Додаток 1

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 № 5
з дисципліни «Основи програмування – 2. Методології програмування»

«Успадкування та поліморфізм»

Варіант 23

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

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

Київ 2022

Лабораторна робота 1

Варіант 23

23. Створити клас TDate, який містить трійку цілих чисел, що представляють число, місяць та рік, і методи для порівняння дат, заданих різними форматами, їх збільшення / зменшення на вказану величину. На основі цього класу створити класи-нащадки TDate1 та TDate2, що представляють дати в форматі "ЧЧ.ММ.РРРР" та "ММ-ЧЧ-РРРР" відповідно. Створити n об'єктів TDate1 та m об'єктів TDate2. Визначити саму пізню дату, а також дати, що належать заданому періоду дат.

Код програми

C++

main.cpp

```cpp
#include "TDate.h"

int main() {
	vector <TDate*> base;
	int n, m;
	string str;
	cout << "Enter n >> "; cin >> n;
	cin.ignore();
	for (int i = 0; i < n; i++) {
		getline(cin, str);
		base.push_back(new TDate1(str));
	}
	cout << "Enter m >> "; cin >> m;
	cin.ignore();
	for (int i = 0; i < m; i++) {
		getline(cin, str);
		base.push_back(new TDate2(str));
	}
	cout << "List of daets in according format:\n";
	TDate* latest = base[0];
	for (int i = 0; i < m + n; i++) {
		base[i]->ShowDate();
		if (base[i]->CompareDates(*latest)) {
			latest = base[i];
		}
	}
	cout << "The latest date: ";
	latest->ShowDate();
	string date1, date2;
	cout << "Enter the lower limit: ";
	getline(cin, date1);
	cout << "Enter the upper limit: ";
	getline(cin, date2);
	cout << "Dates that suit the given range:\n";
	for (int i = 0; i < m + n; i++) {
		if (base[i]->IsInTimeInterval(date1, date2)) {
			base[i]->ShowDate();
		}
	}
}
```

TDate.h

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

using namespace std;

class TDate
{
protected:
	int day;
	int month;
	int year;
public:
	TDate();
	TDate(string);
	int GetDays();
	void IncreaseDate(int day = 0, int month = 0, int year = 0);
	void DecreaseDate(int day = 0, int month = 0, int year = 0);
	bool IsInTimeInterval(string, string);
	bool CompareDates(TDate&);
	virtual void ShowDate();
};

class TDate1 : public TDate {
public:
	TDate1(string);
	void ShowDate() override;
};

class TDate2 : public TDate {
public:
	TDate2(string);
	void ShowDate() override;
};

