**Programozás Teszt**

Név: Veres Zoltán – DZAE6I
Dátum: 2023-01-29

A teszt nem jegyre megy. Ezért fontos,  hogy önállóan old meg. Ez az alap tudás amit Prog 1. tárgy után tudnod kell. Erre fogod építeni a többi programozás tárgyat. Ha ezt nem tanulod meg, nem fogsz tudni mire építeni. Ezért kérlek vedd becsületesen, komolyan.

Anyag: I. - IX. fejezetek a 133 oldalig.
(ILLÉS ZOLTÁN Programozás C# nyelven, BP, 2005.)

A megoldásokat a feladatok közé írd be!

1. Mi a változó és az állandó között a különbség?
    1. A konstansokat deklaráláskor látjuk el értékkel (és típussal) ezekután pedig legfeljebb olvassuk az értékét, míg a változókat definiáláskor (legfeljebb típussal kell ellátni és) a program futása közben bármikor értékadó vagy értékváltoztató utasítással módosíthatjuk.
    2. 24-25. oldal http://compalg.inf.elte.hu/~tony/Informatikai-Konyvtar/09-Programozas%20C-sharp%20nyelven/Programozas-Csharp-nyelven-Konyv.pdf
2. Mi a különbség egy statikus és egy dinamikus élettartamú változó között?
    1. Minden változó definiáláskor (módosító jelzők hiányában), dinamikus élettartamú, azaz az éppen futó deklarációs program-blokk végén a Garbage Collector (GC) kitörli a memóriából.
    2. Statikus módosító jelző használatakor a változók deklarálásánál elérjük, hogy a változó a program teljes futása alatt memóriában tartózkodjon és osztály inicializálás nélkül is hozzá férhető legyen az értéke (már ha osztályon belül helyeztük el a változó deklarálását)
    3. 26. oldal http://compalg.inf.elte.hu/~tony/Informatikai-Konyvtar/09-Programozas%20C-sharp%20nyelven/Programozas-Csharp-nyelven-Konyv.pdf
3. Definiáljon két szöveg típusú változót, adja meg a hosszukat!

```
public static class TestTask3
    {
        public static void run()
        {
            Random random = new Random();
            string text_first, text_second;

            text_first = Helper.RandomString(random.Next(32));
            text_second = Helper.RandomString(random.Next(32));

            Console.WriteLine($"First text: \"{text_first}\" -- Length:
{text_first.Length} characters...");
            Console.WriteLine($"Second text: \"{text_second}\" -- Length: {text_sec-
ond.Length} characters...");
        }
}
```

    1. 28. oldal http://compalg.inf.elte.hu/~tony/Informatikai-Konyvtar/09-Programozas%20C-sharp%20nyelven/Programozas-Csharp-nyelven-Konyv.pdf

