**Міністерство освіти і науки України**

**УДУ імені М.П.Драгоманова**

Факультет інформатики

*Кафедра комп’ютерної інженерії та освітніх вимірювань*

ЛАБОРАТОРНА РОБОТА № 5

з курсу

**«Робота з даними на платформі .NET»**

Студент:

Клімов Антон Валентинович

Група 42 ІПЗ

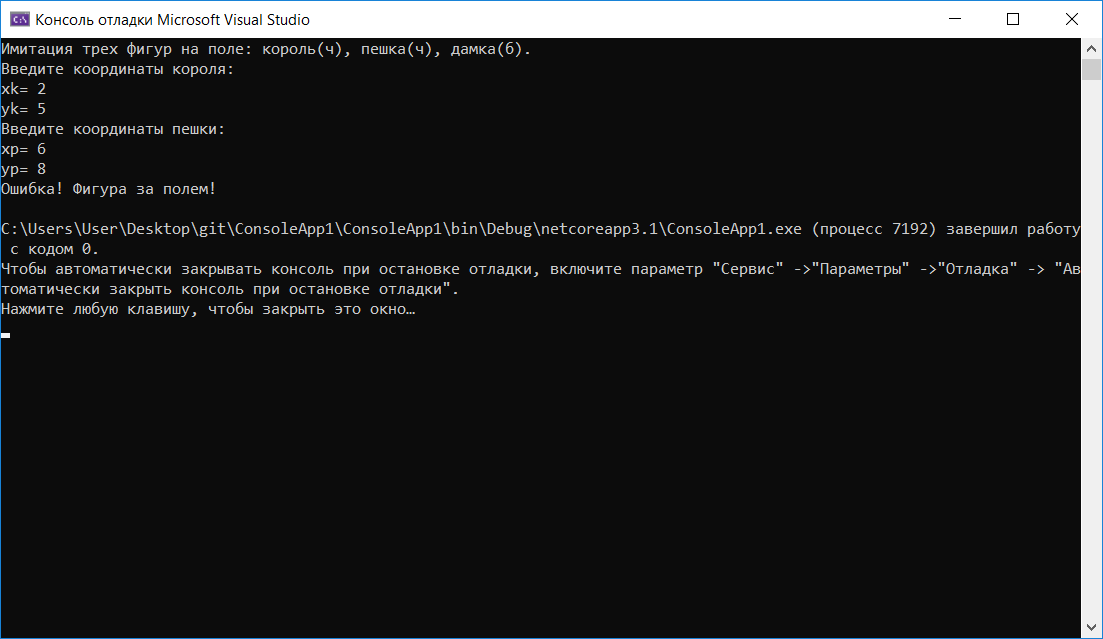
Факультет математики, інформатики та фізики

Викладач: Сулима Д.О.

Київ 2023

1. Ознайомитися з теоретичною частиною.  
2. Обрати тему з переліку запропонованих або визначити власну;

Гра «Шахи»



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

int xk, yk, xp, yp, xf, yf;

Console.WriteLine("Імітація трьох фігур на полі: король(ч), пішак(ч), дамка(б).");

// Введення короля на поле

Console.WriteLine("Введіть координати короля:");

Console.Write("хk= ");

xk = Convert.ToInt32(Console.ReadLine());// введення і ініціалізація хk

Console.Write("yk= ");

yk = Convert.ToInt32(Console.ReadLine());// введення і ініціалізація yk

if (xk <= 0 || xk >= 9 || yk <= 0 || yk >= 9)

{

Console.WriteLine("Помилка! Фігура за полем!");

return;

}

// Введення пішаків на поле

Console.WriteLine("Введіть координати пішака:");

Console.Write("хp= ");

xp = Convert.ToInt32(Console.ReadLine());// введення та ініціалізація хp

Console.Write("yp= ");

yp = Convert.ToInt32(Console.ReadLine());// введення та ініціалізація yp

if (xp <= 0 || xp >= 8 || yp <= 0 || yp >= 8) // пішак за полем або на недозволеному місці?

