

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

Варіант 23

Виконав студент

ІІ-15, Мочалов Дмитро Юрійович
(шифр, прізвище, ім'я, по батькові)

Перевірів

Муха Ірина Павлівна
(прізвище, ім'я, по батькові)

Київ 2022

Завдання

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 C#

```
using System;
using System.Collections.Generic;
using static Lab5.Functions;


namespace Lab5
{
    internal class Program
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("Скільки об'єктів TDate1 бажаєте створити: ");
            int n = int.Parse(Console.ReadLine());
            Console.WriteLine("Скільки об'єктів TDate2 бажаєте створити: ");
            int m = int.Parse(Console.ReadLine());
            List<TDate1> listOfDate1 = new List<TDate1>();
            Input1(listOfDate1, n);
            Console.WriteLine();
            List<TDate2> listOfDate2 = new List<TDate2>();
            Input2(listOfDate2, m);
            Console.WriteLine();
            string oldestDate = OldestDate(listOfDate1, listOfDate2);
            Console.WriteLine("Сама пізня дата: " + oldestDate);
            Console.WriteLine();
            Console.WriteLine("Введіть проміжок дат в одному з двох форматів");
            Console.WriteLine("Початок: ");
            string begin;
            while (true)
            {
                begin = Console.ReadLine();
                if (IsValid1(begin) || IsValid2(begin)) break;
                Console.WriteLine("Ви ввели в неправильному форматі, спробуйте ще раз: ");
            }
            Console.WriteLine("Кінець: ");
            string end;
            while (true)
            {
                end = Console.ReadLine();
                if (IsValid1(end) || IsValid2(end)) break;
                Console.WriteLine("Ви ввели в неправильному форматі, спробуйте ще раз: ");
            }
        }
    }
}
```

```
List<string> resultList = ListDatesBetween(listOfDate1, listOfDate2, date1: begin, date2: end)
Console.WriteLine("Результат:");
if (resultList.Count == 0)
{
    Console.WriteLine("Немає дат з цього проміжку");
}
else
{
    for (int i = 0; i < resultList.Count; i++)
    {
        Console.WriteLine($"Дата {i+1}: {resultList[i]}");
    }
}
```

using ...

namespace Lab5

{

 2 usages  2 inheritors

public abstract class TDate

{

protected int day;
protected int month;
protected int year;

public int Day

{

set ⇒ day = value;
get ⇒ day;

}

public int Month

{

set ⇒ month = value;
get ⇒ month;


}

public int Year

{

set ⇒ year = value;
get ⇒ year;

}

 6 usages

public int CompareTo(string date)

{

string[] numbers;
int day1 = 0;
int month1 = 0;
int year1 = 0;
if (date.Contains('.'))
{
 numbers = date.Split(params separator: '.');
 day1 = int.Parse(numbers[0]);
 month1 = int.Parse(numbers[1]);
 year1 = int.Parse(numbers[2]);
}

```

    }else if (date.Contains('-'))
    {
        numbers = date.Split( params separator: '-');
        day1 = int.Parse(numbers[1]);
        month1 = int.Parse(numbers[0]);
        year1 = int.Parse(numbers[2]);
    }

    if (this.year < year1 ||
        this.year == year1 && (this.month < month1 || this.month == month1 && this.day < day1)) return -1;
    if (this.year == year1 && this.month == month1 && this.day == day1) return 0;
    return 1;
}

public void AddYears(int count)
{
    if (count < 0)
    {
        throw new Exception( message: "unvalid value");
    }
    this.year += count;
}

```

```

public void AddMonth(int count)
{
    if (count < 0)
    {
        throw new Exception( message: "unvalid value");
    }
    int years = count / 12;
    int remainder = count % 12;
    this.year += years;
    this.month += remainder;
    if (this.month > 12)
    {
        this.year++;
        this.month -= 12;
    }
}