4. Milyen szám-szöveg konverziós lehetőségeket ismer?
    1. 8.toString(); - Method conversion
    2. (string) 8; - Type casting
    3. Regex.Match(”8 976 123”, @"^[a-zA-Z][a-zA-Z0-9]*$").Value; - Kiválasztjuk a használható karaktereket
    4. 37-38. oldal http://compalg.inf.elte.hu/~tony/Informatikai-Konyvtar/09-Programozas%20C-sharp%20nyelven/Programozas-Csharp-nyelven-Konyv.pdf
    5. 40. oldal http://compalg.inf.elte.hu/~tony/Informatikai-Konyvtar/09-Programozas%20C-sharp%20nyelven/Programozas-Csharp-nyelven-Konyv.pdf
5. Milyen tömbök létrehozását támogatja a C# nyelv?
    1. Egydimenziós vektorok - int[] szamok;
    2. Multidimenziós vektorok - string[,] nevek; nevek = new string[2, 4];
    3. Vektorok vektora - int[][] szamok= new int[2][]; new int[] {5, 6};
    4. Ismeretlen meretu vektorok vektora - int[, , ] harom = new int[4, 5, 3];
    5. A fent felsorolt típusok keverése és összefűzése - int[][,,][,] mix;
    6. 43-46. oldal http://compalg.inf.elte.hu/~tony/Informatikai-Konyvtar/09-Programozas%20C-sharp%20nyelven/Programozas-Csharp-nyelven-Konyv.pdf
6. Mi a különbség a tömb és a struktúra között?
    1. A struktúrában tárolt adatokra vonatkozik a láthatóság (private, public, internal).
    2. A tömb csak egy szinonima a vektorra, azonos adatok egy név alá csoportosítására használjuk.
    3. 48. oldal http://compalg.inf.elte.hu/~tony/Informatikai-Konyvtar/09-Programozas%20C-sharp%20nyelven/Programozas-Csharp-nyelven-Konyv.pdf
7. Határozzuk meg egy tömb legnagyobb elemét!

```
public static class TestTask7
    {
        public static void run()
        {
            int[] numbers = { 11, 22, 33, 44, 55, 66, 77, 88, 99};
            int largest_integer_value = numbers.Max();
            int largest_integer_index = Array.IndexOf(numbers, largest_integer_value);

            Console.WriteLine($"The array: [{String.Join(", ", numbers)}]");
            Console.WriteLine($"Largest integer value in the array: {largest_inte-
ger_value} -- Index position of the value: {largest_integer_index}");
        }
}
```

8. Ábrázoljuk struktúra típussal az alma legjellemzőbb adatait (név, szín, méret)! Készítsünk 5 elemű alma vektort!

```
public static class TestTask8
    {
        struct Alma
        {
            public string nev;
```

```
            public string szin;
            public int meret;
            public override string ToString()
            {
                return string.Format($"Nev: {nev}, Szin: {szin}, Meret: {meret}");
            }
        };

        public static void run()
        {
            Alma jonatan = new Alma(); jonatan.nev = "Jonatán"; jonatan.szin = "Pi-
ros"; jonatan.meret = 10;
            Alma golden = new Alma(); golden.nev = "Golden"; golden.szin = "Sargas";
golden.meret = 12;
            Alma green = new Alma(); green.nev = "Green"; green.szin = "Zold";
green.meret = 8;
            Alma torpe = new Alma(); torpe.nev = "Torpe"; torpe.szin = "Piros";
torpe.meret = 5;
            Alma vad = new Alma(); vad.nev = "Vad"; vad.szin = "Zold"; vad.meret = 7;

            Alma[] Almak = new Alma[] { jonatan, golden, green, torpe, vad };

            foreach (Alma alma in Almak)
            {
                Console.WriteLine($"Alma -- {alma.ToString()}");
            }
        }
```

9. Milyen elöltesztelő és hátultesztelő ciklusokat használhatunk?
    1. Elől-tesztelő ciklusok:
        1. While
        2. For
    2. Hátul-tesztelő ciklusok:
        1. Do … While
    3. 55. oldal http://compalg.inf.elte.hu/~tony/Informatikai-Konyvtar/09-Programozas%20C-sharp%20nyelven/Programozas-Csharp-nyelven-Konyv.pdf
10. Mire használható a break utasítás?
    1. A „while”, „do”, „for” vagy „switch” utasítás végrehajtása során, a következő ciklusra irányítja a vezérlést.
    2. 59. oldal http://compalg.inf.elte.hu/~tony/Informatikai-Konyvtar/09-Programozas%20C-sharp%20nyelven/Programozas-Csharp-nyelven-Konyv.pdf
11. Írjon rövid programot, amely beolvassa egy focicsapat, adott fordulóban szerzett pontszámát (0,1,3) egy egész típusú változóba, majd felhasználva a switch utasítást a beolvasott érték alapján kiírja a következő szövegeket: győzelem, döntetlen, vereség, hibás adat.

```
public static class TestTask11
    {
        public static void run()
        {
            Console.WriteLine("Please input the result of the football match: ");

            string? user_input = Console.ReadLine();
            if (user_input == null)
            {
                Console.WriteLine($"Invalid user input: {user_input}");
```

```
            return;
        }

        int result_score = String.IsNullOrEmpty(user_input) ? 4 :
Helper.stringToInt(user_input);
        string result_string;

        switch (result_score)
        {
            case 0:
                result_string = "Lost";
                break;
            case 1:
                result_string = "Tie";
                break;
            case 3:
                result_string = "Win";
                break;
            default:
                result_string = "ERROR!!!";
                break;
        };

        Console.WriteLine($"The result of the football match is: {result_score} --
{result_string}");
    }
}
```

