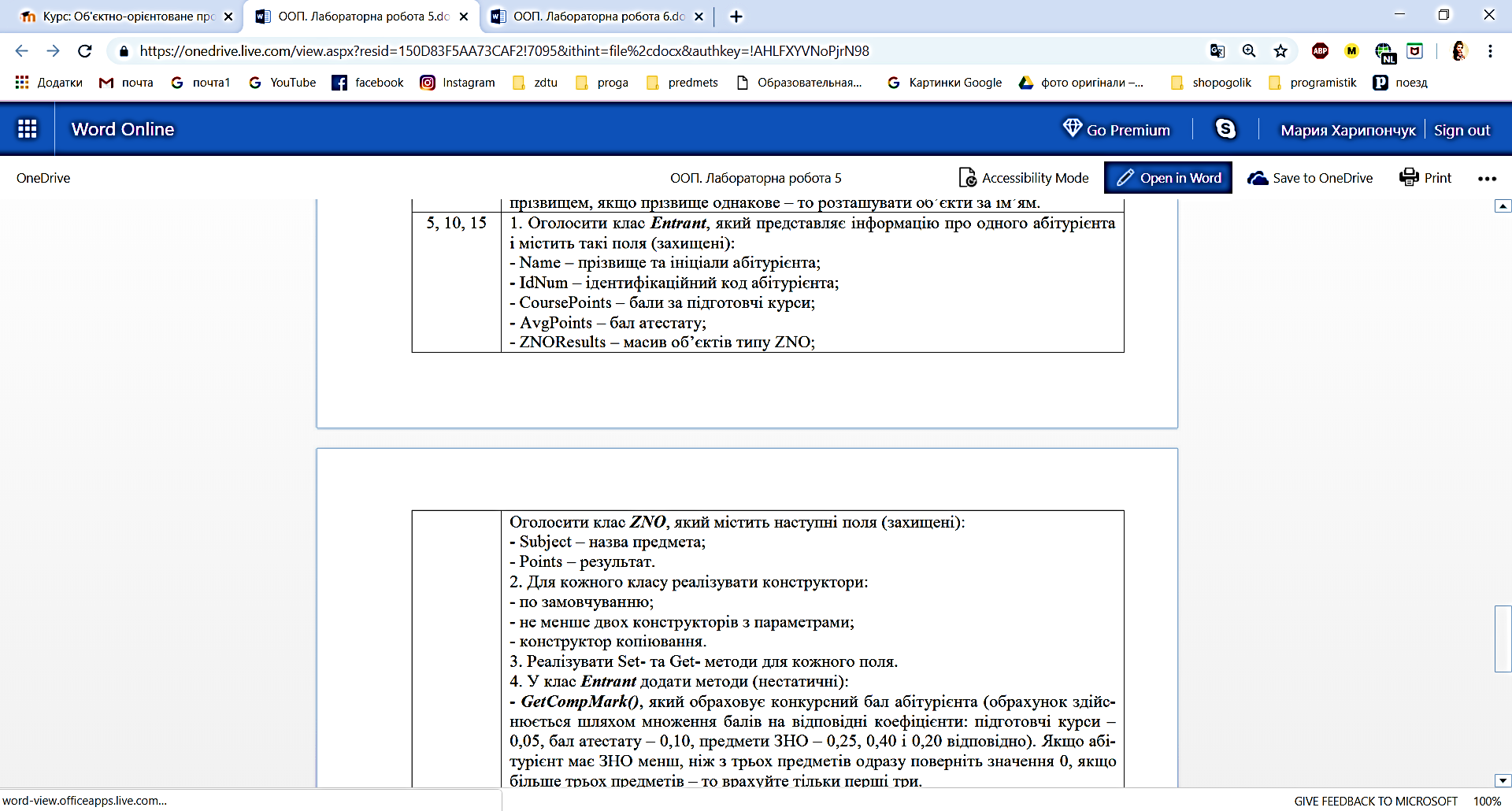
**ЛАБОРАТОРНА РОБОТА № 4**

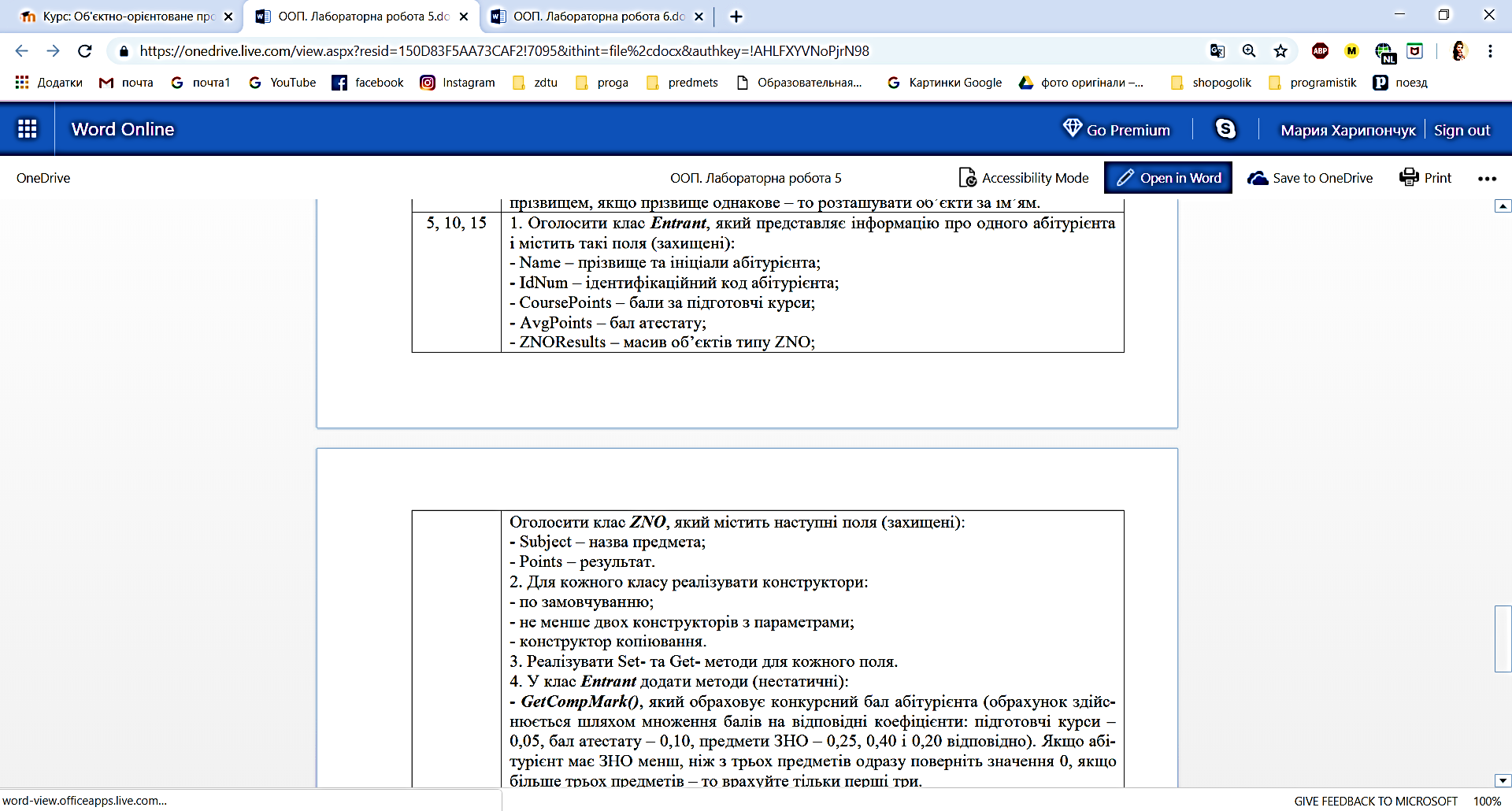
**Класи та об’єкти. Реалізація конструкторів та методів.**

