МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

Національний технічний університет України

Факультет інформатики та обчислювальної техніки

«Київський політехнічний інститут імені Ігоря Сікорського»

Кафедра інформаційних систем та технологій

Звіт

з лабораторної роботи № 2

з дисципліни

«Програмування – 2»

Варіант № 30

|  |  |  |
| --- | --- | --- |
| Перевірив: |  | Виконала: Шелест Поліна |
|  |  | Студентка гр. ІС-11 , ФІОТ |
|  |  | 1 курс |

Київ 2021

**Лабораторна робота № 2**

**Тема:** Поліморфізм. Наслідування. Перегрузка методів

**Завдання 1:** Створити додаток, який задовольняє вимогам, наведеним в завданні. Наслідування застосовувати тільки в тих завданнях, в яких воно логічно обґрунтоване. Аргументувати належність класу кожного створюваного методу і коректно перевизначити для кожного класу методи Equals, GetHashCode, ToString. При виклику будь-которого методу класу, виводити на екран текстове повідомлення.

30.Створити об'єкт класу Рік, використовуючи класи Місяць, День. Методи: задати дату, вивести на консоль день тижня по заданій даті, розрахувати кількість днів, місяців в заданому часовому проміжку.

**Код:**

Program.cs

using System;

namespace Lab2

{

internal class Program

{

static void Main()

{

Console.Write("Enter the date dd/MM/yyyy): ");

string newDateStr = Console.ReadLine();

string[] newDateArrStr = newDateStr.Split('/');

int[] newDateArr = new int[newDateArrStr.Length];

for (int i = 0; i < newDateArrStr.Length; i++)

newDateArr[i] = int.Parse(newDateArrStr[i]);

Date setDate = new Date(newDateArr[2], newDateArr[1], newDateArr[0]);

setDate.ShowDate();

setDate.ShowDayOfWeek();

setDate.CountTimeSpan();

Console.ReadKey();

}

}

}

Date.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab2

{

internal class Date

{

public Year year = new Year(0);

public Month month = new Month(0);

public Day day = new Day(0);

public Date(int \_year, int \_month, int \_day)

{

this.year.Value = \_year;

this.month.Value = \_month;

this.day.Value = \_day;

}

public void SetDate(int year, int month, int day)

{

this.year.Value = year;

this.month.Value = month;

this.day.Value = day;

}

public void ShowDate()

{

Console.WriteLine("\nSet date is {0}/{1}/{2}", day.Value, month.Value, year.Value);

}

public void ShowDayOfWeek()

{

DateTime dateTime = new DateTime(year.Value, month.Value, day.Value);

string dayOfWeek = dateTime.DayOfWeek.ToString();

Console.WriteLine($"\nDay of week is {dayOfWeek}");

}

public void CountTimeSpan()

{

Console.Write("\nEnter date to count timespan: ");

string newDateStr = Console.ReadLine();

string[] newDateArrStr = newDateStr.Split('/');

int[] newDateArr = new int[newDateArrStr.Length];

for(int i = 0; i < newDateArrStr.Length; i++)

newDateArr[i] = int.Parse(newDateArrStr[i]);

DateTime newDate = new DateTime(newDateArr[2], newDateArr[1], newDateArr[0]);

DateTime startDate = new DateTime(year.Value, month.Value, day.Value);

TimeSpan timeSpan = newDate.Subtract(startDate);

Console.WriteLine("\nTimespan is equal to: " + timeSpan.Days + " days and " + timeSpan.Days/30 + " months");

}

}

}

Year.cs

using System;

namespace Lab2

{

class Year

{

public int Value { get; set; }

public Year(int year)

{

this.Value = year;

}

}

}

Month.cs

using System;

namespace Lab2

{

internal class Month

{

public int Value { get; set; }

public Month(int month)

{

this.Value = month;

}

}

}

Day.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab2

{

internal class Day

{

public int Value { get; set; }

public Day(int day)

