

Odpovědi pište na zvláštní odpovědní list s vaším jménem a fotografií. Pokud budete odevzdávat více než jeden list s řešením, tak se na 2. a další listy nezapomeňte podepsat. Do zápatí všech listů vždy napište i/N (kde i je číslo listu, N je celkový počet odevzdaných listů).

### Otázka č. 1

Předpokládejte následující část kódu:

```
public struct S {
    public int A { get; set; }
    public int B { get; set; }
}

public class X {
    private S s;

    public X(S s) {
        this.s = s;
    }

    public int A {
        get { return s.A; }
    }

    public int B {
        get { return s.B; }
    }
}
```

Rozhodněte, zda jsou struktura `S` a třída `X` thread-safe (bez ohledu na kontext použití). U obou vysvětlíte, proč ano, resp. proč ne.

**[1,5 bodu]**

### Otázka č. 2

Předpokládejte následující program:

```
public class Prg2 {
    public static async void f(double a) {
        var sinA = Task.Run(() => Math.Sin(a));

        var b = double.Parse(
            await Task.Run(() => Console.ReadLine())
        );

        var cosB = Task.Run(() => Math.Cos(b));

        Console.WriteLine(await sinA + await cosB);
    }

    public static void Main() {
        f(Math.PI);
        System.Threading.Thread.Sleep(10000);
    }
}
```

Vysvětlíte, jakým způsobem se bude tento program chovat, zvlášť pro jednotlivé relevantní části kódu vysvětlíte, ve kterých vláknech se budou provádět. Vysvětlíte proč.

**[1 bod]**

Pokud uživatel při opakovaných bězích programu zadá na standardní vstup programu opakovaně to samé reálné číslo, liší se nějak výstup programu v jednotlivých bězích? Liší se nějak chování programu podle toho, kdy od spuštění programu uživatel takové číslo zadá a stiskne klávesu Enter (např. zda po 2 sekundách, nebo zda po 20 sekundách)?

**[1 bod]**

### Otázka č. 3

Napište příklad třídy (se všemi potřebnými deklaracemi), u které by bylo vhodné naimplementovat `OnDeserialized` callback, který do vaší implementace také zahrňte.

Dále vysvětlíte, jaký problém a v jaké situaci by mohl nastat, pokud by byl uvedený callback z vaší třídy vypuštěn.

**[1,5 bodu]**

### Otázka č. 4

Uvedte a vysvětlíte konkrétní příklad nějaké jedné situace, kdy by bylo vhodné a pro správné fungování potřebné použít `StringComparer.InvariantCultureIgnoreCase`.

Dále uveďte situaci, kde by naopak bylo vhodnější se spolehnout na „defaultní“ porovnávání řetězců. Vysvětlíte, co je tímto „defaultním“ porovnáváním myšleno.

**[1,5 bodu]**

### Otázka č. 5

Předpokládejte následující program:

```
using System;
using System.Collections.Generic;
using System.Linq;

public class X {
    public List<int> Where(Predicate<int> p) {
        var list = new List<int>();
        for (int i = 0; i < 10; i++) {
            if (p(i)) list.Add(i);
        }
        return list;
    }
}

public class Prg5 {
    public static void Main() {
        var x = new X();
        var odd = from a in x
            where a % 2 != 0
            where a > 5
            select a * 2;

        foreach (var b in odd) {
            Console.WriteLine(b);
            break;
        }
    }
}
```

Rozhodněte, jaký bude typ proměnné `odd`. Popište datovou strukturu, která bude v proměnné `odd` „uložena“, resp. na kterou proměnná `odd` ukazuje, je-li referenčního typu.

**[1 bod]**

Srovnejte první a druhou `where` klauzuli uvedeného LINQ dotazu z pohledu vyhodnocení uvedených podmínek. Kolikrát se podmínky jednotlivých `where` klauzulí vyhodnotí, a kdo bude vyhodnocení každé podmínky „provádět“? Liší se z tohoto pohledu nějak uvedené dvě `where` klauzule?

**[1 bod]**

**Otázka č. 6**

Metadata assembly Library.dll se v nástroji ildasm zobrazí následujícím způsobem:



Dále lze zjistit, že implementace metody `A.m` je v CIL (MSIL) assembleru následující:

```
.method public hidebysig static int32
m() cil managed
{
    .maxstack 1
    .locals init ([0] int32 r)
    ldsfld     int32 A::v
    stloc.0
    ldc.i4.0
    stsfld     int32 A::v
    ldloc.0
    ret
}
```

Zapište v C# deklaraci a tělo takové metody `A.m` (která se bude chovat stejně jako výše uvedený kód).

**[1 bod]**

Bylo by možné v C# napsat program, který by v uvedené assembly Library.dll změnil tělo metody `A.m` tak, že by vždy na standardní výstup vypsala jen hodnotu proměnné `v`, a vrátila hodnotu `0`? Pokud ano, jaký nástroj/API byste k tomu nejlépe použili?

**[0,5 bodu]**