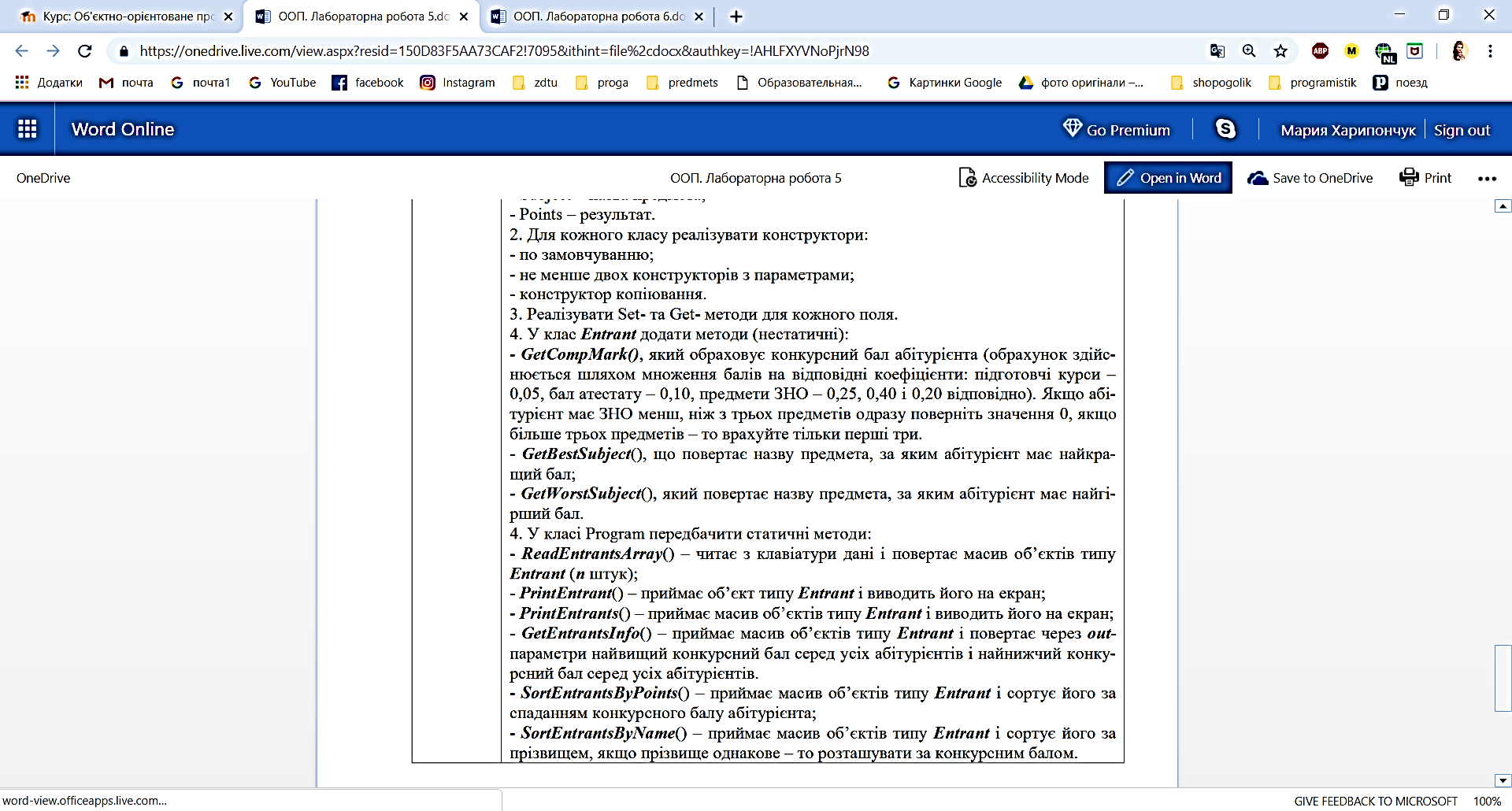
***Мета:*** засвоїти принципи проектування та оголошення класів; вивчити особливості реалізації трьох типів конструкторів (по замовчуванню, з параметрами та копіювання) та методів у класах.

