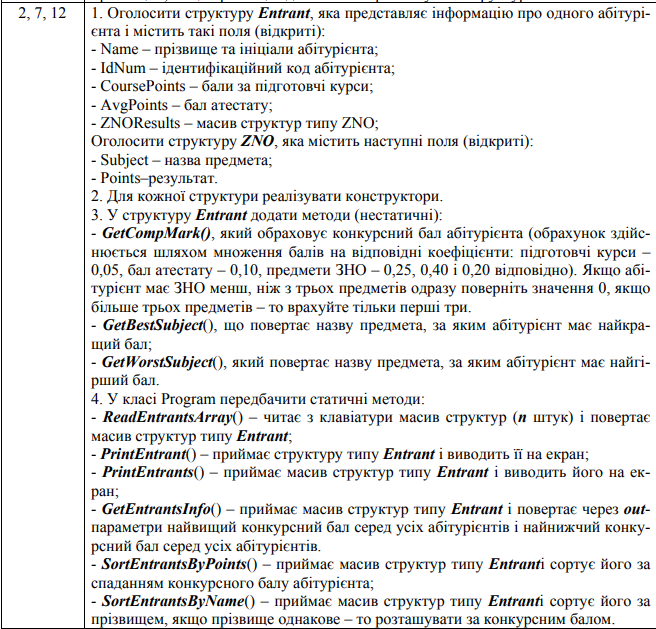
**ЛАБОРАТОРНА РОБОТА № 5**

**Варіант 2**

Структури. Масиви структур. Передача параметрів у методи. Модифікатори параметрів. Додаткові можливості класуConsole.

***Мета*** : засвоїти принципи роботи зі структурами та масивами структур; вивчити особливості передачі параметрів за допомогою модифікаторів ref, out та params; ознайомитись з додатковими можливостями класу Console.

**Хід роботи:**

****

**Завдання 1**:

struct Entrant

{

public string Name;

public int IdNum;

public int CoursePoints;

public int AvgPoints;

public ZNO[] ZNOResults = new ZNO[3];

public Entrant(string name, int idnum, int coursepoints, int avgpoints)

{

Name = name;

IdNum = idnum;

CoursePoints = coursepoints;

AvgPoints = avgpoints;

}

public double GetCompMark()

{

double bal = CoursePoints \* 0.05 + AvgPoints \* 0.10 + ZNOResults[0].Points \* 0.25 + ZNOResults[1].Points \* 0.4 + ZNOResults[0].Points \* 0.2;

return bal;

}

public string GetBestSubject()

{

int max = 0;

string s = null;

for(int i = 0; i < 3; i++)

{

if(ZNOResults[i].Points > max)

{

max = ZNOResults[i].Points;

s = ZNOResults[i].Subject;

}

}

return s;

}

public string GetWorstSubject()

{

int min = 99999;

string s = null;

for (int i = 0; i < 3; i++)

{

if (ZNOResults[i].Points < min)

{

min = ZNOResults[i].Points;

s = ZNOResults[i].Subject;

}

}

return s;

}

}

struct ZNO

{

public string Subject;

public int Points;

public ZNO(string subject, int points)

{

Subject = subject;

Points = points;

}

}

class Program

{

static void Main()

{

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

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

customCulture.NumberFormat.NumberDecimalSeparator = ".";

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

Console.OutputEncoding = System.Text.Encoding.Default;

Console.Title = "Лабораторна робота №5";

Console.SetWindowSize(100, 25);

Console.BackgroundColor = ConsoleColor.White;

Console.ForegroundColor = ConsoleColor.DarkBlue;

Console.Clear();

Console.WriteLine("Лабораторна робота №5");

Console.WriteLine("Виконав: Маньківський В.В., група ВТ-21-1");

Console.WriteLine("Варіант №2");

Console.WriteLine("Завдання 1");

Console.WriteLine("Кількість записів");

bool n0;

int fl;

do

{

n0 = true;

if (int.TryParse(Console.ReadLine(), out fl))

{

n0 = false;

}

else

{

Console.WriteLine("Введіть ще раз");

}

} while (n0);

Entrant[] ent = new Entrant[fl];

ReadEntrantsArray(fl, ent);

for(; ; ) {

Console.ReadKey();

Console.Clear();

Console.WriteLine("Виберіть дію:\n1)Вивести всі дані\n2)Вивести одного студента\n3)Конкурсний бал\n4)Найкращий бал\n5)Найгірший бал\n6)Найкащий/найгірший конкурсний бал\n7)Сортування конкурсний бал\n8)Сортування за прізвищем");

int t, maxi = 0, mini = 0;

double max, min;

do

{

n0 = true;

if (int.TryParse(Console.ReadLine(), out t) && t > 0 && t < 9)

{

n0 = false;

}

else

{

Console.WriteLine("Введіть ще раз");

}

} while (n0);

switch (t) {

case 1:

PrintEntrants(fl, ent);

break;

case 2:

PrintEntrant(fl, ent);

break;

case 3:

for (int i = 0; i < fl; i++)

{

Console.WriteLine($"Студент №{i+1} -> {Math.Round(ent[i].GetCompMark(), 2)}");

}

break;

case 4:

for (int i = 0; i < fl; i++)

{

Console.WriteLine($"Студент №{i + 1} -> {ent[i].GetBestSubject()}");

}

break;

case 5:

for (int i = 0; i < fl; i++)

{

Console.WriteLine($"Студент №{i + 1} -> {ent[i].GetWorstSubject()}");

}

break;

case 6:

GetEntrantsInfo(fl, ent, out max, out min, ref maxi, ref mini);

Console.WriteLine($"Студент №{maxi + 1} -> {max}");

Console.WriteLine($"Студент №{mini + 1} -> {min}");

break;

case 7:

SortEntrantsByPoints(fl, ent);

PrintEntrants(fl, ent);

break;

case 8:

SortEntrantsByName(fl, ent);

PrintEntrants(fl, ent);

break;

default:

Console.WriteLine($"Неправильний ввід");

break;

}

}

}

static void SortEntrantsByName(int fl, Entrant[] ent)

{

Entrant temp;

for (int j = 0; j < fl; j++)

{

for (int i = 0; i < fl - 1; i++)

{

if (needToReOrder(ent[i].Name, ent[i + 1].Name))

{

temp = ent[i];

ent[i] = ent[i + 1];

ent[i + 1] = temp;

}

if(ent[i].Name == ent[i + 1].Name)

{

if (ent[i].GetCompMark() < ent[i + 1].GetCompMark())

{

temp = ent[i];

ent[i] = ent[i + 1];

ent[i + 1] = temp;

}

}

}

}

}

protected static bool needToReOrder(string s1, string s2)

{

for (int i = 0; i < (s1.Length > s2.Length ? s2.Length : s1.Length); i++)

{

if (s1.ToCharArray()[i] < s2.ToCharArray()[i]) return false;

if (s1.ToCharArray()[i] > s2.ToCharArray()[i]) return true;

}

return false;

}

static void SortEntrantsByPoints(int fl, Entrant[] ent)

{

Entrant temp;

for(int j = 0; j < fl; j++)

{

for(int i = 0; i < fl - 1; i++) {

if(ent[i].GetCompMark() < ent[i + 1].GetCompMark())

{

temp = ent[i];

ent[i] = ent[i + 1];

ent[i + 1] = temp;

}

}

}

}

static void GetEntrantsInfo(int fl, Entrant[] ent, out double max, out double min, ref int maxi, ref int mini)

{

max = 0;

min = 99999;

for(int i = 0; i < fl; i++)

{

if(ent[i].GetCompMark() > max)

{

max = Math.Round(ent[i].GetCompMark(), 2);

maxi = i;

}

if(ent[i].GetCompMark() < min)

{

min = Math.Round(ent[i].GetCompMark(), 2);

mini = i;

}

}

}

static void ReadEntrantsArray(int fl, Entrant[] ent)

{

bool n0;

for (int i = 0; i < fl; i++)

{

Console.WriteLine($"--- Данні №{i + 1} ---");

Console.WriteLine("Ім'я");

string n = Console.ReadLine();

Console.WriteLine("Номер");

int d;

do

{

n0 = true;

if (int.TryParse(Console.ReadLine(), out d))

{

n0 = false;

}

else

{

Console.WriteLine("Введіть ще раз");

}

} while (n0);

Console.WriteLine("Курси");

int c;

do

{

n0 = true;

if (int.TryParse(Console.ReadLine(), out c))

{

n0 = false;

}

else

{

Console.WriteLine("Введіть ще раз");

}

} while (n0);

Console.WriteLine("Бал атестату");

int a;

do

{

n0 = true;

if (int.TryParse(Console.ReadLine(), out a))

{

n0 = false;

}

else

{

Console.WriteLine("Введіть ще раз");

}

} while (n0);

ent[i] = new Entrant(n, d, c, a);

for (int j = 0; j < 3; j++)

{

Console.WriteLine($"Предмет №{j+1}");

string s = Console.ReadLine();

Console.WriteLine($"Бал");

int p;

do

{

n0 = true;

if (int.TryParse(Console.ReadLine(), out p))

{

n0 = false;

}

else

{

Console.WriteLine("Введіть ще раз");

}

} while (n0);

ent[i].ZNOResults[j] = new ZNO(s, p);

}

}

}

static void PrintEntrants(int fl, Entrant[] ent)

{

for (int i = 0; i < fl; i++)

{

Console.Write($"\nПрізвище -> {ent[i].Name}\nНомер -> {ent[i].IdNum}\nКурс -> {ent[i].CoursePoints}\nАтестат -> {ent[i].AvgPoints}\n");

for (int j = 0; j < 3; j++)

