.NET FEJLESZTÉS C# NYELVEN

6. forduló

RENDELKEZÉSRE ÁLLÓ IDŐ:

40:00

Ismertető a feladathoz

Sikeresen átálltatok a másodlagos rendszerre, viszont annak az adatbázisa elavult.

Szükségetek lesz migrációra, ami újabb kihívások elé állít titeket.

Felhasznált idő: 00:16/40:00 **Elért pontszám:** 0/9

1. feladat 0/2 pont

Nagymennyiségű adatot szeretnétek a lehető legoptimálisabban tárolni. Milyen szerializert alkalmaztok?

Válasz

- XmlSerializer
- BinaryFormatter
- DataContractSerializer
- DataContractJsonSerializer

Magyarázat

A BinaryFormatter a helyes választás, mivel a bináris formátum a legoptimálisabb választás nagymennyiségű adat tárolására.

2. feladat 0/0 pont

Érzékeny információt kell küldenetek a másodlagos rendszernek, és meg szeretnétek bizonyosodni, hogy érintetlenül érkezett meg az üzenet. Mit használtok erre az alábbiak közül?

Válasz

- X509Certificate2.SignHash
- RSACryptoServiceProvider.Encrypt
- UnicodeEncoding.GetBytes
- Marshal.ZeroFreeBSTR

Magyarázat

Az X509-es digitális tanúsítvány használható a hash-elt adat aláírásához. Ha a másik fél a Verify metódust használja, ellenőrizni tudja, hogy a hash megváltozott-e.

A metódus asziemtrikus algoritmussal titkosít. Nem lehetük biztosak, hogy az üzenetet nem manipulálták.

GetBytes a szöveget bájt sorozatra alakítja. Ez nem védi semmilyen szempontból az adatot.

A ZeroFreeBSTR metódus a nem biztonságos sztringek memóriafelszabadítására használatos.

3. feladat 0/7 pont

Melyik metódus adja vissza a legnagyobb számot?

```
public class RecordExcersise
{

    public record MyClass(string Z, int X, int Y);

    public static string Case1()
    {
        var I = new MyClass("Cream", 5, 7); int III = 1;
        var II = Enumerable.Repeat(I, 5).ToList().ToDictionary(_ => III++);
        var IIII = II.Select(p3 => p3.Value with { X =+ I.Y });
        return IIII.Sum(p5 => p5.X).ToString();
    }

    public static string Case2()
    {
        var I = new MyClass("Triumph", 6, 9); int III = 2;
        var II = Enumerable.Repeat(I, 6).ToList().ToDictionary(_ => III++);
        var IIII = II.Select(p3 => p3.Value with { X =+ I.Y });
        return IIII.Sum(p5 => p5.X).ToString();
    }

    public static string Case3()
    {
        var I = new MyClass("36", 7, 10); int III = 3;
        var II = Enumerable.Repeat(I, 7).ToList().ToDictionary(_ => III++);
        var IIII = II.Select(p3 => p3.Value with { X =- I.Y });
        return IIII.Sum(p5 => p5.X).ToString();
    }

    public static string Case4()
    {
        var I = new MyClass("Forever", 10, 11); int III = 4;
        var II = Enumerable.Repeat(I, 8).ToList().ToDictionary(_ => III++);
        var IIII = II.Select(p3 => p3.Value with { X =- I.Y });
        return IIII.Sum(p5 => p5.X).ToString();
    }

}
```

Válasz

- ○ Case1();
- ○ Case2();
- ○ Case3();
- ○ Case4();

Magyarázat

Case1(); Kimenete: 35

Case2(); Kimenete: 54

Case3(); Kimenete: -70

Case4(); Kimenete: -88