```

```
public void AddDays(int count)
{
    if (count < 0)
    {
        throw new Exception( message: "invalid value");
    }
    int years = count / 365;
    int month = (count % 365) / 30;
    int days = (count % 365) % 30;
    this.year += years;
    this.month += month;
    if (this.month > 12)
    {
        this.year++;
        this.month -= 12;
    }

    this.day += days;
    if (this.day > 30)
    {
        if (this.month < 12)
        {
            this.month++;
        }
        else
        {
            this.year++;
        }

        this.day -= 30;
    }
}

public void SubtractYears(int count)
{
    if (count < 0)
    {
        throw new Exception( message: "invalid value");
    }
}
```

```
public void SubtractYears(int count)
{
    if (count < 0)
    {
        throw new Exception(message: "invalid value");
    }

    this.year -= count;
}

public void SubtractMonth(int count)
{
    if (count < 0)
    {
        throw new Exception(message: "invalid value");
    }

    int years = count / 12;
    int month = count % 12;
    this.year -= years;
    this.month -= month;
    if (month < 0)
    {
        this.year--;
        this.month += 12;
    }
}

public void SubtractDays(int count)
{
    if (count < 0)
    {
        throw new Exception(message: "invalid value");
    }

    int years = count / 365;
    int month = (count % 365) / 30;
    int days = (count % 365) % 30;
```

```
int month = (count % 365) / 30;
int days = (count % 365) % 30;
this.year -= years;
this.month -= month;
if (this.month < 0)
{
    this.year--;
    this.month += 12;
}

this.day -= days;
if (this.day < 30)
{
    if (this.month == 1)
    {
        this.month = 12;
        this.year--;
    }

    this.day += 30;
}
}
}
```


6 usages

```
public class TDate1 : TDate
{
    1 usage
    public TDate1(string date)
    {
        string[] numbers = date.Split( params separator: '.');
        this.day = int.Parse(numbers[0]);
        this.month = int.Parse(numbers[1]);
        this.year = int.Parse(numbers[2]);
    }


    3 usages
    public override string ToString()
    {
        return $"{this.day}.{this.month}.{this.year}";
    }
}
```


6 usages

```
public class TDate2 : TDate
{
    1 usage
    public TDate2(string date)
    {
        string[] numbers = date.Split( params separator: '-');
        this.day = int.Parse(numbers[1]);
        this.month = int.Parse(numbers[0]);
        this.year = int.Parse(numbers[2]);
    }

    2 usages
    public override string ToString()
    {
        return $"{this.month}-{this.day}-{this.year}";
    }
}
```

```
using System;
using System.Collections.Generic;

namespace Lab5
{
     1 usage
    public class Functions
    {
         3 usages
        public static bool isValid1(string date)
        {
            string[] arr = date.Split( params separator: '.');
            foreach (var VARIABLE:string in arr)
            {
                if (!int.TryParse(VARIABLE, out int number)) return false;
            }
            int day = Int32.Parse(arr[0]);
            int month = Int32.Parse(arr[1]);
            int year = Int32.Parse(arr[2]);
            if ((day < 1 || day > 31) || (month < 1 || month > 12)) return false;
            return true;
        }

         3 usages
        public static bool isValid2(string date)
        {
            string[] arr = date.Split( params separator: '-');
            foreach (var VARIABLE:string in arr)
            {
                if (!int.TryParse(VARIABLE, out int number)) return false;
            }
            int day = Int32.Parse(arr[1]);
            int month = Int32.Parse(arr[0]);
            int year = Int32.Parse(arr[2]);
            if (day < 1 || day > 31 || month < 1 || month > 12) return false;
            return true;
        }
    }
}
```

1 usage

```
public static string OldestDate(List<TDate1> list1, List<TDate2> list2)
{
    string result = list1[0].ToString();
    for (int i = 1; i < list1.Count; i++)
    {
        if (list1[i].CompareTo(result) < 0) result = list1[i].ToString();
    }

    for (int i = 0; i < list2.Count; i++)
    {
        if (list2[i].CompareTo(result) < 0) result = list2[i].ToString();
    }

    return result;
}
```

1 usage

```
public static List<string> ListDatesBetween(List<TDate1> list1, List<TDate2> list2, string date1, string date2)
{
    List<string> result = new List<string>();
    for (int i = 0; i < list1.Count; i++)
    {
        if (list1[i].CompareTo(date1) ≥ 0 && list1[i].CompareTo(date2) ≤ 0)
        {
            result.Add(item: list1[i].ToString());
        }
    }
    for (int i = 0; i < list2.Count; i++)
    {
        if (list2[i].CompareTo(date1) ≥ 0 && list2[i].CompareTo(date2) ≤ 0)
        {
            result.Add(item: list2[i].ToString());
        }
    }

    return result;
}
```

```

public static void Input1(List<TDate1> list,int n)
{
    for (int i = 0; i < n; i++)
    {
        Console.WriteLine("Введіть дату для першого об'єкта в форматі 'ЧЧ.ММ.РРРР': ");
        string date;
        do
        {
            date = Console.ReadLine();
            if (IsValid1(date))
            {
                break;
            }

            Console.WriteLine("Ви ввели дату в неправильному форматі, спробуйте ще раз: ");
        } while (true);
        list.Add(item: new TDate1(date));
    }
}

1 usage
public static void Input2(List<TDate2> list,int m)
{
    for (int i = 0; i < m; i++)
    {
        Console.WriteLine("Введіть дату для другого об'єкта в форматі 'ММ-ЧЧ-РРРР': ");
        string date;
        do
        {
            date = Console.ReadLine();
            if (IsValid2(date))
            {
                break;
            }

            Console.WriteLine("Ви ввели дату в неправильному форматі, спробуйте ще раз: ");
        } while (true);
        list.Add(item: new TDate2(date));
    }
}

