

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:

```
KeyValuePair<string, int> x =
    new KeyValuePair<string, int>(
        key: "two", value: 14
    );
Console.WriteLine(x.Key);
Console.WriteLine(x.Value);
x = new { Key = "one", Value = 12 };
```

Je možné v C# zapsat uvedené přiřazení do proměnné x na posledním řádku? Vysvětlete proč.

[1 bod]

Otázka č. 2

Předpokládejte následující třídu:

```
class X {
    private List<int> data;

    public X(IEnumerable<int> source) {
        data = new List<int>(source);
    }

    public int m(int a) {
        int value = 0;
        for (int i = 0; i < data.Count; i++) {
            value = Process(value, data[i]);
        }

        if (value == 0 || value < a) {
            return value;
        } else if (value == a) {
            data.Add(a);
            return -1;
        } else {
            data.RemoveAt(0);
            return -1;
        }
    }

    private static int Process(int a, int b) {
        return ... // Some complex math here!
    }
}
```

Dále můžete předpokládat, že metoda Process provádí nějaký složitý matematický výpočet, ke kterému používá jen běžné operátory jazyka C# a metody ze třídy `Math`. Je třída `X` thread-safe? Pokud ano, tak detailně vysvětlete proč. Pokud ne, tak ji nejvhodnějším způsobem opravte tak, aby thread-safe byla.

[1,5 bodu]

Otázka č. 3

Předpokládejte následující třídu:

```
class X {
    public static async Task<int> m(int a, int b) {
        var sr = new StreamReader("x.txt");

        string line;
        int sum = 0;
        int i = 0;
        while (
            (line = await sr.ReadLineAsync()
            ) != null)
        {
            if (i >= a && i <= b) {
                sum += line.Length;
            }
            i++;
        }

        sr.Dispose();
        return sum;
    } }
}
```

Je to nejvhodnější implementace metody `m`? Pokud ano, vysvětlete proč. Pokud ne, vysvětlete proč.

[1 bod]

Přepište třídu `X` do ekvivalentního kódu bez použití `async/await` patternu (tj. využijte pouze `task continuations`).

[1,5 bodu]

Otázka č. 4

Předpokládejte následující program:

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

public class Prg4 {
    public static void Main() {
        int[] src = { 1, 5, 2, 4, 3 };
        (
            from x in src.AsParallel().AsOrdered()
            where x > 2 select x * 2
        ).ForEach(
            x => Console.WriteLine(x)
        );
    } }
}
```

Napište, co program vypíše na standardní výstup. Bude výsledek programu na uvedených datech vždy stejný?

[1 bodu]

Detailně vysvětlete, jakým způsobem se bude tento program chovat, jakým způsobem bude probíhat vyhodnocení LINQ dotazu (nakreslete obrázek), a pro jednotlivé relevantní části kódu vysvětlete, ve kterých vláknech se budou provádět.

[1 bod]

Otázka č. 5

Předpokládejte následující program:

```
using System;
using System.Reflection;

class Program {
    static void Mai() {
        AppDomain ad2 =
            AppDomain.CreateDomain("ad2");

        X x = (X) ad2.CreateInstanceAndUnwrap(
            Assembly.GetExecutingAssembly()
                .ToString(),
            typeof(X).Name
        );
        x.SetA(10);
        Console.WriteLine(x.f(1));
        // x.g(new Y { Value = 42 });
    }
}

class X : MarshalByRefObject {
    public int a;

    public void SetA(int value) {
        a = value;
    }

    public int f(int b) {
        return a + b;
    }

    public void g(Y y) {
    }
}

class Y {
    public int Value;
}
```

Detailně vysvětlete, co se přesně bude dít při vykonávání řádku s voláním metody `x.f` (tj. mimo jiné jak `a` v jakých vláknech se bude provádět uvedený kód).

[1 bod]

Bylo by možné odkomentovat zakomentovaný řádek volající metodu `x.g`? Vysvětlete proč.

[0,5 bodu]

Otázka č. 6

Uveďte příklad aplikace, resp. její části, kde byste použili UDP komunikaci. S využitím API .NETu popište kostru takového kódu pro komunikaci přes UDP/IP. V čem bude v dané situaci výhoda oproti komunikaci přes TCP/IP?

[1,5 bodu]