{

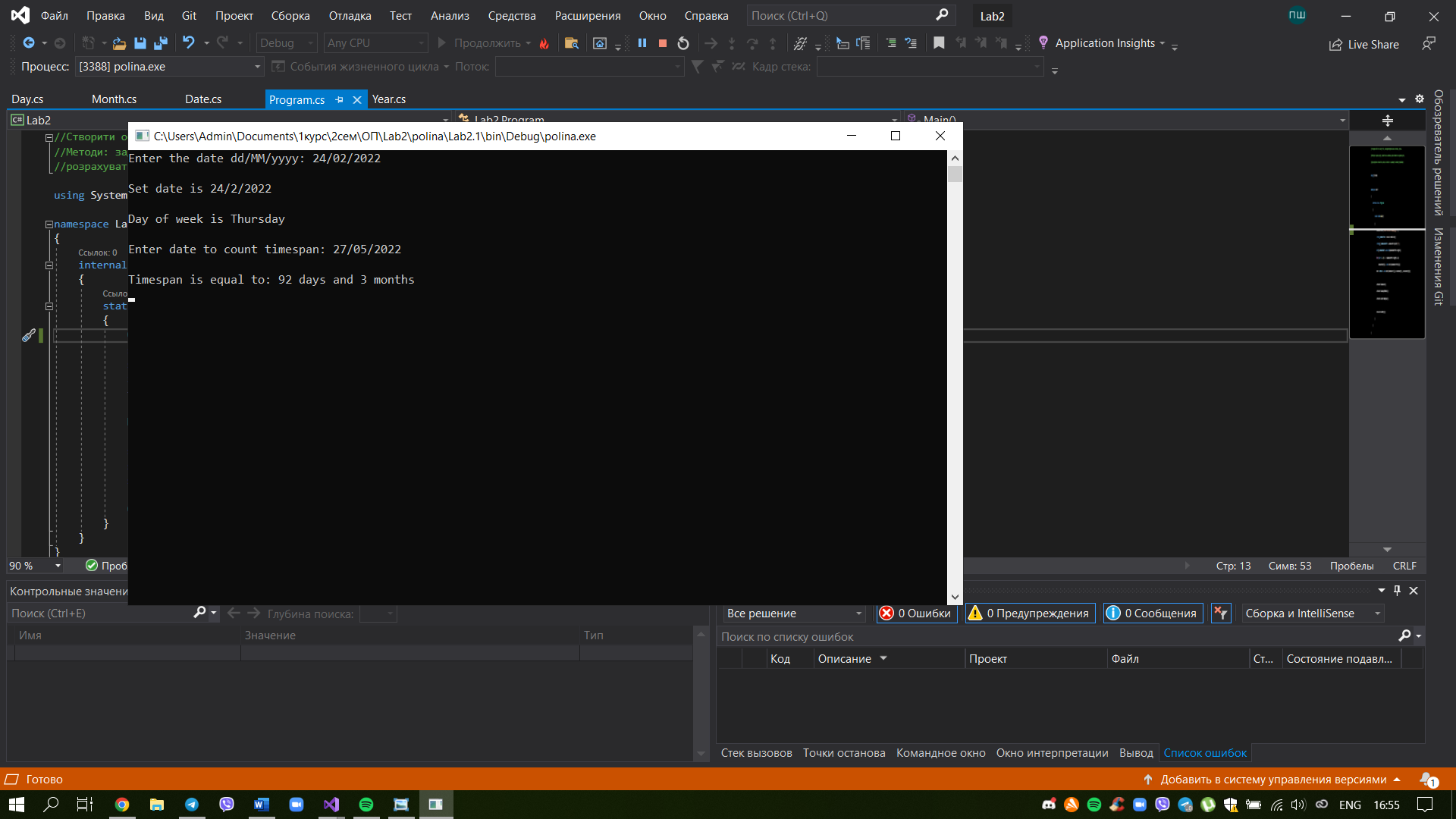
this.Value = day;

}

}

}

Результат виконання:



**Завдання 2:** Створити програму, яка задовольняє наступним вимогам:

• Використовувати можливості ООП: класи, наслідування, поліморфізм, інкапсуляція.

• Кожен клас повинен мати змістовну назву та інформативний склад.

• Наслідування має застосовуватися тільки тоді, коли це має сенс.

• Класи повинні бути грамотно розкладені по пакетах.

• Консольне меню повинно бути мінімальним.

30. Рахунки. Клієнт може мати кілька рахунків у банку. Враховувати можливість блокування / розблокування рахунку. Реалізувати пошук і сортування рахунків. Обчислення загальної суми по рахунках. Обчислення суми по всіх рахунках, мають позитивний і негативний баланси окремо.

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab2.Bank

{

class Program

{

static void Main(string[] args)

{

var client1 = new Client(7);

Console.WriteLine("Client ID is " + client1.Id);

var account1 = new Account(13);

client1.Accounts.Add(account1);

var foundAccount = client1.GetAccountById(13); //пошук рахунку з id=13

Console.WriteLine("Balance of account with ID = 13 is " + foundAccount.Balance);

client1.Accounts.Add(new Account(6) { Balance = -5 });;

client1.Accounts.Add(new Account(7) { Balance = 10, Blocked = true});

client1.Accounts.Add(new Account(5, true, 15));

var accountIds = client1.Accounts.Select(a => a.Id);

Console.WriteLine("IDs of client's acounts: " + string.Join(", ", accountIds));

var sortedAccounts = client1.Accounts.OrderBy(a => a.Balance);

var s = string.Join(", ", sortedAccounts.Select(a => a.Balance));

Console.WriteLine("Sorted accounts by balance (ascending): " + s);

var positivebalances = client1.SumOfPositive();

Console.WriteLine("Sum of positive balances of the client is " + positivebalances);

var negativebalances = client1.SumOfNegative();

Console.WriteLine("Sum of negative balances of the client is " + negativebalances);

Console.ReadLine();

}

}

}

Client.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab2.Bank

{

public class Client

{

private int \_id;

public int Id

{

get {

return \_id;

}

}

public List <Account> Accounts { get; }

public Client(int id)

{

\_id = id;

Accounts = new List <Account> ();

}

public Account GetAccountById(int accountId)

{

return Accounts.FirstOrDefault(x => x.Id == accountId);

}

public int SumOfPositive()

{

return Accounts

.Where(a => a.Balance > 0)

.Select(a => a.Balance)

.Sum();

}

public int SumOfNegative()

{

return Accounts

.Where(a => a.Balance < 0)

.Select(a => a.Balance)

.Sum();

}

}

Account.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab2.Bank

{

public class Account

{

public int Id { get; }

public bool Blocked { get; set; }

public int Balance { get; set; }

public Account(int id)

{

Id = id;

}

public Account(int id, bool blocked, int balance)

{

Id = id;

Blocked = blocked;

Balance = balance;

}

}

}

Результат виконання