```

Код Python

```

n = int(input("Скільки об'єктів TDate1 бажаєте створити: "))
m = int(input("Скільки об'єктів TDate2 бажаєте створити: "))
list_of_Date1 = []
input1(list_of_Date1, n)

list_of_Date2 = []
input2(list_of_Date2, m)
oldest_Date = oldest_date(list_of_Date1, list_of_Date2)

print('Сама пізня дата: ' + oldest_Date + '\n')
print("Введіть проміжок дат в одному з двох форматів")
while(True):
    begin = input('Початок: ')
    if is_valid1(begin) or is_valid2(begin):
        break
    print('Ви ввели в неправильному форматі, спробуйте ще раз')
while(True):
    end = input('Кінець: ')
    if is_valid1(end) or is_valid2(end):
        break
    print('Ви ввели в неправильному форматі, спробуйте ще раз')

result_list = list_dates_between(list_of_Date1, list_of_Date2, begin, end)
print('Результат: ')
if len(result_list) == 0:
    print('Немає дат з цього проміжку')
else:
    for i in range(len(result_list)):
        print("Дата {0}: {1}".format(i+1, result_list[i]))

```

```
class TDate:
    def __init__(self):
        self._day = 0
        self._month = 0
        self._year = 0

    def compare_to(self, date):
        day1 = 0
        month1 = 0
        year1 = 0
        if '.' in date:
            numbers = date.split('.')
            day1 = int(numbers[0])
            month1 = int(numbers[1])
            year1 = int(numbers[2])
        elif '-' in date:
            numbers = date.split('-')
            day1 = int(numbers[1])
            month1 = int(numbers[0])
            year1 = int(numbers[2])
        if self._year < year1 or (
            self._year == year1 and (self._month < month1 or self._month == month1 and (self._day < day1))):
            return -1
        if self._year == year1 and self._month == month1 and self._day == day1:
            return 0
        return 1
```

```
def add_years(self, count):
    if count < 0:
        raise Exception("unvalid value")
    self._year += count

def add_month(self, count):
    if count < 0:
        raise Exception("unvalid value")
    years = count // 12
    remainder = count % 12
    self._year += years
    self._month += remainder
    if self._month > 12:
        self._year += 1
        self._month -= 12

def add_days(self, count):
    if count < 0:
        raise Exception("unvalid value")
    years = count // 365
    month_ = count % 365 // 30
    days = (count % 365) % 30
    self._year += years
    self._month += month_
    if self._month > 12:
        self._year += 1
        self._month -= 12
    self._day += days
    if self._day > 30:
        if self._month < 12:
```

```
self._day += days
if self._day > 30:
    if self._month < 12:
        self._month += 1
    else:
        self._year += 1
    self._day -= 30

def subtract_years(self, count):
    if count < 0:
        raise Exception("unvalid value")
    self._year -= count

def subtract_month(self, count):
    if count < 0:
        raise Exception("unvalid value")
    years = count // 12
    month_ = count % 12
    self._year -= years
    self._month -= month_
    if month_ < 0:
        self._year -= 1
        self._month += 12
```



```
def subtract_days(self, count):
    if count < 0:
        raise Exception("unvalid value")
    years = count // 365
    month_ = (count % 365) // 30
    days = (count % 365) % 30
    self._year -= years
    self._month -= month_
    if self._month < 0:
        self._year -= 1
        self._month += 12
    self._day -= days
    if self._day < 30:
        if self._month == 1:
            self._month = 12
            self._year -= 1
        self._day += 30
```

```
class TDate1(TDate):
    def __init__(self, date):
        super().__init__()
        numbers = date.split('.')
        self._day = int(numbers[0])
        self._month = int(numbers[1])
        self._year = int(numbers[2])

    def __str__(self):
        return "{0}.{1}.{2}".format(self._day, self._month, self._year)

class TDate2(TDate):
    def __init__(self, date):
        super().__init__()
        numbers = date.split('-')
        self._day = int(numbers[1])
        self._month = int(numbers[0])
        self._year = int(numbers[2])

    def __str__(self):
        return "{0}-{1}-{2}".format(self._month, self._day, self._year)
```

```
main.py × TDate.py × functions.py ×
1 from TDate import *
2
3
4 def is_valid1(date):
5     arr = date.split('.')
6     for symbol in arr:
7         if not symbol.isdigit():
8             return False
9     day = int(arr[0])
10    month = int(arr[1])
11    year = int(arr[2])
12    if day < 1 or day > 31 or month < 1 or month > 12:
13        return False
14    return True
15
16
17 def is_valid2(date):
18     arr = date.split('-')
19     for symbol in arr:
20         if not symbol.isdigit():
21             return False
22     day = int(arr[1])
23     month = int(arr[0])
24     year = int(arr[2])
25     if day < 1 or day > 31 or month < 1 or month > 12:
26         return False
27     return True
```

```
def oldest_date(list1, list2):
    result = list1[0].__str__()
    for i in range(len(list1)):
        if list1[i].compare_to(result) < 0:
            result = list1[i].__str__()
    for i in range(len(list2)):
        if list2[i].compare_to(result) < 0:
            result = list2[i].__str__()
    return result

def list_dates_between(list1, list2, date1, date2):
    result = []
    for ele in list1:
        if ele.compare_to(date1) >= 0 and ele.compare_to(date2) <= 0:
            result.append(ele.__str__())
    for ele in list2:
        if ele.compare_to(date1) >= 0 and ele.compare_to(date2) <= 0:
            result.append(ele.__str__())
    return result
```

```

def input1(list, n):
    for i in range(n):
        print("Введіть дату для першого об'єкта в форматі 'ЧЧ.ММ.РРРР': ")
        date = ''
        while(True):
            date = input()
            if is_valid1(date):
                break
            print("Ви ввели дату в неправильному форматі, спробуйте ще раз: ")
        list.append(TDate1(date))

def input2(list, n):
    for i in range(n):
        print("Введіть дату для другого об'єкта в форматі 'ММ-ЧЧ-РРРР': ")
        date = ''
        while(True):
            date = input()
            if is_valid2(date):
                break
            print("Ви ввели дату в неправильному форматі, спробуйте ще раз: ")
        list.append(TDate2(date))