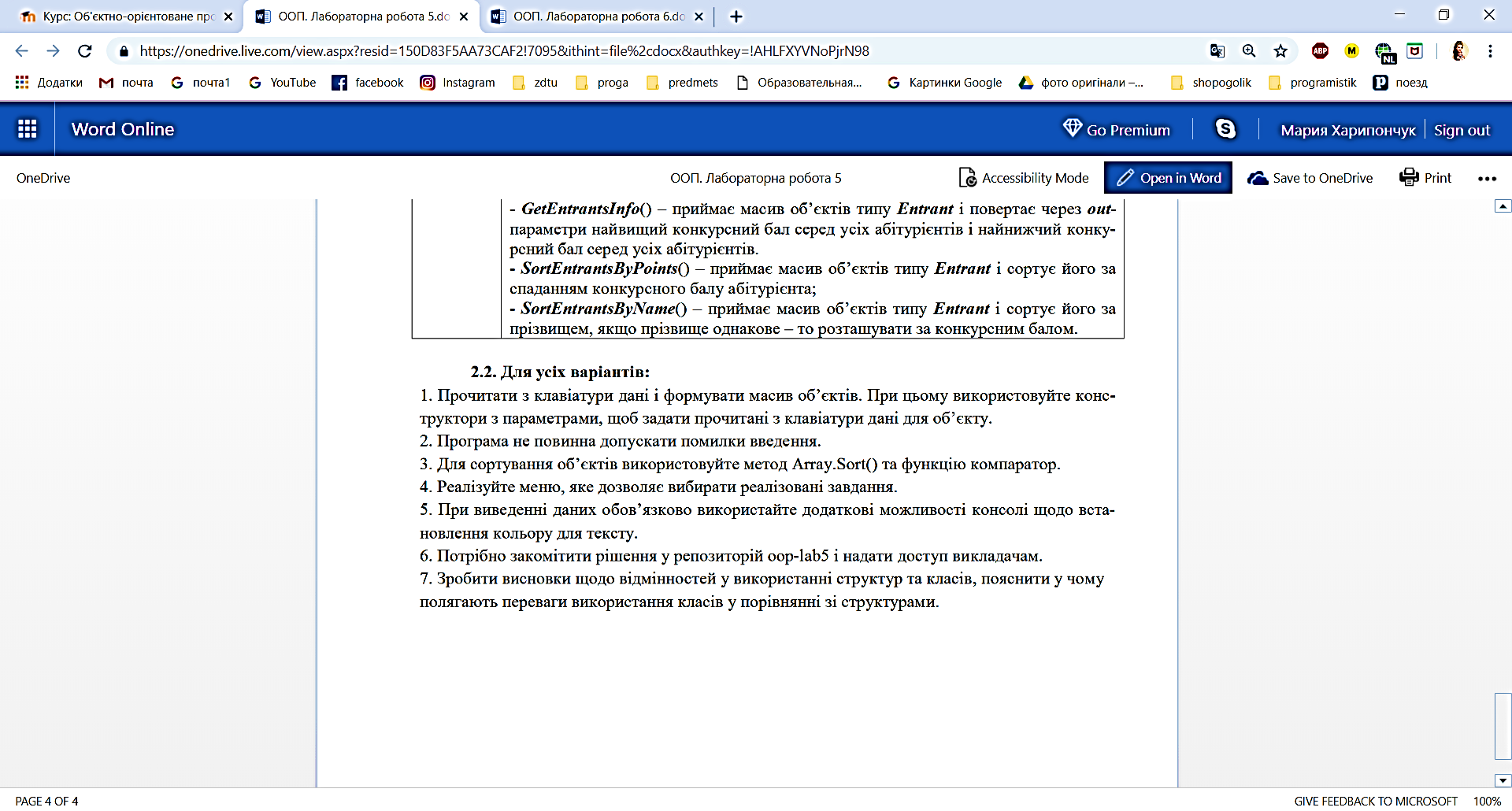
**Хід роботи:**

**Завдання:**









Лістинг програми (main):

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using static System.Console;

using SimpleClassLibrary;

namespace SimpleClassConlsole

{

class Program

{

static void CheckInt(out int x, int min, int max)

{

bool ok;

do

{

ok = int.TryParse(ReadLine(), out x);

if (!ok || x < min || x > max)

{

ForegroundColor = ConsoleColor.Red;

WriteLine(" Помилка! Введено некоректне значення. Будь-ласка повторіть введення ще раз!!!");

ResetColor();

}

} while (!ok || x < min || x > max);

}

static void CheckDouble(out double x, int min, int max)

{

bool ok;

do

{

ok = double.TryParse(ReadLine(), out x);

if (!ok || x < min || x > max)

{

ForegroundColor = ConsoleColor.Red;

WriteLine(" Помилка! Введено некоректне значення. Будь-ласка повторіть введення ще раз!!!");

ResetColor();

}

} while (!ok || x < min || x > max);

}

static public Entrant[] array = new Entrant[0];

static Entrant[] ReadEntrantsArray()

{

Write("Введiть к-сть елементiв масиву: ");

int n = GetValue();

int length = array.Length;

for (int i = length; i < length + n; i++)

{

Array.Resize(ref array, length + n);

array[i] = new Entrant();

array[i].Sub1 = new ZNO();

array[i].Sub2 = new ZNO();

array[i].Sub3 = new ZNO();

array[i].Sub4 = new ZNO();

WriteLine($"------- Абiтурiєнт[{i + 1}] -------");

Write("Прiзвище та iнiцiали абiтурiєнта: ");

array[i].Name = ReadLine();

Write("ID ");

array[i].IdNum = ReadLine();

Write("Бали за пiдготовчi курси (0 - 100) ");

CheckInt(out array[i].CoursePoints, 0, 100);

Write("Бал атестата (0 - 12) ");

CheckDouble(out array[i].AvgPoints, 0, 12);

Write("Назва 1 предмету ЗНО ");

array[i].Sub1.Subject = ReadLine();

Write("Бал (100 - 200) ");

CheckDouble(out array[i].Sub1.Points, 100, 200);

Write("Назва 2 предмету ЗНО ");

array[i].Sub2.Subject = ReadLine();

Write("Бал (100 - 200) ");

CheckDouble(out array[i].Sub2.Points, 100, 200);

Write("Назва 3 предмету ЗНО ");

array[i].Sub3.Subject = ReadLine();

Write("Бал (100 - 200) ");

CheckDouble(out array[i].Sub3.Points, 100, 200);

Write("Назва 4 предмету ЗНО ");

array[i].Sub4.Subject = ReadLine();

if (array[i].Sub4.Subject != "")

{

Write("Бал (100 - 200) ");

CheckDouble(out array[i].Sub4.Points, 100, 200);

}

array[i].CompMark = Entrant.GetCompMark(array[i]);

}

return array;

}

static void PrintTable()

{

ForegroundColor = ConsoleColor.Magenta;

WriteLine(" № | Прiзвище та iнiцiали працiвника | ID | Отриманi бали |");

ResetColor();

}

static void PrintEntrant(Entrant x)

{

Write($"{x.Name,35}{x.IdNum, 12}{null,10} ");

if (x.CoursePoints > 0)

Write($"Курси: {x.CoursePoints}\n{null,60}");

Write($"Атестат: {x.AvgPoints}\n{null,60}");

Write($"{x.Sub1.Subject}: {x.Sub1.Points}\n{null,60}");

Write($"{x.Sub2.Subject}: {x.Sub2.Points}\n{null,60}");

Write($"{x.Sub3.Subject}: {x.Sub3.Points}\n{null,60}");

if (x.Sub4.Points > 0)

Write($"{x.Sub4.Subject}: {x.Sub4.Points}\n{null,60}");

WriteLine($"Середнiй: {x.CompMark}\n");

}

static void PrintEntrants(Entrant[] array)

{

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

{

ForegroundColor = ConsoleColor.Magenta;

Write($"{i}");

ResetColor();

PrintEntrant(array[i]);

}

}

static void GetEntrantsInfo(Entrant[] array, out int maxCompMark, out int minCompMark)

{

minCompMark = 0;

maxCompMark = 0;

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

{

if (array[i].CompMark > array[maxCompMark].CompMark)

maxCompMark = i;

if (array[i].CompMark < array[minCompMark].CompMark)

minCompMark = i;

}

}

static void SortEntrantsByPoints(Entrant[] array)

{

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

{

array[i].CompMark = Entrant.GetCompMark(array[i]);

}

double[] arrayTmp = new double[array.Length];

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

arrayTmp[i] = array[i].CompMark;

Array.Sort(arrayTmp, array);

}

static void SortEntrantsByName(Entrant[] array)

{

string[] arrayTmp = new string[array.Length];

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

arrayTmp[i] = array[i].Name;

Array.Sort(arrayTmp, array);

}

static void GetInfoCost()

{

double costMonth = 2400;

double costYear = costMonth \* 10;

double costAllTime = costMonth \* 40;

ForegroundColor = ConsoleColor.DarkMagenta;

WriteLine("Вартість оплати навчання:\n" +

"1 - за місяць;\n" +

"2 - за рік;\n" +

"3 - за весь період навчання;\n" +

"0 - Вихiд\n");

ResetColor();

int num = GetValue();

switch (num)