12. Írjon programot, amelyik beolvas egy kettővel osztható, 10 és 100 közé eső egész számot! (Ha rossz értéket adnak meg, akkor addig folytassa a beolvasást, amíg a feltételeknek megfelelő számot nem sikerül megadni!)

```
public static class TestTask12
    {
        public static void run()
        {
            bool continue_looping = true;
            do
            {
                Console.WriteLine("Please input an even number between 10 and
100...:");
                string? user_input = Console.ReadLine();
                int user_number = Helper.stringToInt(user_input);

                if (!String.IsNullOrEmpty(user_input))
                {
                    if (user_number <= 100 && user_number >= 10)
                    {
                        if (user_number % 2 == 0)
                        {
                            Console.WriteLine("Congratulations... your input is cor-
rect!");
                            continue_looping = false;
                        }
                        else
                        {
                            Console.WriteLine("Sorry, but the given user input needs
to be an even number...");
                        }
                    }
                    else
                    {
                        Console.WriteLine("Sorry, but the given user input needs to be
between 10 and 100...");
```

```
                    }
                }
                else
                {
                    Console.WriteLine("Sorry, but the given user input seems to be
empty...");
                }
            }
            while (continue_looping);
        }
}
```

13. Milyen paraméterátadási módokat ismer?
    1. Érték szerinti paraméterátadás
    2. Függvényeredmény paramétere
    3. Tömbök paraméterátadása
    4. Parancssori paraméterek
    5. Params kulcsszavas változó méretű lista
    6. 64-70. oldal http://compalg.inf.elte.hu/~tony/Informatikai-Konyvtar/09-Programozas%20C-sharp%20nyelven/Programozas-Csharp-nyelven-Konyv.pdf
14. Mikor definiálhatunk azonos névvel függvényeket egy osztályban?
    1. Ha a polimorfizmust (overloading) módszertanát szeretnénk alkalmazni, azaz különböző bemeneti paraméter típusokra és mennyiségre, más-más függvény végrehajtást kívánunk elérni.
15. Hogyan kell változó paraméterszámú függvényt használni?
    1. params object[] list … Objektum lista, végig iterálhatjuk.
    2. 71. oldal http://compalg.inf.elte.hu/~tony/Informatikai-Konyvtar/09-Programozas%20C-sharp%20nyelven/Programozas-Csharp-nyelven-Konyv.pdf
16. Írjon egy függvényt, amely a paraméterként megadott néhány név közül a leghosszabb hosszat adja meg!

```
public static class TestTask16
    {
        public static void run()
        {
            string[] name_list = { "Zoltán", "Dániel", "Megszentségteleníthe-
tetlenségeskedéseitekért Magda", "Hosszú Katinka" };

            int longest_name_index = 0;
            int longest_name_length = 0;
            int current_index = 0;

            foreach (string name in name_list)
            {
                if (name.Length > longest_name_length)
                {
                    longest_name_index = current_index;
                    longest_name_length = name.Length;
                }

                current_index++;
            }
```

```
            Console.WriteLine($"The longest name is: {name_list[longest_name_in-
dex]}");
            Console.WriteLine($"The longest name index is: {longest_name_index}");
        }
}
```