  bool IsDateValid(cmatch);
```

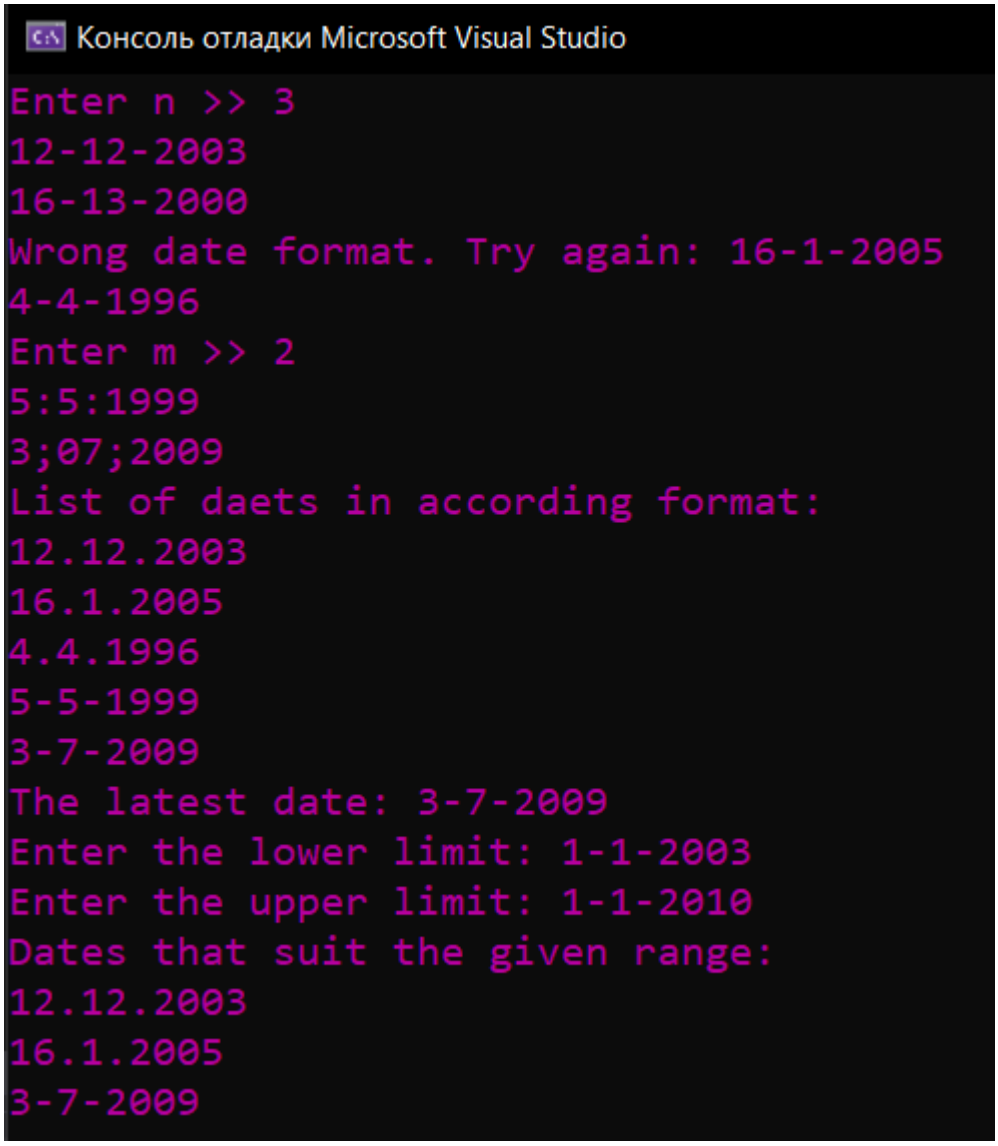
TDate.cpp

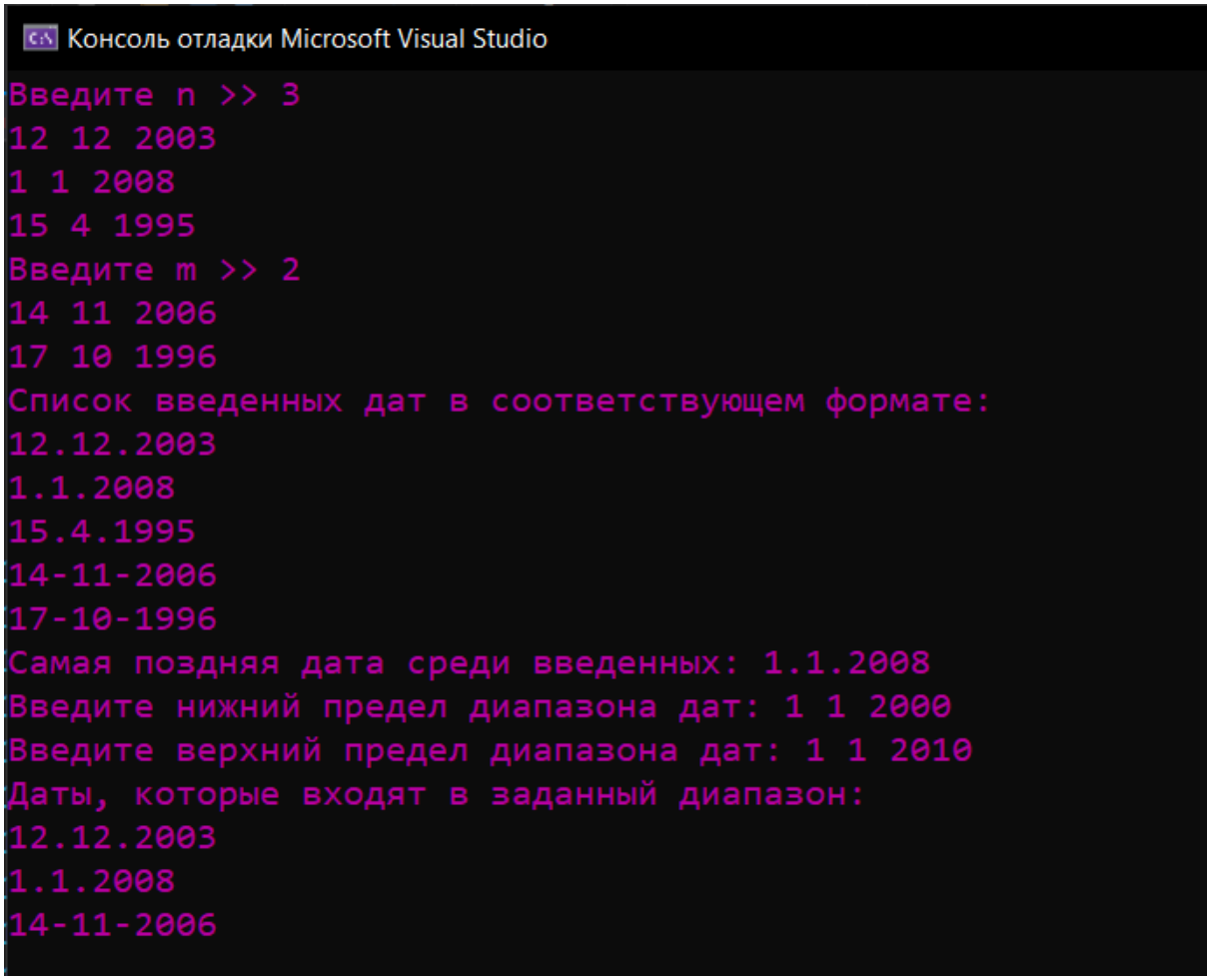
```
#include "TDate.h"

