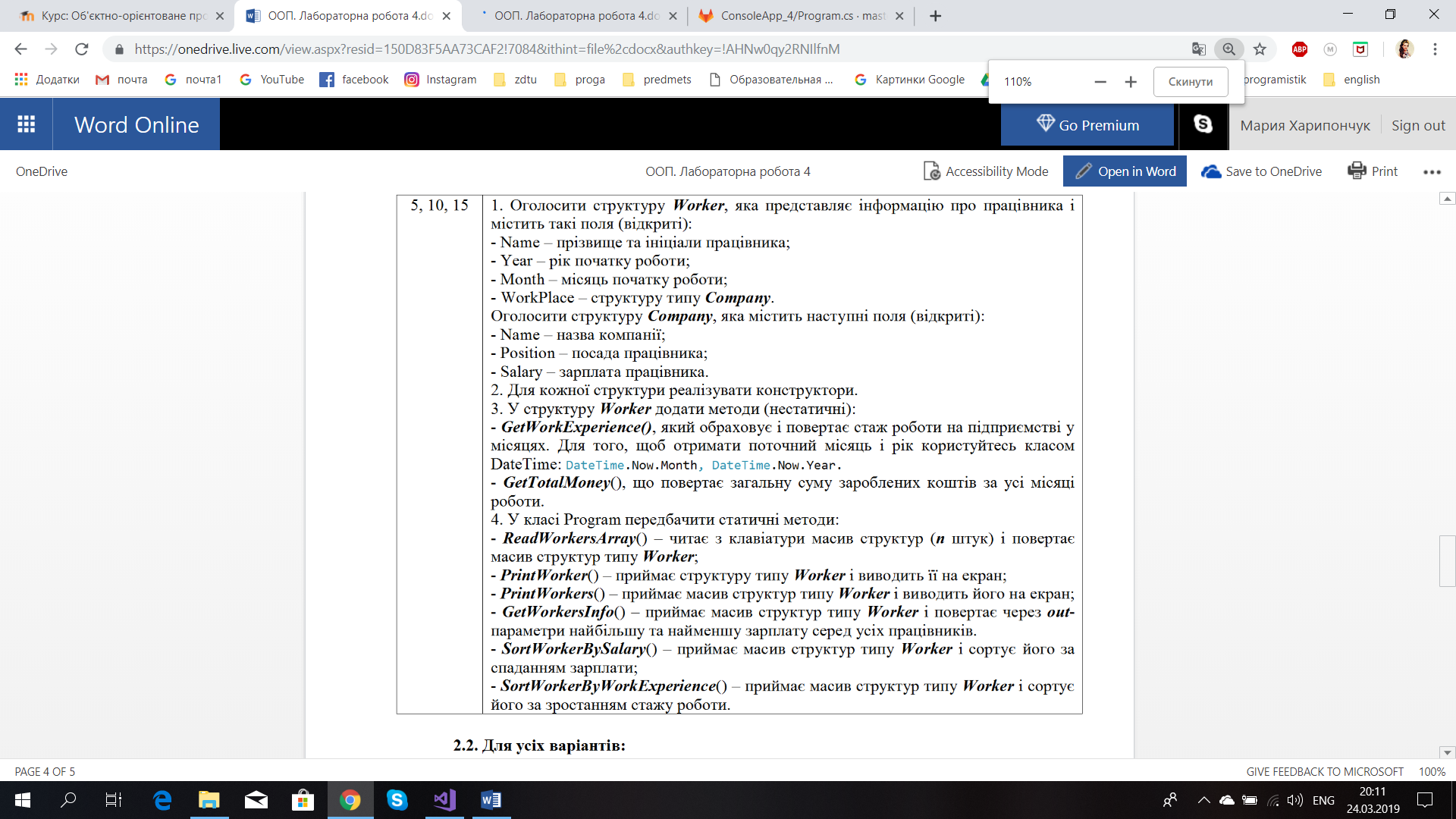
**ЛАБОРАТОРНА РОБОТА № 4**

**Структури. Масиви структур. Передача параметрів у методи. Модифікатори параметрів. Додаткові можливості класу Console**