{

Console.WriteLine("Помилка! Фігура за полем!");

return;

}

if (yp == 1) // пішака стала дамкою дійшовши до кінця дошки?

{

Console.WriteLine("Помилка! Пішак стала дамкою!");

return;

}

if (xp == xk && yp == yk) // пішак на королі?

{

Console.WriteLine("Помилка! Фігура на фігурі!");

return;

}

// Введення дамки на поле

Console.WriteLine("Введіть координати дамки:");

Console.Write("хf= ");

xf = Convert.ToInt32(Console.ReadLine());// введення та ініціалізація хf

Console.Write("yf= ");

yf = Convert.ToInt32(Console.ReadLine());// введення та ініціалізація yf

if (xf <= 0 || xf >= 9 || yf <= 0 || yf >= 9) // дамка за полем?

{

Console.WriteLine("Помилка! Фігура за полем!");

return;

}

if ((xf == xk && yf == yk) || (xf == xp && yf == yp)) // дамка на іншій фігурі?

{

Console.WriteLine("Помилка! Фігура на фігурі!");

return;

}

Console.WriteLine("Фігури розставлені.");

Console.WriteLine("Король [{0}, {1}], пішак [{2}, {3}], дамка [{4},{5}]", xk, yk, xp, yp, xf, yf);

// ------------------------- Перевірка бою та висновок-----------------------------

bool boy\_korol = false, boy\_peshka = false, peshka\_Zkorol = false, korol\_Zpeshka = false;

// Дамка під боєм короля?

if ((xk + 1 == xf || xk == xf || xk - 1 == xf) && (yk - 1 == yf || yk == yf || yk - 1 == yf))

{

Console.WriteLine("Дамка знаходиться під боєм короля");

}

// Дамка під боєм пішака?

if (yp - 1 == yf && (xp - 1 == xf || xp + 1 == xf))

{

Console.WriteLine("Дамка знаходиться під боєм пішака ");

}

// Король під боєм дамки?

if ((xf == xk) || (yf == yk)) // По вертикалі та горизонталі

{

if (((xk > xp && xk < xf) || (xk < xp && xk > xf)) ||

((yk > yp && yk < yf) || (yk < yp && yk > yf)))

{

korol\_Zpeshka = true; // Король під боєм дамки та захищає пішака

}

else

{

boy\_korol = true; // Король під боєм дамки без захисту пішака

}

}

if (Math.Abs(xk - xf) == Math.Abs(yk - yf)) // За діагоналями

{

if (Math.Abs(xk - xf) < Math.Abs(xp - xf))

{

korol\_Zpeshka = true;// Король під боєм дамки та захищає пішака

}

else

{

boy\_korol = true; // Король під боєм дамки без захисту пішака

}

}

// Пішак під боєм дамки?

if ((xf == xp) || (yf == yp)) // По вертикалі та горизонталі

{

if (((xp > xk && xp < xf) || (xp < xk && xp > xf)) ||

((yp > yk && yp < yf) || (yp < yk && yp > yf)))

{

peshka\_Zkorol = true; // Пішак під боєм дамки і захищає короля

}

else

{

boy\_peshka = true; // Пішак під боєм дамки без захисту короля

}

}

if (Math.Abs(xp - xf) == Math.Abs(yp - yf)) // За діагоналями

{

if (Math.Abs(xp - xf) < Math.Abs(xk - xf))

{

peshka\_Zkorol = true;// Пішак під боєм дамки і захищає короля

}

else

{

boy\_peshka = true;// Пішак під боєм дамки без захисту короля

}

}

// Виводить хто під чиїм боєм

if ((!korol\_Zpeshka) && (boy\_peshka)) { Console.WriteLine("Пішак знаходиться під боєм дамки "); }

if (peshka\_Zkorol) { Console.WriteLine("Пішак знаходиться під боєм дамки і захищає короля "); }

if ((!peshka\_Zkorol) && (boy\_korol)) { Console.WriteLine("Король знаходиться під боєм дамки "); }

if (korol\_Zpeshka) { Console.WriteLine("Король знаходиться під боєм дамки та захищає пішака"); }

if (!korol\_Zpeshka && !boy\_peshka && !peshka\_Zkorol && !boy\_korol)