{

Console.Write($"{ent[i].ZNOResults[j].Subject} -> {ent[i].ZNOResults[j].Points}\n");

}

Console.WriteLine();

}

}

static void PrintEntrant(int fl, Entrant[] ent)

{

bool n0;

int i;

Console.WriteLine("Введіть номер запису");

do

{

n0 = true;

if (int.TryParse(Console.ReadLine(), out i) && i <= fl && i > 0)

{

n0 = false;

}

else

{

Console.WriteLine("Введіть ще раз");

}

} while (n0);

Console.Write($"\nПрізвище -> {ent[i-1].Name}\nНомер -> {ent[i-1].IdNum}\nКурс -> {ent[i-1].CoursePoints}\nАтестат -> {ent[i-1].AvgPoints}\n");

for (int j = 0; j < 3; j++)

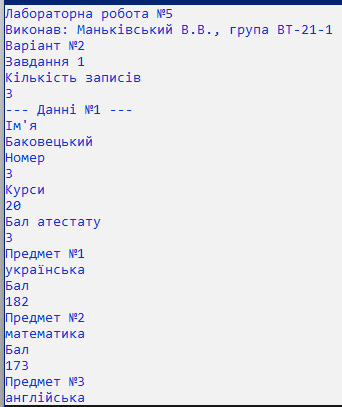
{

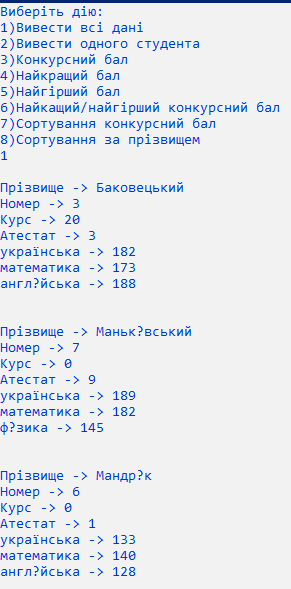
Console.Write($"{ent[i-1].ZNOResults[j].Subject} -> {ent[i-1].ZNOResults[j].Points}\n");

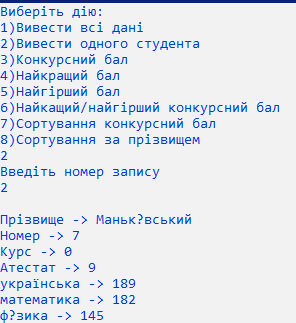
}

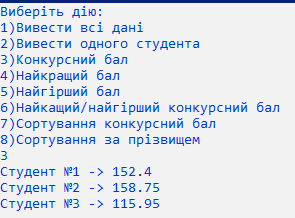
}

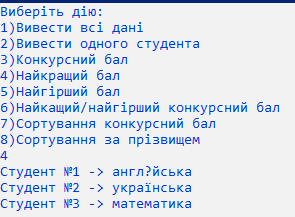
}

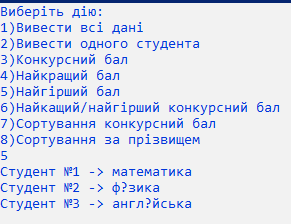
******

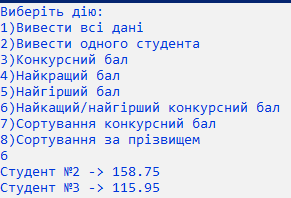
******

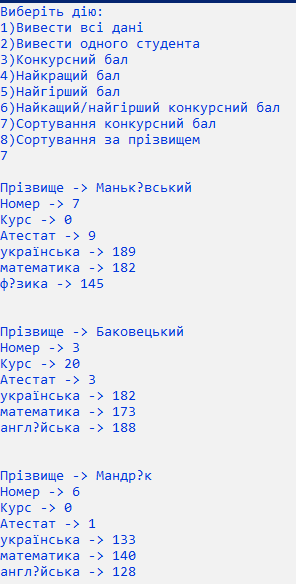
******

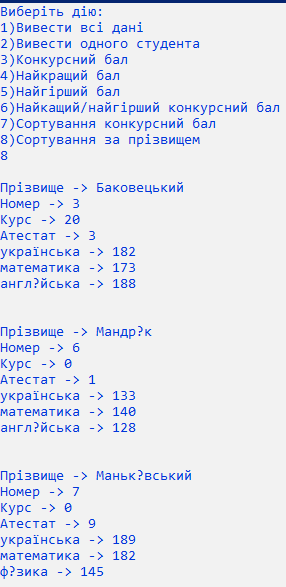
******

******

******

******

******

******

***Висновки:*** я засвоїв принципи роботи зі структурами та масивами структур; вивчив особливості передачі параметрів за допомогою модифікаторів ref, out та params; ознайомився з додатковими можливостями класу Console.