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

**Хід роботи:**

**Завдання:**



Лістининг програми:

using System;

using System.Text;

using static System.Console;

namespace oop\_lab4

{

class Program

{

struct Company

{

public string NameCompany;

public string Position;

public double Salary;

public Company(string nameCompany, string position, double salary)

{

NameCompany = nameCompany;

Position = position;

Salary = salary;

}

}

struct Worker

{

public string Name;

public int Year;

public int Month;

public Company WorkPlace;

public Worker(string name, string nameCompany, string position, double salary, int year, int month)

{

Name = name;

Year = year;

Month = month;

WorkPlace = new Company (nameCompany, position, salary);

}

public int GetWorkExperience()

{

int exp;

int year = DateTime.Now.Year - Year;

int month = DateTime.Now.Month - Month;

return exp = (year \* 365) + month;

}

public double GetTotalMoney(Company n)

{

double money = n.Salary\* GetWorkExperience();

return money;

}

}

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 повторіть введення ще раз!!!");

ResetColor();

}

} while (!ok);

return x;

}

readonly static string[] FieldNames =

{

" Прізвище та ініціали працівника:",

" Рік початку роботи: ",

" Місяць початку роботи: ",

" Назва компанії: ",

" Посада працівника: ",

" Зарплата працівника: ",

};

static Worker[] ReadWorkersArray()

{

Write("Введіть к-сть елементів масиву: ");

int n = GetValue();

string name, nameCompany, position;

double salary;

int year, month;

bool f;

Worker[] arr = new Worker[n];

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

{

WriteLine($"------- Працівник[{i + 1}] -------\n");

WriteLine($"{FieldNames[0]}");

name = ReadLine();

WriteLine($"{FieldNames[1]}");

do

{

f = int.TryParse(ReadLine(), out year);

if (year <= 0 || year > DateTime.Now.Year)

{

ForegroundColor = ConsoleColor.Red;

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

ResetColor();

f = false;

}

}

while (!f);

WriteLine($"{FieldNames[2]}");

do

{

f = int.TryParse(ReadLine(), out month);

if (month <= 0 || month > 12)

{

ForegroundColor = ConsoleColor.Red;

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

ResetColor();

f = false;

}

}

while (!f);

WriteLine($"{FieldNames[3]}");

nameCompany = ReadLine();

WriteLine($"{FieldNames[4]}");

position = ReadLine();

WriteLine($"{FieldNames[5]}");

do

{

f = double.TryParse(ReadLine(), out salary);

if (salary < 0)

f = false;

if(f == false)

{

ForegroundColor = ConsoleColor.Red;

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

ResetColor();

}

}

while (!f);

arr[i] = new Worker(name, nameCompany, position, salary, year, month);

}

return arr;

}

static void PrintWorker(Worker elem)

{

WriteLine($"{elem.Name,34}" +

$"{elem.Month, 19}/{elem.Year, 4}" +

$"{FieldNames[3], 30}{elem.WorkPlace.NameCompany}\n {FieldNames[4], 92}{elem.WorkPlace.Position}\n {FieldNames[5], 94}{elem.WorkPlace.Salary}\n");

}

static void PrintWorkers(Worker[] arr)

{

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

{

ForegroundColor = ConsoleColor.Magenta;

Write($" {i + 1}");

ResetColor();

PrintWorker(arr[i]);

}

}

static void GetWorkersInfo(Worker[] arr, out int iMaxOut, out int iMinOut)

{

double min = 0, max = 0;

int iMin = 0, iMax = 0;

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

{

double tmp = arr[i].WorkPlace.Salary;

if (tmp > max)

{

max = tmp;

iMax = i;

}

else if (tmp < min)

