::Date(char \*str){

char \*tPtr;

tPtr = strtok(str," .");

day = atoi(tPtr);

tPtr = strtok(NULL," .");

month = atoi(tPtr);

tPtr = strtok(NULL," .");

year = atoi(tPtr);

check\_date(day, month, year, flag);

}

Date Date :: operator+ (unsigned int days\_to\_add){

int d, m = month, y = year;

if(day + days\_to\_add > max\_day(month, year)){

while(days\_to\_add > 31){

days\_to\_add -= max\_day(m, y);

m += 1;

if(m > 12){

m = 1;

y = year + 1;

}

}

d = day + days\_to\_add;

if(d > max\_day(m, y)){

d = day + days\_to\_add - max\_day(m, y);

m += 1;

}

if(m > 12){

m = 1;

y = year + 1;

}

}

else {

d = day + days\_to\_add;

m = month;

y = year;

}

return Date(d, m, y);

}

Date Date :: operator- (int days\_to\_subtract){

int d = 0, m = month, y = year, flag\_ = 0;

if((days\_to\_subtract < max\_day(m, y)) && (flag\_ == 0)){

d = day - days\_to\_subtract;

}

if((day - days\_to\_subtract == 0) && (flag\_ == 0)){

d = max\_day(month - 1, year);

m = month - 1;

y = year;

}

else if((day - days\_to\_subtract > 0) && (flag\_ == 0)) {

d = day - days\_to\_subtract;

m = month;

y = year;

}

return Date(d, m, y);

}

Date operator - (Date &d1, Date &d2){

int d = 0, m = 0, y = 0;

Date final\_date(d, m, y);

if((d1.day < d2.day) && (d1.month < d2.month) && (d1.year < d2.year)){

cout << "Вычитание невозможно, потому что первая дата меньше, чем вторая" << endl;

}

else{

if(d1.year == d2.year){

final\_date.year = 0;

if(d1.month == d2.month){

final\_date.month = 0;

}

else{

final\_date.month = d1.month - d2.month;

}

final\_date.day = d1.day - d2.day;

}

else{

final\_date.year = d1.year - d2.year;

if(d1.month == d2.month){

final\_date.month = d1.month;

}

else if(d1.month > d2.month){

final\_date.month = d1.month - d2.month;

}

else{

final\_date.month = 12 + (d1.month - d2.month);

}

if(d1.day == d2.day){

final\_date.day = d1.day;

}

else if(d1.day > d2.day){

final\_date.day = d1.day - d2.day;

}

else{

final\_date.day = final\_date.max\_day(d1.month - 1, (d1.year - d2.year)) + (d1.day - d2.day);

final\_date.month = d1.month - d2.month - 1;

}

}

}

return final\_date;

}

bool operator ==(Date &d1, Date &d2){

return ((d1.day == d2.day) && (d1.month == d2.month) && (d1.year == d2.year));

}

void Date::show\_date(){

string sMonth[] = { "января", "февраля", "марта", "апреля", "мая",

"июня", "июля", "августа", "сентября", "октября", "ноября", "декабря"};

cout << setw(2) << setfill('0') << day<<'.'<< setw(2) << setfill('0')

<< month<<'.' << setw(4) << setfill('0') << year << endl;

}

int main(){

cout <<"Введите дату: ";

char d1[11];

cin >> d1;

Date date1(d1);

char d2[11];

int task;

cout << "\nВыберите действие, которое вы хотите выполнить: " << endl;

cout << "1 - Прибавить определенное количество дней к дате" << endl;

cout << "2 - Вычесть определенное количество дней из даты" << endl;

cout << "3 - Вычесть из первой даты вторую" << endl;

cout << "4 - Cравнить даты на соответствие" << endl;

cin >> task;

Date result;

bool verdict;

int count\_of\_days;

if (task == 1){

date1.flag = 1;

cout << "Введите количество дней, которые вы хотите добавить: ";

cin >> count\_of\_days;

result = date1 + count\_of\_days;

result.show\_date();

}

else if (task == 2) {

cout << "Введите количество дней, которые вы хотите вычесть: ";

cin >> count\_of\_days;

result = date1 - count\_of\_days;

result.show\_date();

