

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)

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

# 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

```
#include "TDate.h"

int main() {
    setlocale(LC_ALL, "rus");
    vector<TDate*> base;
    int n, m;
    string str;
    cout << "Введите n >> "; cin >> n;
    cin.ignore();
    for (int i = 0; i < n; i++) {
        getline(cin, str);
        base.push_back(new TDate1(str));
    }
    cout << "Введите m >> "; cin >> m;
    cin.ignore();
    for (int i = 0; i < m; i++) {
        getline(cin, str);
        base.push_back(new TDate2(str));
    }
    cout << "Список введенных дат в соответствующем формате:\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 << "Самая поздняя дата среди введенных: ";
    latest->ShowDate();
    string date1, date2;
    cout << "Введите нижний предел диапазона дат: ";
    getline(cin, date1);
    cout << "Введите верхний предел диапазона дат: ";
    getline(cin, date2);
    cout << "Даты, которые входят в заданный диапазон:\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>

using namespace std;

class TDate
{
protected:
    int day = 1;
    int month = 1;
    int year = 1970;
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 line):TDate(line) {};
    void ShowDate() override;
};

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

vector<string> timesplit(string line);
```

## TDate.cpp

```
#include "TDate.h"

TDate::TDate(string line) {
    this->day = stoi(timesplit(line)[0]);
    this->month = stoi(timesplit(line)[1]);
    this->year = stoi(timesplit(line)[2]);
}

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);
}

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

```

```

}

vector<string> timesplit(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;
}

void PrintVector(vector <TDate*> base) {
    for (int i = 0; i < base.size(); i++) {
        base[i]->ShowDate();
    }
}

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

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
        while year != 0:
            self.year += 1
            year -= 1
```

```

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

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.