{

min = tmp;

iMin = i;

}

}

iMaxOut = iMax;

iMinOut = iMin;

}

static int CompBySalary(Worker a, Worker b)

{

double salaryA = a.WorkPlace.Salary;

double salaryB = b.WorkPlace.Salary;

if (salaryA > salaryB)

return 1;

else if (salaryA < salaryB)

return -1;

else

return 0;

}

static int CompByWorkExperience(Worker a, Worker b)

{

double expA = a.GetWorkExperience(), expB = b.GetWorkExperience();

if (expA > expB)

return 1;

else if (expA < expB)

return -1;

else

return 0;

}

static Worker[] SortWorkerBySalary(Worker[] arr)

{

Array.Sort(arr, CompBySalary);

return arr;

}

static Worker[] SortWorkerByWorkExperience(Worker[] arr)

{

Array.Sort(arr, CompByWorkExperience);

return arr;

}

static void PrintTable()

{

ForegroundColor = ConsoleColor.Magenta;

WriteLine(" № | Прізвище та ініціали працівника | Місяць/рік початку роботи | Працівник |");

ResetColor();

}

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

static void ShowMenu()

{

BackgroundColor = ConsoleColor.Red;

ForegroundColor = ConsoleColor.White;

WriteLine(" МЕНЮ ");

ResetColor();

ForegroundColor = ConsoleColor.DarkMagenta;

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

"2 - Вивести записи\n" +

"3 - Сортувати масив працівників\n" +

"4 - Інформація про зарплату працівників\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 №4, Харипончук М.М., Варіант №10";

SetWindowSize(110, 25);

BackgroundColor = ConsoleColor.White;

ForegroundColor = ConsoleColor.Black;

Worker[] arr = null;

int ch;

do

{

WriteLine("\n->");

ReadKey();

Clear();

ShowMenu();

ch = GetValue();

switch (ch)

{

case 1:

Clear();

arr = ReadWorkersArray();

break;

case 2:

if (arr == null)

{

ForegroundColor = ConsoleColor.Red;

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

ResetColor();

break;

}

ForegroundColor = ConsoleColor.DarkMagenta;

WriteLine("1 - Усі записи");

WriteLine("2 - Один запис");

WriteLine("0 - Вихід");

ResetColor();

int check = GetValue();

switch (check)

{

case 1:

Clear();

PrintTable();

PrintWorkers(arr);

break;

case 2:

Clear();

ForegroundColor = ConsoleColor.DarkMagenta;

WriteLine($"Введіть номер працівника:\n(всього {arr.Length} записів)");

ResetColor();

int num = GetValue() - 1;

if (num > arr.Length - 1 || num < 0)

{

ForegroundColor = ConsoleColor.Red;

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

ResetColor();

break;

}

else

{

PrintTable();

PrintWorker(arr[num]);

}

break;

case 0:

break;

default:

ForegroundColor = ConsoleColor.Red;

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

ResetColor();

break;

}

break;

case 3:

if (arr == null)

{

ForegroundColor = ConsoleColor.Red;

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

ResetColor();

break;

}

ForegroundColor = ConsoleColor.DarkMagenta;

WriteLine("1 - Сортувати за зарплатою;\n" +

"2 - Сортувати за стажем роботи;\n" +

"0 - Вийти\n");

ResetColor();

int number = GetValue();

switch (number)

{

case 1:

SortWorkerBySalary(arr);

Clear();

PrintTable();

PrintWorkers(arr);

break;

case 2:

SortWorkerByWorkExperience(arr);

Clear();

PrintTable();

PrintWorkers(arr);

break;

case 0:

break;

default:

ForegroundColor = ConsoleColor.Red;

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

ResetColor();

break;

}

break;

case 4:

if (arr == null)

{

ForegroundColor = ConsoleColor.Red;

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

ResetColor();

break;

}

int iMin, iMax;