TDate::TDate() {
    this->day = 1;
    this->month = 1;
    this->year = 1970;
}

TDate::TDate(string line) {
    regex regular("([\\d]{1,2})[- /:;.,_]([\\d]{1,2})[- /:;.,_]([\\d]{0,4})");
    cmatch result;
    while (!regex_match(line.c_str(), result, regular) || !IsDateValid(result)) {
        cerr << "Wrong date format. Try again: ";
        getline(cin, line);
    }
    day = stoi(result[1]);
    month = stoi(result[2]);
    year = stoi(result[3]);
```

```
}

int TDate::GetDays() {
    return (this->year * 365 + this->month * 30 + this->day);
}

void TDate::IncreaseDate(int day, int month, int year) {
    int days[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
    while (day > 0) {
        this->day++;
        if (this->day > days[this->month - 1]) {
            this->month++;
            if (this->month > 12) {
                this->year++;
                this->month = 1;
            }
            this->day = 1;
        }
        day--;
    }
    while (month > 0) {
        this->month++;
        if (this->month > 12) {
            this->month = 1;
            this->year++;
        }
        month--;
    }
    this->year += year;
}

void TDate::DecreaseDate(int day, int month, int year) {
    int days[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
    while (day > 0) {
        this->day--;
        if (this->day == 0) {
            this->month--;
            if (this->month == 0) {
                this->month = 12;
                this->year--;

            }
            this->day = days[this->month - 1];
        }
        day--;
    }
    while (month > 0) {
        this->month--;
        if (this->month == 0) {
            this->year--;
            this->month = 12;
        }
        month--;
    }
    this->year -= year;
}

bool TDate::CompareDates(TDate& other) {
    if (this->GetDays() > other.GetDays()) {
        return true;
    }
    else {
        return false;
    }
```

```
}

bool TDate::IsInTimeInterval(string date1, string date2) {
    TDate lower_date = TDate(date1);
    TDate upper_date = TDate(date2);
    int lower = lower_date.GetDays();
    int upper = upper_date.GetDays();
    return (this->GetDays() >= lower && this->GetDays() <= upper);
}

bool IsDateValid(cmatch date) {
    int day = stoi(date[1]);
    int month = stoi(date[2]);
    int year = stoi(date[3]);
    if ((year % 4 == 0) && (month == 2) && (day > 29) || (year % 4 != 0) && (month == 2)
&& (day > 28)) {
        return false;
    }
    if (((month == 4) || (month == 6) || (month == 9) || (month == 11)) && (day > 30)) {
        return false;
    }
    if (((month == 1) || (month == 3) || (month == 5) || (month == 7) || (month == 8) ||
        (month == 10) || (month == 12)) && (day > 31)) {
        return false;
    }
    if (month > 12 || month < 1) return false;
    return true;
}

void TDate::ShowDate() {
    printf("%d %d %d\n", day, month, year);
}

void PrintVector(vector <TDate*> base) {
    for (auto& s : base) {
        s->ShowDate();
    }
}