```

Робота програми

C#

```

Скільки об'єктів в TDate1 бажаєте створити: 2
Скільки об'єктів в TDate2 бажаєте створити: 3
Введіть дату для першого об'єкта в формат? 'ЧЧ.ММ.РРРР': 32.15.2000
Ви ввели дату в неправильному формат?, спробуйте ще раз: 16.12.2003
Введіть дату для першого об'єкта в формат? 'ЧЧ.ММ.РРРР': 01.02.2015

Введіть дату для другого об'єкта в формат? 'ММ-ЧЧ-РРРР': 10-02-2022
Введіть дату для другого об'єкта в формат? 'ММ-ЧЧ-РРРР': 01-01-2000
Введіть дату для другого об'єкта в формат? 'ММ-ЧЧ-РРРР': 05-20-2003

Сама п'зня дата: 1-1-2000

Введіть проміжок дат в одному з двох форматів
Початок: 16.12.2003
Кінець: 01.01.2016
Результат:
Дата 1: 16.12.2003
Дата 2: 1.2.2015

```

Python

```
Скільки об'єктів TDate1 бажаєте створити: 2
Скільки об'єктів TDate2 бажаєте створити: 3
Введіть дату для першого об'єкта в форматі 'ЧЧ.ММ.РРРР':

Ви ввели дату в неправильному форматі, спробуйте ще раз:
16.15.2003
Ви ввели дату в неправильному форматі, спробуйте ще раз:
16.12.2003
Введіть дату для першого об'єкта в форматі 'ЧЧ.ММ.РРРР':
01.05.2012
Введіть дату для другого об'єкта в форматі 'ММ-ЧЧ-РРРР':
05-01-2000
Введіть дату для другого об'єкта в форматі 'ММ-ЧЧ-РРРР':
12-15-2003
Введіть дату для другого об'єкта в форматі 'ММ-ЧЧ-РРРР':
05-06-2014
Сама пізня дата: 5-1-2000

Введіть проміжок дат в одному з двох форматів
Початок: 16.12.2003
Кінець: 01.01.2016
Результат:
Дата 1: 16.12.2003
Дата 2: 1.5.2012
Дата 3: 5-6-2014
```