{

Console.WriteLine("Фігури не перетинаються ");

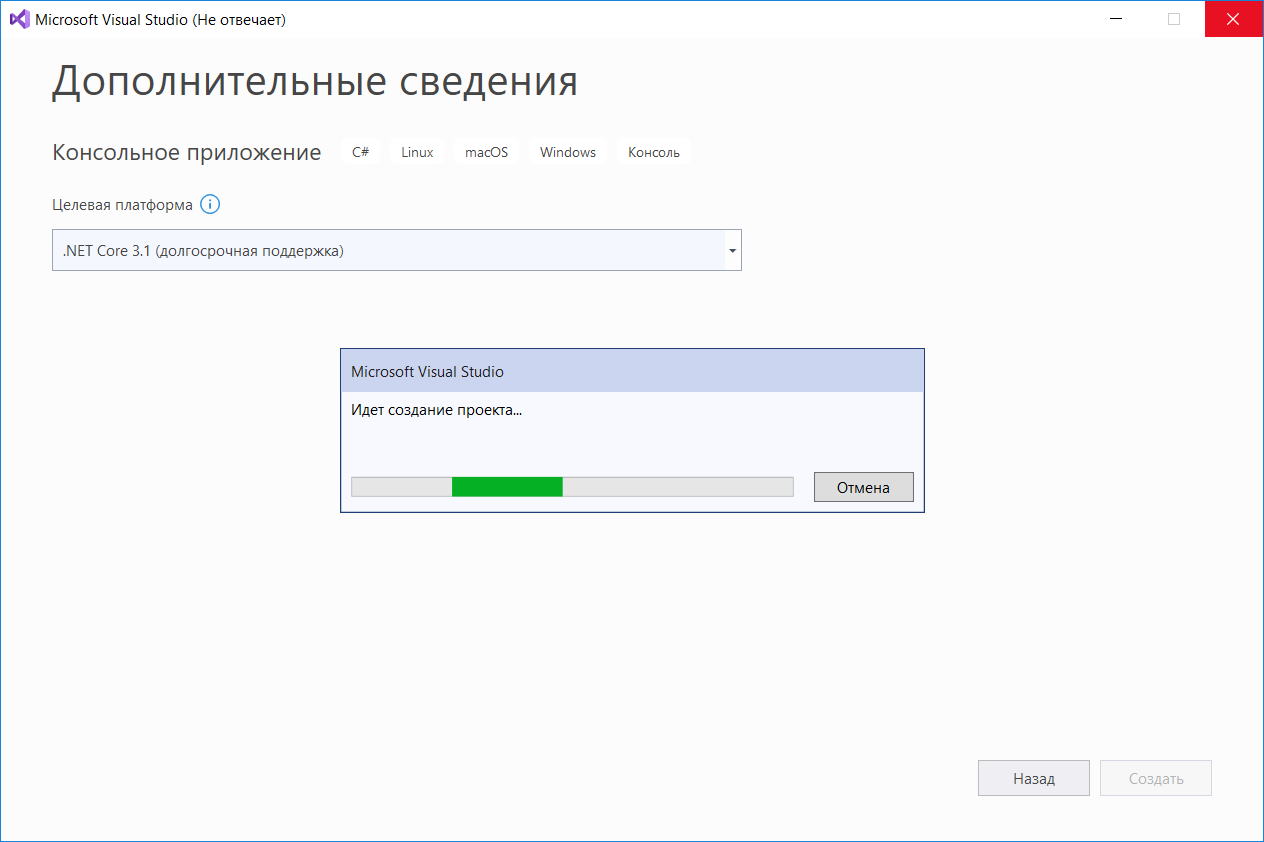
}

Console.Read();

}

}

}  
5. Створити новий консольний проект на мові програмування C#;



6. Визначити та створити необхідні колекції, які будуть використовуватись в обраній темі;

using System;

using System.Collections; //First in, first out.

public class SamplesQueue {

public static void Main() {

// Creates and initializes a new Queue.