GetWorkersInfo(arr, out iMax, out iMin);

Clear();

WriteLine($"Найменша зарплата: {arr[iMin].WorkPlace.Salary}");

WriteLine($"Найбільша зарплата: {arr[iMax].WorkPlace.Salary}");

break;

case 0:

break;

default:

ForegroundColor = ConsoleColor.Red;

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

ResetColor();

break;

}

} while (ch != 0);

}

}

}

Результат виконання програми:

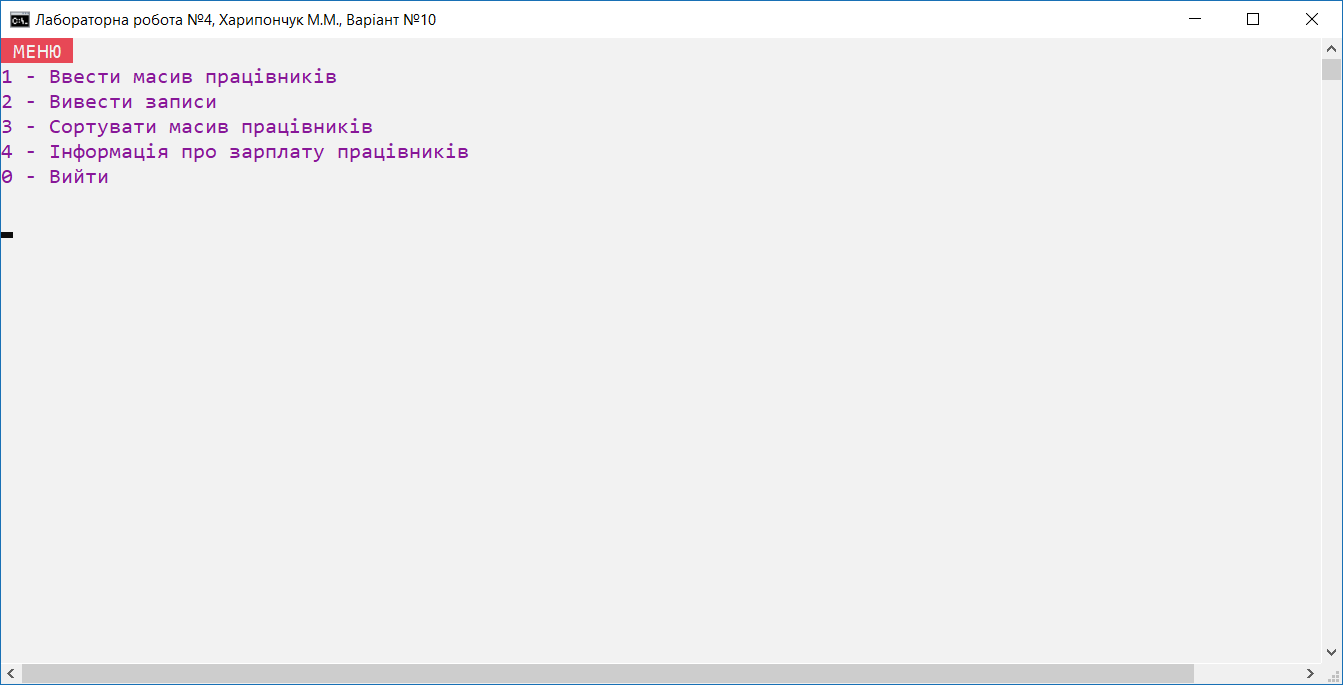


Рис. 1. Головне меню

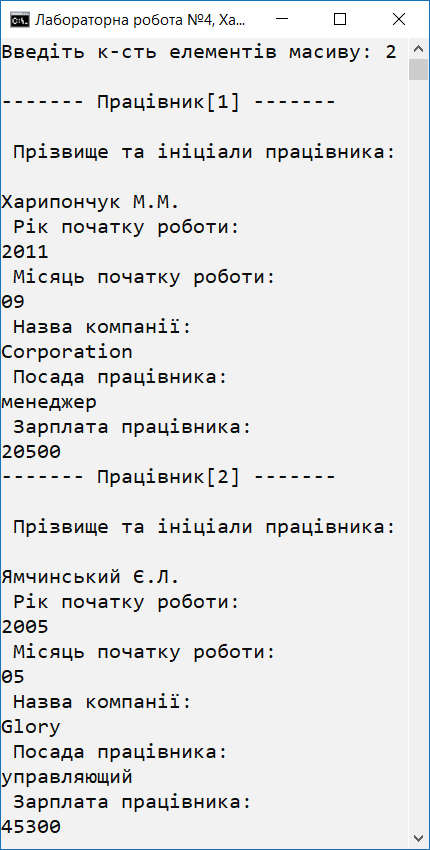


Рис. 2. Створення запису

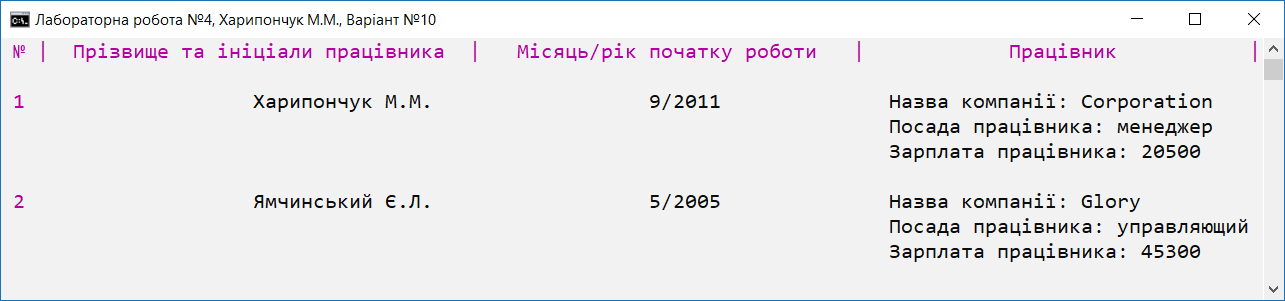


Рис. 3. Виведення записів

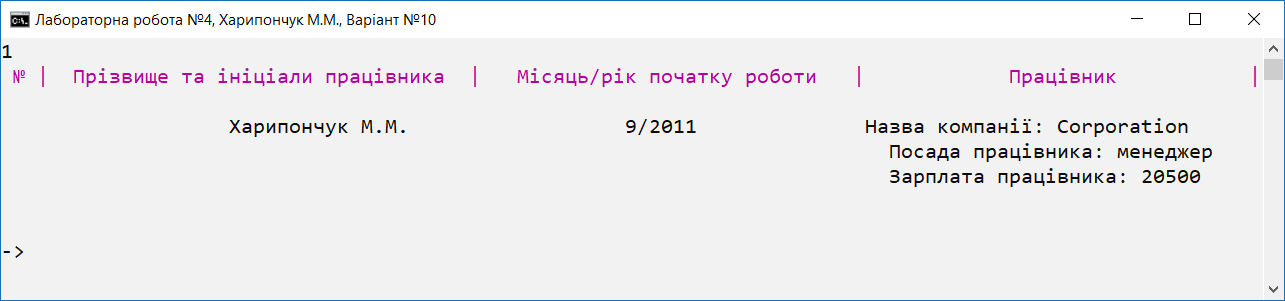


Рис. 4. Виведення одного запису

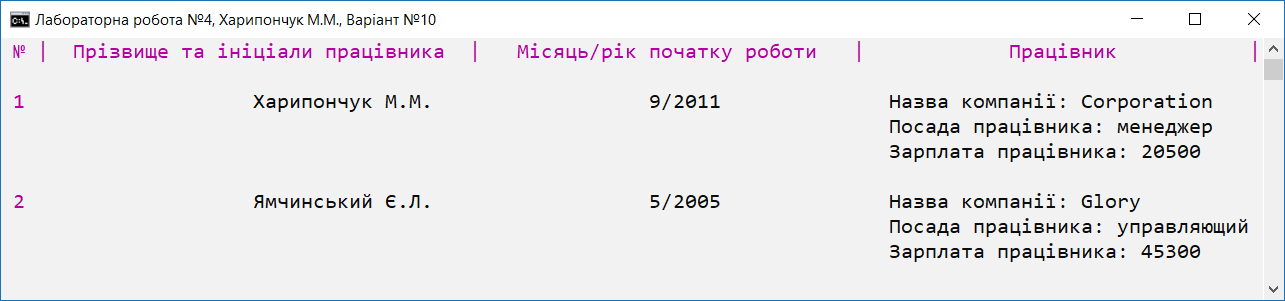


Рис. 5. Сортування за зарплатою

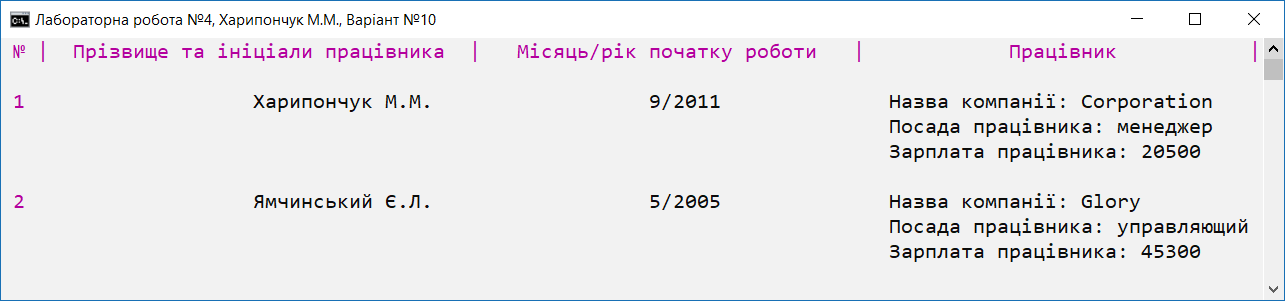