{

case 1:

Clear();

WriteLine($"Вартість оплати навчання за місяць: {costMonth}грн. \n");

break;

case 2:

Clear();

WriteLine($"Вартість оплати навчання за рік: {costYear}грн. \n");

break;

case 3:

Clear();

WriteLine($"Вартість оплати навчання за весь період навчання: {costAllTime}грн. \n");

break;

case 0:

break;

default:

ForegroundColor = ConsoleColor.Red;

WriteLine(" Помилка! Введено некоректне значення. Будь-ласка повторiть введення ще раз!!!");

ResetColor();

break;

}

}

static int GetValue()

{

int x;

bool ok = false;

do

{

ok = int.TryParse(ReadLine(), out x);

if (ok == false)

{

ForegroundColor = ConsoleColor.Red;

WriteLine(" Помилка! Введено некоректне значення. Будь-ласка по-вторiть введення ще раз!!!");

ResetColor();

}

} while (!ok);

return x;

}

//----- Menu -----

static void ShowMenu()

{

BackgroundColor = ConsoleColor.Red;

ForegroundColor = ConsoleColor.White;

WriteLine(" МЕНЮ ");

ResetColor();

ForegroundColor = ConsoleColor.DarkMagenta;

WriteLine("1 - Ввести масив студентiв\n" +

"2 - Вивести записи\n" +

"3 - Сортувати масив студентiв\n" +

"4 - Iнформацiя про конкурсний бал\n" +

"5 - Iнформацiя про вартість оплати навчання\n" +

"0 - Вийти\n");

ResetColor();

}

static void Main(string[] args)

{

InputEncoding = Encoding.Unicode;

OutputEncoding = Encoding.Unicode;

System.Globalization.CultureInfo customCulture = (System.Globalization.CultureInfo)

System.Threading.Thread.CurrentThread.CurrentCulture.Clone();

customCulture.NumberFormat.NumberDecimalSeparator = ".";

System.Threading.Thread.CurrentThread.CurrentCulture = customCulture;

Title = "Лабораторна робота №5, Харипончук М.М., Варіант №10";

SetWindowSize(100, 25);

BackgroundColor = ConsoleColor.White;

ForegroundColor = ConsoleColor.Black;

int check;

Entrant[] array = null;

do

{

WriteLine("\n->");

ReadKey();

Clear();

ShowMenu();

check = GetValue();

switch (check)

{

case 1:

Clear();

array = ReadEntrantsArray();

break;

case 2:

if (array == null)

{

ForegroundColor = ConsoleColor.Red;

WriteLine("Помилка!!! Спочатку додайте данi");

ResetColor();

break;

}

ForegroundColor = ConsoleColor.DarkMagenta;

WriteLine("1 - Усi записи");

WriteLine("2 - Один запис");

WriteLine("0 - Вихiд");

ResetColor();

int ch = GetValue();

switch (ch)

{

case 1:

Clear();

PrintTable();

PrintEntrants(array);

break;

case 2:

Clear();

ForegroundColor = ConsoleColor.DarkMagenta;

WriteLine($"Введiть номер студента:\n(всього {array.Length} записiв)");

ResetColor();

int x = GetValue() - 1;

if (x > array.Length - 1 || x < 0)

{

ForegroundColor = ConsoleColor.Red;

WriteLine(" Помилка! Введено некоректне значення. Будь-ласка повторiть введення ще раз!!!");

ResetColor();

break;

}

else

{

PrintTable();

PrintEntrant(array[x]);

}

break;

case 0:

break;

default:

ForegroundColor = ConsoleColor.Red;

WriteLine(" Помилка! Введено некоректне значення. Будь-ласка повторiть введення ще раз!!!");

ResetColor();

break;

}

break;

case 3:

if (array == null)

{

ForegroundColor = ConsoleColor.Red;

WriteLine("Помилка!!! Спочатку додайте данi");

ResetColor();

break;

}

ForegroundColor = ConsoleColor.DarkMagenta;

WriteLine("1 - Сортувати за середнім балом;\n" +

"2 - Сортувати за прiзвищем;\n" +

"0 - Вихiд\n");

ResetColor();

int num = GetValue();

switch (num)

{

case 1:

SortEntrantsByPoints(array);

Clear();

PrintTable();

PrintEntrants(array);

break;

case 2:

SortEntrantsByName(array);

Clear();

PrintTable();

PrintEntrants(array);

break;

case 0:

break;

default:

ForegroundColor = ConsoleColor.Red;

WriteLine(" Помилка! Введено некоректне значення. Будь-ласка повторiть введення ще раз!!!");

ResetColor();

break;

}

break;

case 4:

if (array == null)

{

ForegroundColor = ConsoleColor.Red;

WriteLine("Помилка!!! Спочатку додайте данi");

ResetColor();

break;

}

Clear();

int minCompMark, maxCompMark;

GetEntrantsInfo(array, out maxCompMark, out minCompMark);

WriteLine($"Найвищий конкурсний бал: {array[maxCompMark].CompMark}\nНайнижчий конкурсний бал: {array[minCompMark].CompMark}");

break;

case 5:

if (array == null)

{

ForegroundColor = ConsoleColor.Red;

WriteLine("Помилка!!! Спочатку додайте данi");

ResetColor();

break;

}

Clear();

GetInfoCost();

break;

case 0:

break;

default:

ForegroundColor = ConsoleColor.Red;

WriteLine(" Помилка! Введено некоректне значення. Будь-ласка повторiть введення ще раз!!!");

ResetColor();

break;

}

} while (check != 0);

}

}

}

Лістинг бібліотеки SimpleClassLibrary

Entrant

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using static System.Console;

namespace SimpleClassLibrary

{

public class Entrant

{

protected string name; //прізвище та ініціали абітурієнт

protected string idNum; //ідентифікаційний код абітурієнта

protected int coursePoints; //бали за підготовчі курс

protected double avgPoints; //бал атестат

protected ZNO ZNOResults;

public ZNO Sub1, Sub2, Sub3, Sub4;

public double compMark;

public string Name

{

get { return name; }

set

{

if (value.Length == 0)

{

ForegroundColor = ConsoleColor.Red;

WriteLine("Некоректне введення!!! Повторіть спробу");

ResetColor();

}

else

name = value;

}

}

public string IdNum

{

get { return idNum; }

set

{

if (value.Length == 0)

{

ForegroundColor = ConsoleColor.Red;

WriteLine("Некоректне введення!!! Повторіть спробу");

ResetColor();

}

else

idNum = value;

}

}

public int CoursePoints;

public double AvgPoints;

public double CompMark;

public Entrant()

{

Name = "Anonim Anonimus";

IdNum = "21584GK41584";

CoursePoints = 120;

AvgPoints = 8.3;

ZNOResults = new ZNO();

}

public Entrant(string name, string idNum, int coursePoints, double avgPoints, ZNO sub1, ZNO sub2, ZNO sub3, ZNO sub4, string subject, double points, double compMark)

{

Name = name;

IdNum = idNum;

CoursePoints = coursePoints;

AvgPoints = avgPoints;

Sub1 = new ZNO(sub1);