Queue myQ = new Queue();

myQ.Enqueue("Welcome");

myQ.Enqueue("to");

myQ.Enqueue("Chess");

// Displays the properties and values of the Queue.

Console.WriteLine( "myQ" );

Console.WriteLine( "\tCount: xk, yk, xp, yp, xf, yf", myQ.Count );

Console.Write( "\tValues:" );

PrintValues( myQ );

}

public static void PrintValues( IEnumerable myCollection ) {

foreach ( Object obj in myCollection )

Console.Write( " {0}", obj );

Console.WriteLine();

}

}

7. Використати мінімум 3 методи розширення вибірки даних (на власний розсуд);

* Where(): Метод Where() фільтрує колекцію за певною умовою.

List<string> chess =

new List<string> { "король", "пішак", "дамка" };

IEnumerable<string> query = chess.Where(chess => chess.Length < 6);

foreach (string chess in query)

{

Console.WriteLine(chess);

}

* Select(): Метод Select() дозволяє вибрати певні поля або властивості об'єктів з колекції.

IEnumerable<int> chess =

Enumerable.Range(1, 3).Select(x => x \* x);

foreach (int xk in squares)

{

Console.WriteLine(xk);

}

Замість: xk = Convert.ToInt32(Console.ReadLine());

* OrderBy(): Метод OrderBy() сортує елементи колекції в порядку зростання.

class Chess

{

public string Name { get; set; }

public int XK { get; set; }

public int YK { get; set; }

}

public static void OrderByEx1()

{

Chess[] figures = { new Chess { Name="King", XK=8, YK=1},

new Chess { Name="Queen", XK=4, YK=6 },

new Chess { Name="Pawn", XK=1, YK=3 } };

IEnumerable<Chess> query = figures.OrderBy(figure => figure.XK, figure.YK);

foreach (Chess figure in query)

{

Console.WriteLine("{0} - {1}", figure.Name, figure.XK, figure.YK);

}

}

8. Використати мінімум 3 методи розширення зміни порядку даних (на власний розсуд);

* OrderByDescending - метод сортує елементи в порядку спадання.

var XK = new List<int> { 5, 2, 8, 1, 9 };

var sortedXK = XK.OrderByDescending(n => n);

* ThenBy - методи використовуються для сортування за декількома критеріями

string[] figures = { "king", "queen", "pawn" };

IEnumerable<string> query =

figures.OrderBy(figure => figure.Length).ThenBy(figure => figure);

foreach (string figure in query)

{

Console.WriteLine(figure);

}

* OrderBy(): Метод OrderBy() сортує елементи колекції в порядку зростання.

class Chess

{

public string Name { get; set; }

public int XK { get; set; }

public int YK { get; set; }

}

public static void OrderByEx1()

{

Chess[] figures = { new Chess { Name="King", XK=8, YK=1},

new Chess { Name="Queen", XK=4, YK=6 },

new Chess { Name="Pawn", XK=1, YK=3 } };

IEnumerable<Chess> query = figures.OrderBy(figure => figure.XK, figure.YK);

foreach (Chess figure in query)

{

Console.WriteLine("{0} - {1}", figure.Name, figure.XK, figure.YK);

}

}

9. Використати мінімум 2 методи розширення вибірки даних (на власний розсуд);

* Distinct() - цей метод повертає унікальні елементи з колекції.

List<int> XKs = new List<int> { 3, 1, 1, 6, 2, 4, 5, 5 };

IEnumerable<int> distinctAges = XKs.Distinct();

Console.WriteLine("Distinct XKs:");

foreach (int XK in distinctXK)

{

Console.WriteLine(XK);

}

* Count() - повертає кількість елементів у вибірці.

string[] figures = { "king", "queen", "pawn"};

try

{

int numberOfFigures = figures.Count();

Console.WriteLine(

"There are {0} figures in the board.",

numberOfFigures);

}

10. Використати мінімум 1 метод розширення управління запитами (на власний розсуд);  
int[] numbers = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

var evenNumbers = numbers.Where(n => n % 7 == 0);

foreach (var XK in evenNumbers)

{

Console.WriteLine(XK);

}