Рис. 6. Сортування за стажем роботи

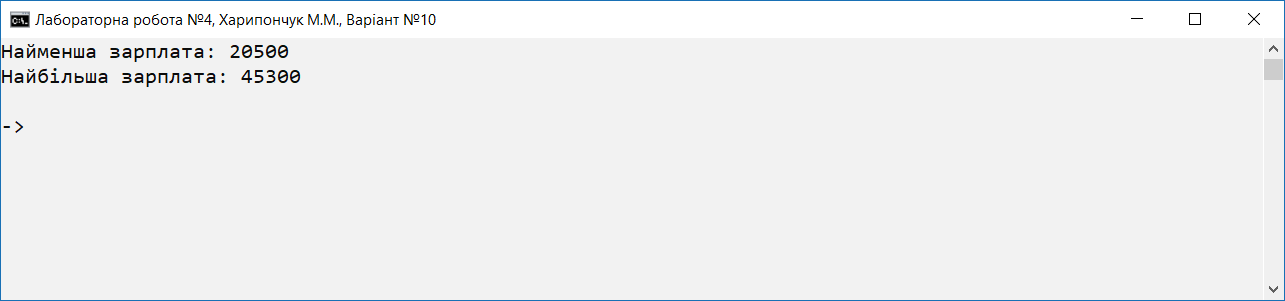


Рис. 7. Виведення інформації про зарплату працівників

***Висновки:*** в ході виконання лабораторної роботи було засвоєно принципи роботи зі структурами та масивами струк-тур; вивчено особливості передачі параметрів за допомогою модифікаторів ref, out та params; ознайомлено з додатковими можливостями класу Console.