17. Mi a különbség egy struktúra és egy osztály között?
    1. Struct az egy érték típus, Class az egy referencia típus
    2. Struct a stack memória része, Class a heap memória része
    3. Szükséges memória szerint, Struct kevesebb allokációt vesz igénybe, de a Class rendelkezik garbage collectionnal a heapben való allokáció miatt.
    4. Mivel a Struct érték típusú, paraméterként való átadása memóriában való másolást jelent, a meglévő, referenciaként átadott Classokhoz képest.
    5. 47. oldal: http://compalg.inf.elte.hu/~tony/Informatikai-Konyvtar/09-Programozas%20C-sharp%20nyelven/Programozas-Csharp-nyelven-Konyv.pdf
    6. 2023-02-04 https://schwabencode.com/blog/2021/09/20/dotnet-class-vs-struct-performance
18. Milyen szerepe van a konstruktoroknak, destruktoroknak? Milyen konstruktorok definiálhatók?
    1. Konstruktorok legfőbb szerepe a Class és Struct felállítása, kezdeti műveletek lefuttatása, kezdőértékek beállítása.
    2. Destructorok feladata a garbage collector előtt az osztály felkészítése a memóriából való törlésre.
    3. Konstruktorok típusai:
        1. Statikus konstruktor – Csakis egyszer fut le a program kezdetekor
        2. Privát konstruktor – Megakadályozza, hogy példányosítsunk vagy meghívjunk egy osztályt.
    4. 80 - 88. oldal: http://compalg.inf.elte.hu/~tony/Informatikai-Konyvtar/09-Programozas%20C-sharp%20nyelven/Programozas-Csharp-nyelven-Konyv.pdf
19. Mi a különbség az osztály, az absztrakt osztály is az interface között?
    1. Az osztály feladata, hogy keretbe foglalja a program egy egységét.
    2. Az interface szerepe, hogy ezen egységnek meghatározza a szerepét, anélkül, hogy leírná a megoldást.
    3. Az absztrak osztály szerepe, hogy az egységnek leírást adjon a feladat megoldásához.
    4. 100. oldal: http://compalg.inf.elte.hu/~tony/Informatikai-Konyvtar/09-Programozas%20C-sharp%20nyelven/Programozas-Csharp-nyelven-Konyv.pdf
20. Definiáljon bútor osztályt a jellemző tulajdonságokkal! (Bútor neve, alapanyaga, rendeltetése, ára) A megadott tulajdonságokhoz készítse el a kezelő függvényeket!

```
public class Bútor
    {
        string megnevezes { get; set; }
```

```
        string alapanyag { get; set; }
        string rendeltetes { get; set; }
        float ar { get; set; }

        public Bútor(
            string megnevezes,
            string alapanyag,
            string rendeltetes,
            float ar
        ) {
            this.megnevezes = megnevezes;
            this.alapanyag = alapanyag;
            this.rendeltetes = rendeltetes;
            this.ar = ar;
        }
}
```

21. Készítsen lapraszerelt névvel interface-t, amiben az összeszerelési utasítást írjuk elő! Módosítsa az előző bútor osztályt lapraszerelt bútorra, amelyik implementálja a lapraszerelt interface-t, biztosítva azt, hogy ennek a típusnak biztosan legyen összeszerelési utasítása.

```
public interface ILapraszerelt
    {
        public string getOsszeszerelesiUtasitas();
    }

    public class Lapraszerelt: ILapraszerelt
    {
        public string osszeszerelesi_utasitas;
        public Lapraszerelt(string osszeszerelesi_utasitas)
        {
            this.osszeszerelesi_utasitas = osszeszerelesi_utasitas;
        }

        public string getOsszeszerelesiUtasitas()
        {
            return this.osszeszerelesi_utasitas;
        }
}
```