}

else if (task == 3) {

cout <<"Введите вторую дату: ";

cin >> d2;

Date date2(d2);

result = date1 - date2;

result.show\_date();

}

else if (task == 4){

cout <<"Введите вторую дату: ";

cin >> d2;

Date date2(d2);

verdict = date1 == date2;

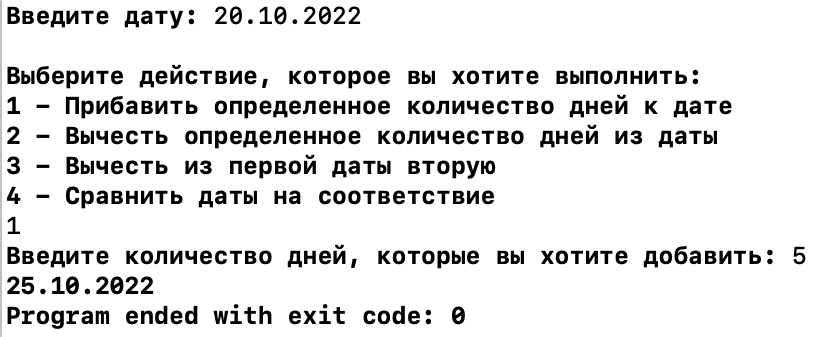
cout << verdict << endl;

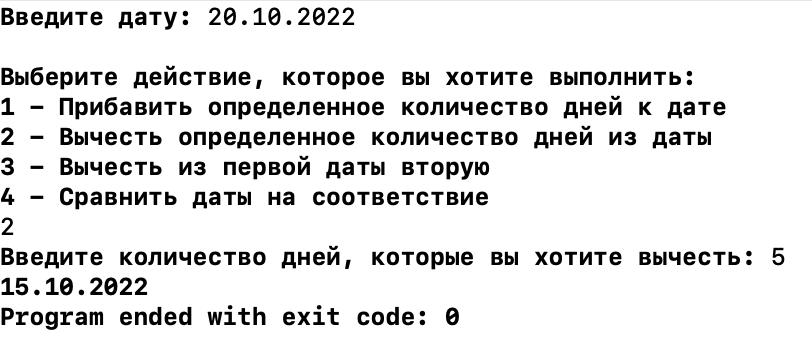
}

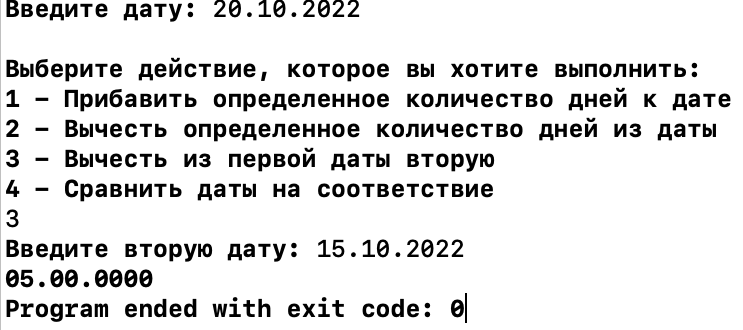
return 0;

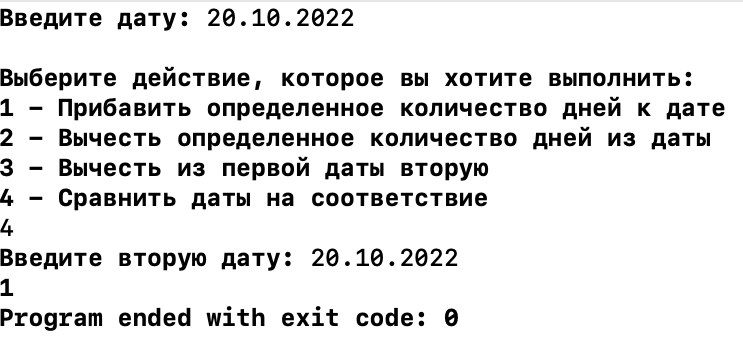
}

Скриншот с результатами:









1. **Вывод**

В ходе лабораторной работы я изучил механизм перегрузки операторов для типов, определенных пользователем посредством использования методов класса и дружественных функций.