

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).

### Společná část pro otázky označené X

V příloze najdete dokumentaci metody `RemoveAt` standardní .NET třídy `List<T>`.

#### Otázka č. 1 (X)

Předpokládejte, že metodu `List<T>.RemoveAt` standardní knihovny implementujeme a v rámci implementace ní chceme napsat sadu unit testů – s využitím testovacího frameworku `MSTest`, `nUnit` nebo `xUnit` napište sadu alespoň 3 vhodných relevantních unit testů testujících rozdílné aspekty chování metody `List<T>.RemoveAt`.

**[1 bod]**

### Společná část pro otázky označené Y

Předpokládejte následující implementaci metody, která se bez chyb přeloží, a jejíž implementace je po funkční stránce otestována, že se chová dle našich požadavků:

```
/// <summary>
/// Removes one item from the list.
/// </summary>
/// <param name="list">The list</param>
/// <param name="item">The item to remove</param>
/// <returns>Result of item removal</returns>
static bool PleaseRemoveOneItemFromTheListPassed(
    List<int> list, int item)
{
    try {
        (
            list ?? throw new ArgumentException()
        ).RemoveAt(item);
    } catch (ArgumentOutOfRangeException) {
        try {
            list.RemoveAt(list.Count - 1);
        } catch {
            return false;
        }
    }
    return true;
}
```

#### Otázka č. 2 (X, Y)

Je vzhledem k výše implementovanému chování metody `PleaseRemoveOneItemFromTheListPassed` dostatečný (vhodně formulovaný) připojený dokumentační komentář? A je vůbec metoda a její parametry vhodně pojmenované? Detailně vysvětlete proč. Pokud je vaše odpověď ne, tak navrhněte lepší znění dokumentačního komentáře, resp. lepší pojmenování metody, případně jejích parametrů (ideálně anglicky, ale je možné i česky/slovensky).

**[1 bod]**

#### Otázka č. 3 (X, Y)

Je výše uvedená implementace statické metody `PleaseRemoveOneItemFromTheListPassed` nejvhodnější? Detailně vysvětlete proč. Pokud je vaše odpověď ne, tak napište, jak by měla být naprogramována lépe (se zachováním otestované funkcionality).

**[1 bod]**

### Otázka č. 4

Předpokládejte následující program, který se bez chyb přeloží:

```
using System;

class Student {
    public string FirstName { get; set; }
    public string LastName { get; set; }
}

struct Color {
    public byte Red { get; }
    public byte Green { get; }
    public byte Blue { get; }
    public Color(byte red, byte green, byte blue) {
        Red = red; Green = green; Blue = blue;
    }
}

class Prg4 {
    public static void m() {
        var pink = new Color(255, 192, 203);
        var jan = new Student {
            FirstName = "Jan",
            LastName = "Radsetoulal"
        };
        Console.WriteLine($"{jan} {pink}");
    }

    static void Main(string[] args) {
        Console.WriteLine("I'm here!");
        m();
        Console.WriteLine("I'm dying, so sad!");
    }
}
```

(A) Napište, co program po spuštění vypíše na standardní výstup, a detailně vysvětlete proč. Je výstup programu daný návrhem jazyka C# nebo implementací .NETu?

**[1 bod]**

(B) Předpokládejte, že kód uvedené třídy `Prg4` přeložíme C# překladačem do .EXE assembly, a z ní poté CIL kód metody `m` dekompilujeme zpět do C# decompilerem, který nepodporuje koncept interpolovaných řetězců (např. `ILSpy` s vhodným nastavením). Jak bude asi dekompilovaný C# kód vypadat? Jak je deklarována použitá varianta metody `Console.WriteLine`? Napište co nejpřesněji.

**[1 bod]**

(C) V jakém místě programu (okamžiku jeho běhu) dojde k uvolnění paměti, kterou zabírá instance třídy `Student` reprezentovaná v programu proměnnou `jan`, a kde dojde k uvolnění paměti instance struktury `Color` reprezentované proměnnou `pink`? Detailně vysvětlete proč. Pro obě varianty odhadněte, kolik bytů paměti se uvolněním instancí `Student`, resp. `Color` pravděpodobně získá, pokud bude program spuštěn na 32-bitové platformě.

**[1 bod]**

**Otázka č. 5**

Detailně vysvětlete, co to je tzv. *inlinování metod*, a kde se s ním v kontextu C#/.NET můžeme setkat. Kdo inlinování metod provádí a na základě rámcově jakých podmínek?

**[1 bod]****Otázka č. 6**

Předpokládejte následující třídu, kterou se nám nedaří přeložit:

```
class Prg6 {
    public ref int m(ref int a) {
        var x = 5;

        ref int r1 = ref x;

        return ref m(ref r1);
    }
}
```

Při pokusu o překlad nám C# překladač vrátí následující chybu (na červeně podtrženém a označeném místě):

```
Error CS8164: Cannot return by reference a result of 'Prg6.m(ref int)' because the argument passed to parameter 'a' cannot be returned by reference
```

Vysvětlete, co tato chyba znamená, a proč nám C# překladač uvedený kód nedovolí přeložit. Co by se mohlo obecně stát, kdyby uvedený kód překladač dovolil přeložit?

**[1 bod]****Otázka č. 7**

Následující program se bez chyb přeloží.

```
using System;

public class A {
    public int m(int a) => a + 1;
}

public class B : A {
    public new int m(int b) => b * 2;
}

public class C : B {
    public new int m(int c) => c / 10;
}

class Prg7 {
    static void Main(string[] args) {
        B b = new C();
        A a = b;
        Console.WriteLine(a.m(100));
    }
}
```

**(A)** Napište, co program po spuštění vypíše na standardní výstup a detailně vysvětlete proč.

**[1 bod]**

**(B)** Výše uvedený program se přeloží zcela bez chyb a warningů. Pokud ale z programu odstraníme tučně podtržené klíčové slovo new, tak se program opět přeloží bez překladových chyb (error), ale k danému řádku překladač vypíše warning:

```
Warning CS0108: 'B.m(int)' hides inherited member 'A.m(int)'. Use the new keyword if hiding was intended.
```

Vysvětlete, proč je absence klíčového slova new pouze warning a nikoliv překladová chyba (error). Navrhněte příklad scénáře, kde by reportování vynechání new jako error bylo nevhodné.

**[1 bod]**