Sub2 = new ZNO(sub2);

Sub3 = new ZNO(sub3);

Sub4 = new ZNO(sub4);

ZNOResults = new ZNO(subject, points);

CompMark = compMark;

}

public Entrant(Entrant obj)

{

Name = obj.name;

IdNum = obj.idNum;

CoursePoints = obj.coursePoints;

AvgPoints = obj.avgPoints;

ZNOResults = new ZNO(obj.ZNOResults);

CompMark = obj.compMark;

}

static public double GetCompMark(Entrant x)

{

double result = x.CoursePoints \* 0.05 + x.AvgPoints \* 0.1 + x.ZNOResults.Points \* 0.25 + x.ZNOResults.Points \* 0.4 + x.ZNOResults.Points \* 0.2;

return result;

}

public string GetBestSubject(Entrant x)

{

string best;

if (x.Sub1.Points > x.Sub2.Points && x.Sub1.Points > x.Sub3.Points && x.Sub1.Points > x.Sub4.Points)

best = x.Sub1.Subject;

else

if (x.Sub2.Points > x.Sub3.Points && x.Sub2.Points > x.Sub4.Points)

best = x.Sub2.Subject;

else

if (x.Sub3.Points > x.Sub4.Points)

best = x.Sub3.Subject;

else

best = x.Sub4.Subject;

return best;

}

public string GetWorstSubject(Entrant x)

{

string worst;

if (x.Sub1.Points < x.Sub2.Points && x.Sub1.Points < x.Sub3.Points && x.Sub1.Points < x.Sub4.Points)

worst = x.Sub1.Subject;

else

if (x.Sub2.Points < x.Sub3.Points && x.Sub2.Points < x.Sub4.Points)

worst = x.Sub2.Subject;

else

if (x.Sub3.Points < x.Sub4.Points)

worst = x.Sub3.Subject;

else

worst = x.Sub4.Subject;

return worst;

}

}

}

ZNO

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SimpleClassLibrary

{

public class ZNO

{

protected string subject; //назва предмет

protected double points; //результат

public string Subject;

public double Points;

public ZNO()

{

}

public ZNO(string subject, double points)

{

Subject = subject;

Points = points;

}

public ZNO(ZNO obj)

{

Subject = obj.subject;

Points = obj.points;

}

}

}

Результат виконання програми:

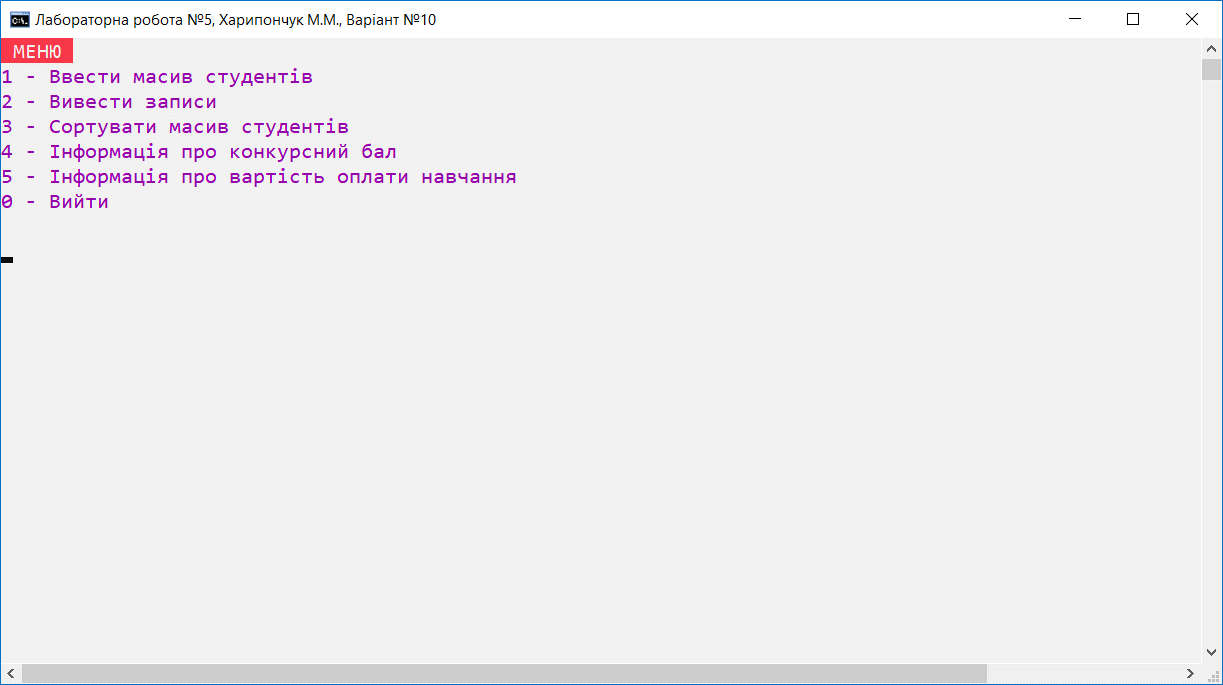


Рис. 1. Головне меню

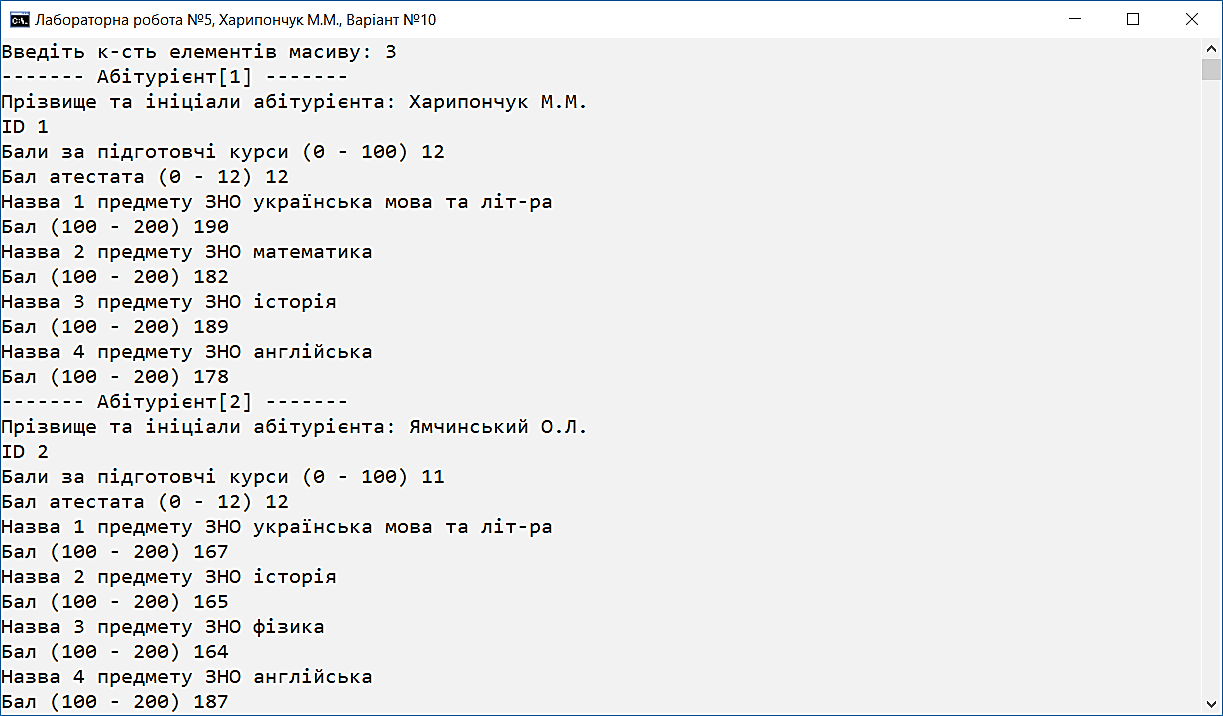


Рис. 2. Створення запису

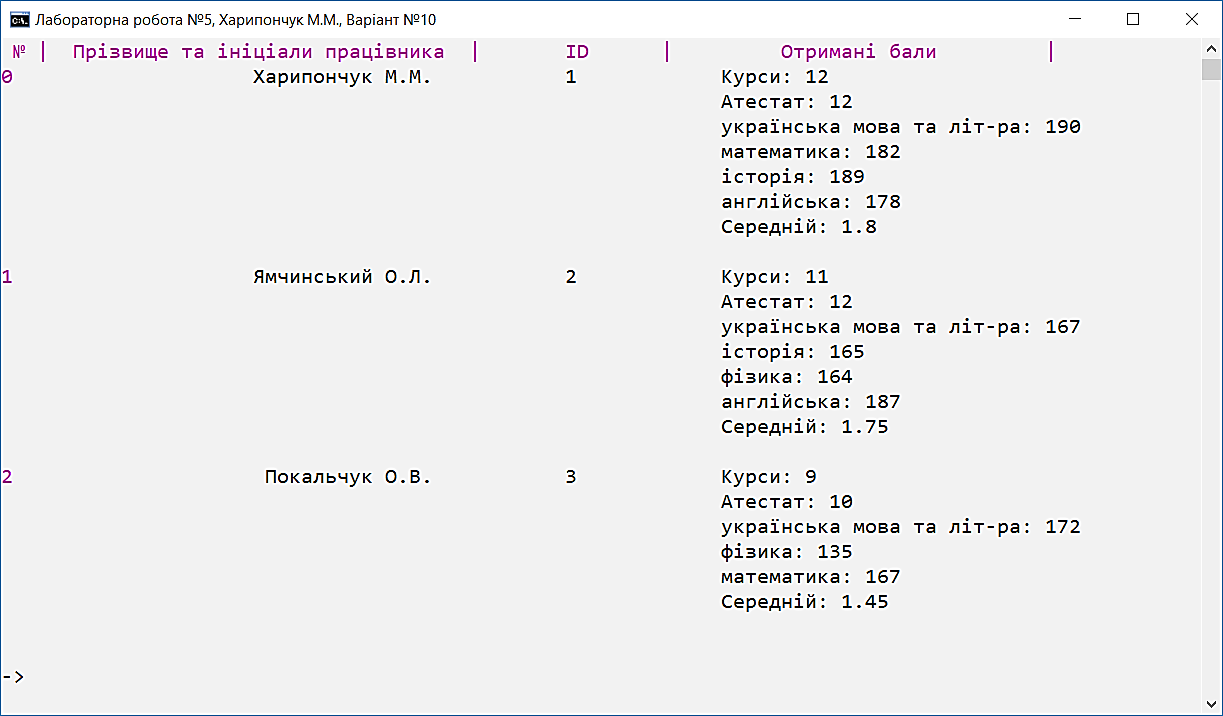


Рис. 3. Виведення записів

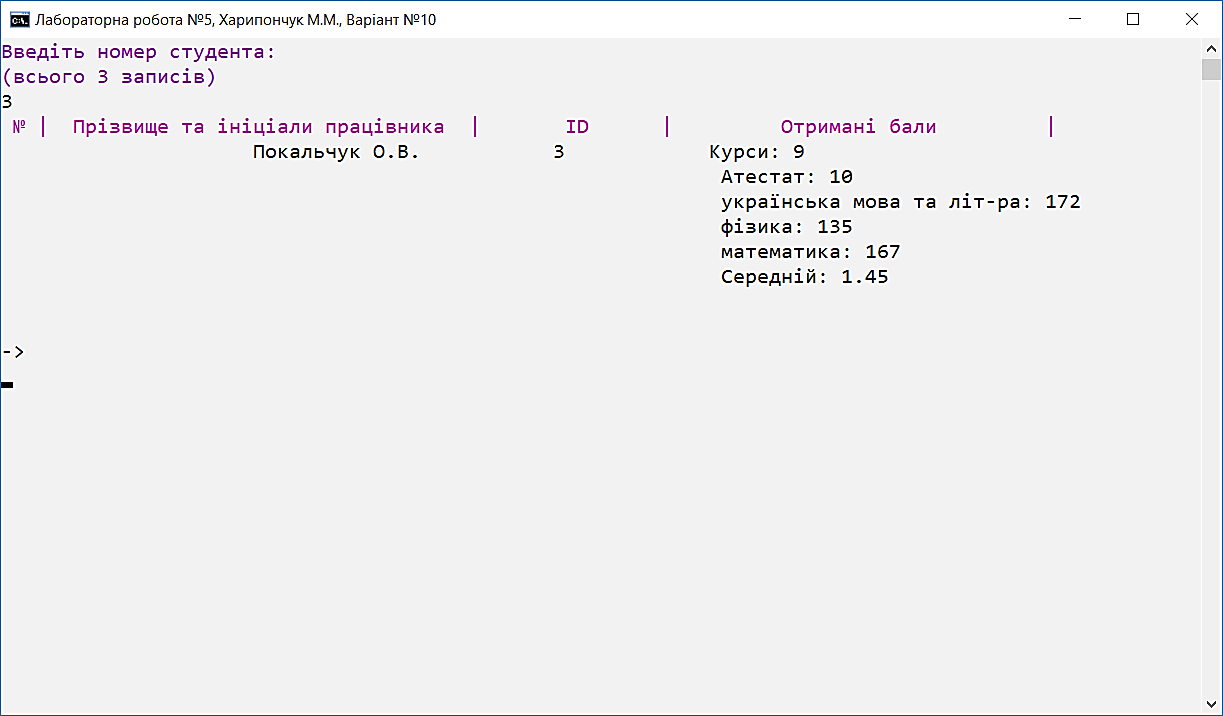


Рис. 4. Виведення одного запису

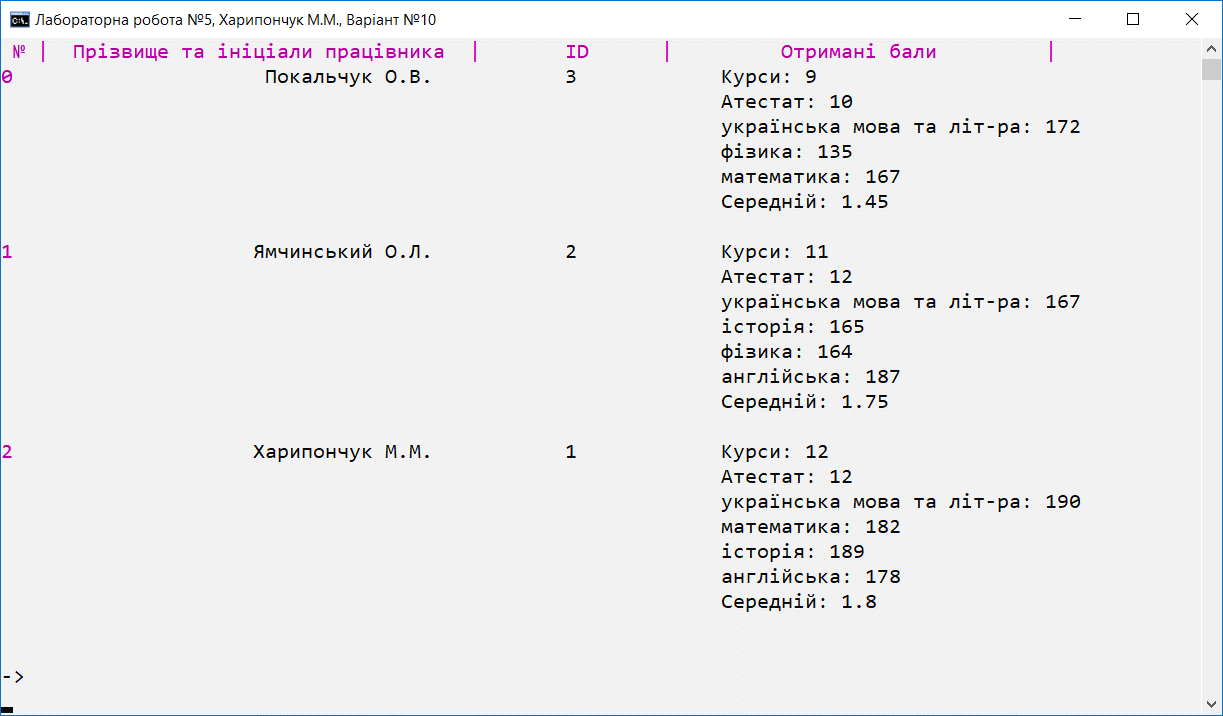