22. Mit jelent a checked, unchecked kulcsszó, hogyan tudjuk használni?

    1. Számokkal való műveleteknél, hogy a számábrázolás sajátos határait súrolva, ne legyenek figyelmetlenségből adód hibák, kódunkat a Checked blokkba helyezhetjük, hogy biztosan kivételt dobjon a nem várt műveletekor és értékek elérésénél (overflow).
    2. 116. oldal: http://compalg.inf.elte.hu/~tony/Informatikai-Konyvtar/09-Programozas%20C-sharp%20nyelven/Programozas-Csharp-nyelven-Konyv.pdf
    3. 2023-02-05 https://www.javatpoint.com/c-sharp-checked-and-unchecked#:~:text=C%23%20provides%20checked%20and%20unchecked,ignored%20and%20result%20is%20truncated.

23. Hogyan tudunk saját kivételt (típust) definiálni?

    1. Származtatjuk az eredeti ősből: Exception vagy egy még pontosabból, ha megvan a hibánk amit kezelnénk. Utána, hogy le is kezeljük, a try ágban, egy vizsgálatunkkor dobunk (throw) egy hibát. Utána egy catch ágban elkapva a hibánkat, megakadályozhatjuk a pogram-futás megszűntetését, hibára hivatkozva.
    2. 117. oldal: http://compalg.inf.elte.hu/~tony/Informatikai-Konyvtar/09-Programozas%20C-sharp%20nyelven/Programozas-Csharp-nyelven-Konyv.pdf

24. Mi a különbség a bináris és szöveges fájl között?

    1. Szöveges fájlok bármiféle komplexebb dekódolás (persze a karakter kódolás ide is szükséges) nélkül beolvashatók, míg a bináris fájlok dekódolásánál tudnunk kell, hogy a megjelenítés és az adat-folyam során mit várnánk el. Például egy kép esetében, nem tudunk sok mindent kezdeni egy szöveges megjelenítéssel, de ha a pixeleket egymás után rakjuk ki, jutunk is valamerre.

    2. 125 - 126. oldal: http://compalg.inf.elte.hu/~tony/Informatikai-Konyvtar/09-Programozas%20C-sharp%20nyelven/Programozas-Csharp-nyelven-Konyv.pdf

25. Írj egy minimális programot, ami bekér egy fájl nevet és a tartalmát a képernyőre listázza.

    1. Asd

```
public static class TestTask25
    {
        public static void run()
        {
            Console.WriteLine("Please input a file name with path:...");
            string? file_path = Console.ReadLine();
            if (String.IsNullOrEmpty(file_path))
            {
                file_path = "C:\\Users\\Admin\\Desktop\\projects\\THE\\Programozas
modszertana\\THEProgTerv1\\lorem.txt";
            }

            string text = System.IO.File.ReadAllText(@file_path);
            System.Console.WriteLine($"Contents of file: {@file_path}");
            System.Console.Write(text);
        }
 }
```

26. Tárolunk egy nevet és egy számot, írjuk ki ezeket adatok .bin névvel bináris, majd adatok.txt névvel szöveges formában!

    1. 2022-02-05 https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/file-system/how-to-write-to-a-text-file

```
public static class TestTask26
    {
        public static async Task run()
        {
            Random random = new Random();
            string szoveg = Helper.RandomString(random.Next(32));
            int szam = Helper.RandomNumber(random.Next(32000));

            Console.WriteLine($"Outputting the following string: \"{szoveg}\"");
            Console.WriteLine($"Outputting the following number: \"{szam}\"");

            await File.WriteAllTextAsync(
                "C:\\Users\\Admin\\Desktop\\projects\\THE\\Programozas modszer-
tana\\THEProgTerv1\\adatok.txt",
                $"{szoveg} -- {szam}"
            );

            using (BinaryWriter binary_writer = new BinaryWriter(File.Open(
              @"C:\\Users\\Admin\\Desktop\\projects\\THE\\Programozas modszer-
tana\\THEProgTerv1\\bin.bin",
              FileMode.Create)))
            {
                binary_writer.Write($"{szoveg} -- {szam}");
            }
        }
```

```
}
```