TDate1::TDate1(string line) :TDate(line) {}

TDate2::TDate2(string line) : TDate(line) {}

void TDate1::ShowDate() {
    printf("%d.%d.%d\n", day, month, year);
}

void TDate2::ShowDate() {
    printf("%d-%d-%d\n", day, month, year);
}
```

Тестування:


Консоль отладки Microsoft Visual Studio
Enter n >> 3
12-12-2003
16-13-2000
Wrong date format. Try again: 16-1-2005
4-4-1996
Enter m >> 2
5:5:1999
3;07;2009
List of daets in according format:
12.12.2003
16.1.2005
4.4.1996
5-5-1999
3-7-2009
The latest date: 3-7-2009
Enter the lower limit: 1-1-2003
Enter the upper limit: 1-1-2010
Dates that suit the given range:
12.12.2003
16.1.2005
3-7-2009


Консоль отладки Microsoft Visual Studio
Введите n >> 3
12 12 2003
1 1 2008
15 4 1995
Введите m >> 2
14 11 2006
17 10 1996
Список введенных дат в соответствующем формате:
12.12.2003
1.1.2008
15.4.1995
14-11-2006
17-10-1996
Самая поздняя дата среди введенных: 1.1.2008
Введите нижний предел диапазона дат: 1 1 2000
Введите верхний предел диапазона дат: 1 1 2010
Даты, которые входят в заданный диапазон:
12.12.2003
1.1.2008
14-11-2006

Python

main.py

```
from TDate import *

n = int(input("Введите n: "))
lst = [TDate1(input()) for i in range(n)]
m = int(input("Введите m: "))
lst += [TDate2(input()) for j in range(m)]
print(max(lst, key=lambda x: x.get_days_since_year_0()))
date1 = input("Введите нижний предел диапазона дат: ")
date2 = input("Введите верхний предел диапазона дат: ")
print("Даты, которые входят в заданный диапазон")
for i in lst:
    if i.is_in_time_interval(date1, date2):
        print(i)
```

TDate.py

```
import re
from abc import ABC, abstractmethod

DAYS = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]


class TDate(ABC):
    day = 1
    month = 1
    year = 1

    def __init__(self, line: str):
        symbols = ['.', '-', ':', ',', ';']
        for i in line:
            if i in symbols:
                line = line.replace(i, ' ')
        line = line.split()
        self.day = int(line[0])
        self.month = int(line[1])
        self.year = int(line[2])

    def increase_date(self, day=0, month=0, year=0):
        while day != 0:
            self.day += 1
            if self.day > DAYS[self.month - 1]:
                self.day = 1
                self.month += 1
                if self.month > 12:
                    self.year += 1
                    self.month = 1
            day -= 1
        while month != 0:
            self.month += 1
            if self.month > 12:
```

```
                self.year += 1
                self.month = 1
            month -= 1
        self.year += year

    def decrease_date(self, day=0, month=0, year=0):
        while day != 0:
            self.day -= 1
            if self.day == 0:
                self.month -= 1
                if self.month == 0:
                    self.year -= 1
                    self.month = 12
                self.day = DAYS[self.month-1]
            day -= 1
        while month != 0:
            self.month -= 1
            if self.month == 0:
                self.month = 12
                self.year -= 1
            month -= 1
        self.year -= year

    def is_in_time_interval(self, date1, date2):
        date1 = TDate1(date1).get_days_since_year_0()
        date2 = TDate1(date2).get_days_since_year_0()
        if date1 <= self.get_days_since_year_0() <= date2:
            return True

    def get_days_since_year_0(self):
        return self.year * 365 + self.month * 30 + self.day

    @abstractmethod
    def __str__(self):
        pass


class TDate1(TDate):

    def __str__(self):
        return f"{self.day}.{self.month}.{self.year}"


class TDate2(TDate):

    def __str__(self):
        return f"{self.day}-{self.month}-{self.year}"
```

Тестування:

```
Введите n: 3
12 12 2003
15 2 2000
1 8 1995
Введите m: 2
16 2 2002
14 9 1995
12.12.2003
Введите нижний предел диапазона дат: 1 1 2000
Введите верхний предел диапазона дат: 1 1 2010
Даты, которые входят в заданный диапазон
12.12.2003
15.2.2000
16-2-2002
```

Висновки:

Я вивчив та використав на практиці механізми створення класів та об’єктів.