Рис. 5. Сортування за середнім балом

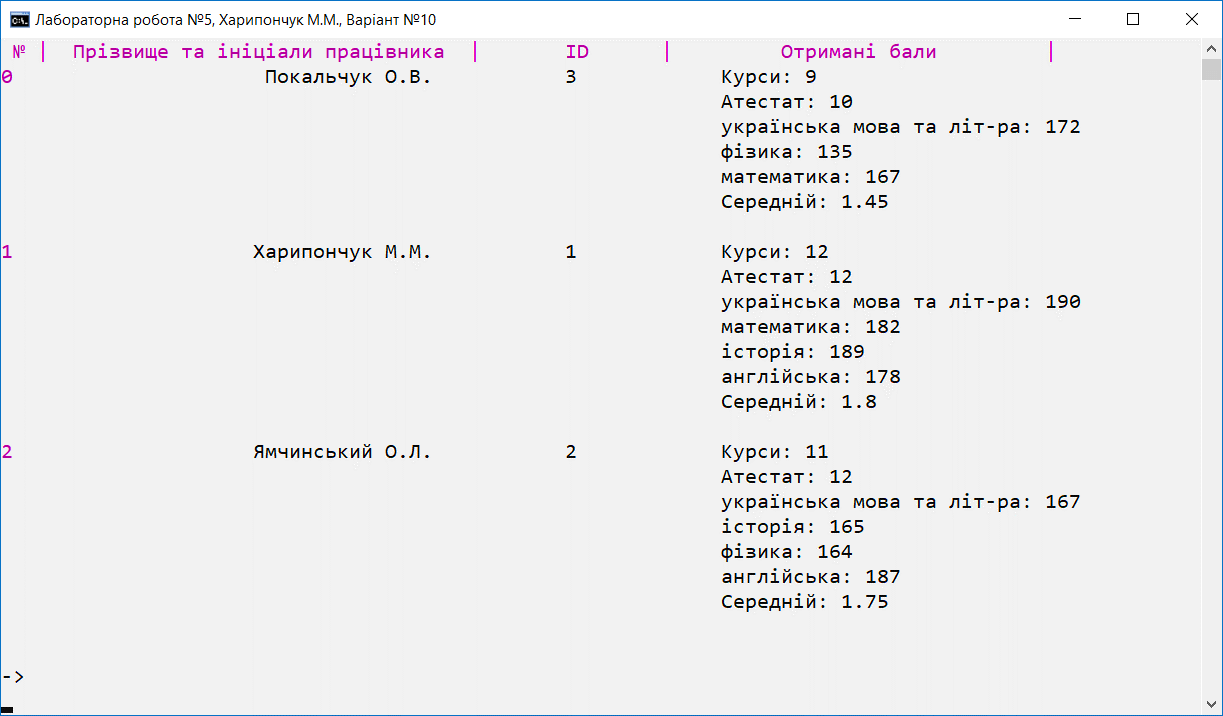


Рис. 6. Сортування за прізвищем

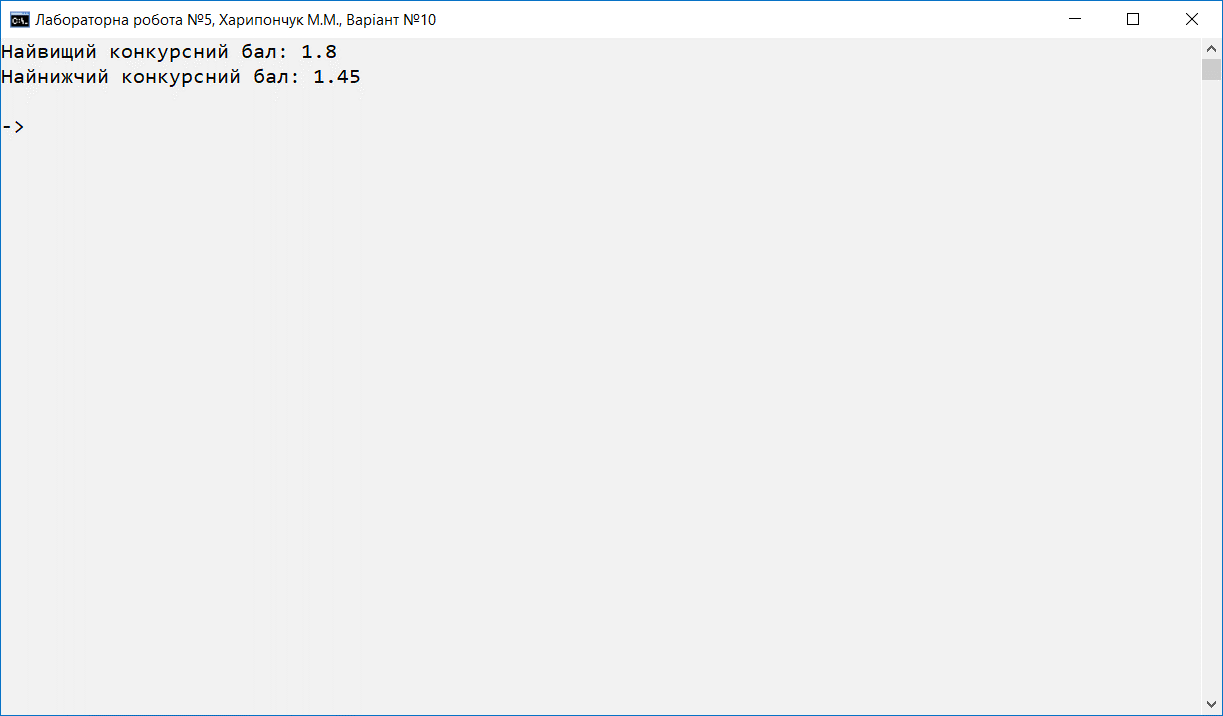
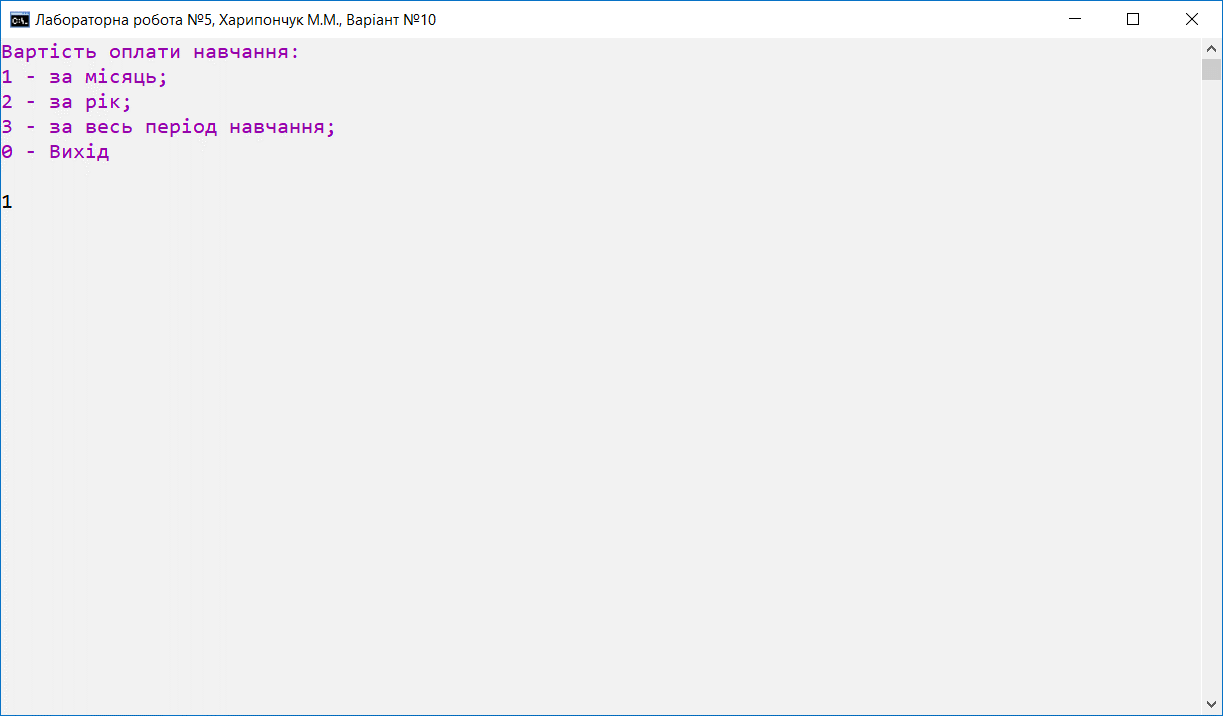


Рис. 7. Виведення інформації про конкурсний бал



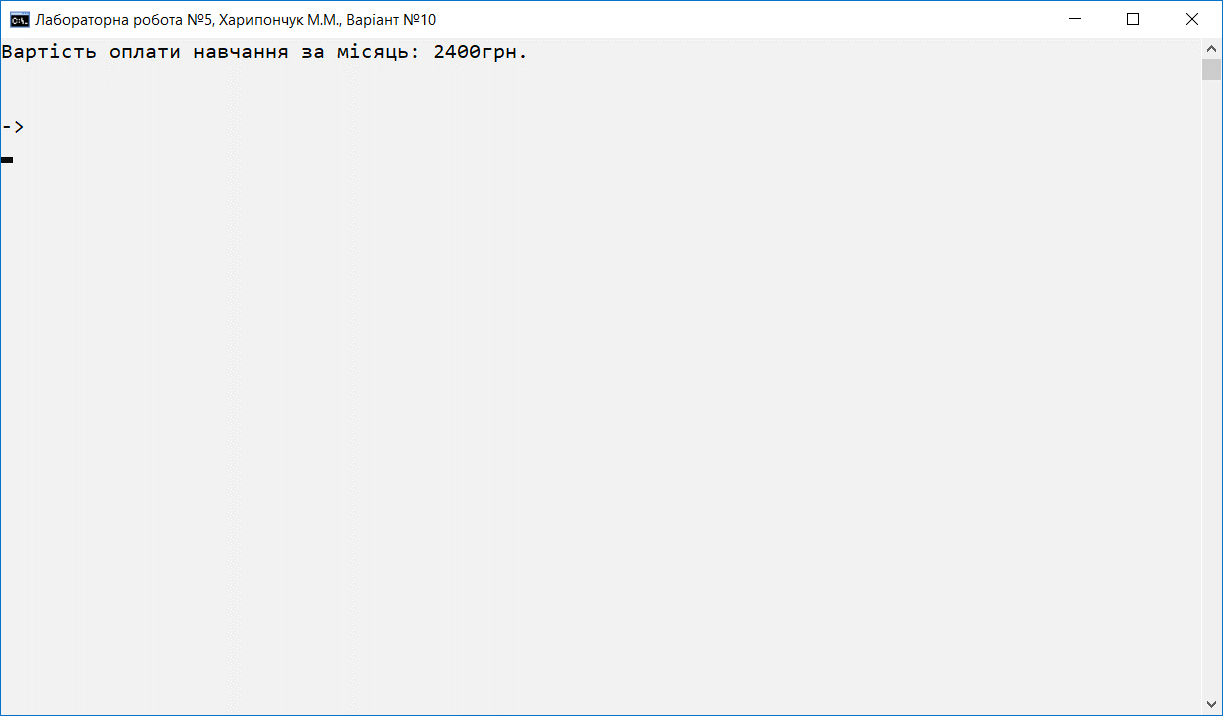


Рис. 8. Виведення інформації про вартість оплати навчання

***Висновки:*** в ході виконання лабораторної роботи було засвоєно принципи роботи проектування та оголошення класів; вивчено особливості реалізації трьох типів конструкторів (по замовчуванню, з параметрами та копію-вання) та методів у класах..