

Wydział Nauk Inżynierskich ANS w Nowym Sączu Optymalizacja dyskretna - projekt		
Temat: Programowanie dyskretnie. Wprowadzenie do programowania w języku c#. Operacje na tablicach, funkcje.		Symbol: OD_P1
Nazwisko i imię: Rydzek Arkadiusz	Ocena sprawozdania:	Zaliczenie:
Data wykonania ćwiczenia:	Oceniane efekty uczenia się: EUK1=....., EUK2=....., EUK3=....., EUK4=.....	

Wstęp

Celem zajęć było zapoznanie się z podstawowymi elementami języka C# oraz środowiskiem programistycznym Visual Studio Code (przygotowanie IDE do pracy z C#, pobranie odpowiednich rozszerzeń, .NET itd.). W ramach ćwiczeń wykonano kilka prostych programów, które miały na celu utrwalenie poznanych konstrukcji językowych, a konkretnie operacji na tablicach oraz zapisu/odczytu danych z plików tekstowych. Poniżej przedstawiono opis wykonanych zadań oraz kod źródłowy poszczególnych programów.

Zadanie 1

Napisz program w języku c#, który wczytuje elementy numeryczne do tablicy dwuwymiarowej $n \times n$. Wartości można podać z klawiatury oraz wczytać liczby losowe z zakresu od 0-100. Program posiada menu wyboru:

1. Wczytaj klawiaturą,
2. Wczytaj losowe,
3. Wyświetl tablice,
4. ESC - wyjście.

Poniżej przedstawiono kod źródłowy programu realizującego powyższe zadanie.

```

1 #nullable disable

3 Console.WriteLine("Enter the size of the matrix:");
4 int size = int.Parse(Console.ReadLine());
5 int[,] matrix = new int[size, size];
6 Random rand = new Random();

8 int option;

10 for (;;)
11 {
12     Console.WriteLine(
13         "1. Load matrix with your own values\n2. Load matrix with random values\n3. Display the matrix\n4. Exit"
14     );
15     option = int.Parse(Console.ReadLine());

17     switch (option)
18     {
19         case 1:
20             Console.WriteLine("Input your own values...");
21             for (int i = 0; i < size; i++)
22             {

```

```

23         for (int j = 0; j < size; j++)
24         {
25             Console.Write($"Enter value for position [{i},{j}]: ");
26             matrix[i, j] = int.Parse(Console.ReadLine());
27         }
28     }
29     break;
30     case 2:
31         Console.WriteLine("Loading random values.");
32         for (int i = 0; i < size; i++)
33         {
34             for (int j = 0; j < size; j++)
35             {
36                 matrix[i, j] = rand.Next(1, 101);
37             }
38         }
39         break;
40
41     case 3:
42         Console.WriteLine("Displaying the matrix:");
43         for (int i = 0; i < size; i++)
44         {
45             for (int j = 0; j < size; j++)
46             {
47                 Console.Write(matrix[i, j] + "\t");
48             }
49             Console.WriteLine();
50         }
51         break;
52     case 4:
53         Console.WriteLine("Exiting...");
54         return;
55     default:
56         Console.WriteLine("Invalid option.");
57         break;
58 }
59 }

```

Listing 1: Kod źródłowy programu realizującego zadanie 1

Program implementuje menu wyboru umożliwiające użytkownikowi wczytanie danych do tablicy dwuwymiarowej. Funkcjonalności programu obejmują:

1. Wprowadzenie wartości z klawiatury - użytkownik może ręcznie wprowadzić wartości dla każdej pozycji w tablicy,
2. Wygenerowanie liczb losowych z zakresu 1-100 - program automatycznie wypełnia tablicę losowymi wartościami,
3. Wyświetlenie zawartości tablicy - program prezentuje aktualną zawartość tablicy w formie czytelnej dla użytkownika,
4. Wyjście z programu - zakończenie działania aplikacji.

Program działa w pętli nieskończonej, umożliwiając wielokrotne wykonywanie operacji do momentu wyboru opcji wyjścia.

Zadanie 2

Napisz funkcję do odczytu i zapisu danych z pliku „p1.txt”. Wklej do sprawozdania zrzut ekranu z działającym programem. Menu:

1. Wczytaj klawiatura,
2. Wczytaj losowe,
3. Wyświetl tablice,
4. Wczytaj z pliku,
5. Zapisz do pliku,
6. ESC wyjście.

Poniżej przedstawiono kod źródłowy programu realizującego powyższe zadanie.

```
1 #nullable disable
2 using System.IO;

4 Console.WriteLine("Enter the size of the matrix:");
5 int size = int.Parse(Console.ReadLine());
6 int[,] matrix = new int[size, size];
7 Random rand = new Random();

9 int option;

11 for (; ; )
12 {
13     Console.WriteLine(
14         "1. Load matrix with your own values\n2. Load matrix with random values\n3. Display the matrix\n4. Load matrix from file\n5. Save matrix to file\n6. Exit"
15     );
16     option = int.Parse(Console.ReadLine());

18     switch (option)
19     {
20         case 1:
21             Console.WriteLine("Input your own values...");
22             for (int i = 0; i < size; i++)
23             {
24                 for (int j = 0; j < size; j++)
25                 {
26                     Console.Write($"Enter value for position [{i},{j}]: ");
27                     matrix[i, j] = int.Parse(Console.ReadLine());
28                 }
29             }
30             break;
31         case 2:
32             Console.WriteLine("Loading random values.");
33             for (int i = 0; i < size; i++)
34             {
35                 for (int j = 0; j < size; j++)
36                 {
37                     matrix[i, j] = rand.Next(1, 101);
38                 }
39             }
40         case 3:
41             // Display the matrix
42             for (int i = 0; i < size; i++)
43             {
44                 for (int j = 0; j < size; j++)
45                 {
46                     Console.Write(matrix[i, j] + " ");
47                 }
48                 Console.WriteLine();
49             }
50             break;
51         case 4:
52             // Load matrix from file
53             string filePath = "p1.txt";
54             if (File.Exists(filePath))
55             {
56                 using (StreamReader reader = new StreamReader(filePath))
57                 {
58                     string line;
59                     while ((line = reader.ReadLine()) != null)
60                     {
61                         string[] values = line.Split(' ');
62                         for (int i = 0; i < size; i++)
63                         {
64                             for (int j = 0; j < size; j++)
65                             {
66                                 matrix[i, j] = int.Parse(values[i * size + j]);
67                             }
68                         }
69                     }
70                 }
71             }
72             break;
73         case 5:
74             // Save matrix to file
75             string filePath2 = "p1.txt";
76             using (StreamWriter writer = new StreamWriter(filePath2))
77             {
78                 for (int i = 0; i < size; i++)
79                 {
80                     for (int j = 0; j < size; j++)
81                     {
82                         writer.Write(matrix[i, j] + " ");
83                     }
84                     writer.WriteLine();
85                 }
86             }
87             break;
88         case 6:
89             // Exit
90             break;
91     }
92 }
```

```

39         }
40         break;

42     case 3:
43         Console.WriteLine("Displaying the matrix:");
44         for (int i = 0; i < size; i++)
45         {
46             for (int j = 0; j < size; j++)
47             {
48                 Console.Write(matrix[i, j] + "\t");
49             }
50             Console.WriteLine();
51         }
52         break;
53     case 4:
54         try
55         {
56             Console.Write("Enter filename (e.g., p1.txt): ");
57             string filename = Console.ReadLine();
58             StreamReader sr = new StreamReader(filename);

60             int fileSize = int.Parse(sr.ReadLine());

62             if (fileSize != size)
63             {
64                 size = fileSize;
65                 matrix = new int[size, size];
66             }

68             for (int i = 0; i < size; i++)
69             {
70                 string[] values = sr.ReadLine().Split(' ');
71                 for (int j = 0; j < size; j++)
72                 {
73                     matrix[i, j] = int.Parse(values[j]);
74                 }
75             }

77             sr.Close();
78             Console.WriteLine("Matrix loaded successfully from file.");
79         }
80         catch (Exception e)
81         {
82             Console.WriteLine("Exception: " + e.Message);
83         }
84         break;
85     case 5:
86         try
87         {
88             StreamWriter sw = new StreamWriter("p1.txt");

90             sw.WriteLine(size);

92             for (int i = 0; i < size; i++)
93             {

```

```

94         for (int j = 0; j < size; j++)
95         {
96             sw.Write(matrix[i, j]);
97             if (j < size - 1)
98                 sw.Write(" ");
99         }
100        sw.WriteLine();
101    }

103        sw.Close();
104        Console.WriteLine("Matrix saved successfully to p1.txt");
105    }
106    catch (Exception e)
107    {
108        Console.WriteLine("Exception: " + e.Message);
109    }
110    break;

112    case 6:
113        Console.WriteLine("Exiting...");
114        return;
115    default:
116        Console.WriteLine("Invalid option.");
117        break;
118    }
119 }

```

Listing 2: Kod źródłowy programu realizującego zadanie 2

Program rozszerza funkcjonalność z zadania pierwszego o operacje na plikach. Dodano dwie nowe opcje menu:

4. Wczytanie tablicy z pliku - program odczytuje rozmiar tablicy oraz jej zawartość z pliku tekstowego. Format pliku zawiera w pierwszej linii rozmiar tablicy, a w kolejnych liniach wartości elementów oddzielone spacjami,
5. Zapis tablicy do pliku - program zapisuje aktualną zawartość tablicy do pliku „p1.txt” w formacie zgodnym z opcją odczytu.

Operacje na plikach są zabezpieczone mechanizmem obsługi wyjątków (try-catch), co zapewnia stabilność działania programu w przypadku problemów z dostępem do pliku lub nieprawidłowym formatem danych.

Podsumowanie

W ramach ćwiczeń zapoznano się z podstawami programowania w języku C#. Wykonane zadania pozwoliły na praktyczne zastosowanie poznanych konstrukcji językowych, takich jak operacje na tablicach oraz obsługa plików tekstowych, co będzie przydatne na dalszych zajęciach z optymalizacji dyskretnej.