

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 Insert standardní .NET třídy `List<T>`.

Otázka č. 1 (X)

Předpokládejte, že metodu `List<T>.Insert` standardní knihovny implementujeme a v rámci implementace kni chceme napsat sadu unit testů – s využitím testovacího frameworku MSTest, NUnit nebo xUnit napište sadu alespoň 3 vhodných unit testů testujících metodu `List<T>.Insert`.

[1 bod]

Společná část pro otázky označené Y

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

```
/// <summary>
/// Inserts item on selected index or on the end of
/// the list.
/// </summary>
/// <typeparam name="T">Type T</typeparam>
/// <param name="list">List</param>
/// <param name="index">Index</param>
/// <param name="item">Item</param>
static void InsertOrAddToEnd<T>(
    List<T> list, int index, T item
) {
    if (index < 0)
        throw new IndexOutOfRangeException();
    try {
        list.Insert(index, item);
    } catch {
        list.Insert(list.Count, item);
    }
}
```

Otázka č. 2 (X, Y)

Je vzhledem k výše implementovanému chování metody `InsertOrAddToEnd` dostatečný (vhodně formulovaný) připojený dokumentační komentář? Detailně vysvětlete proč. Pokud je vaše odpověď ne, tak navrhnete jeho lepší znění (ideálně anglicky, ale je možné i česky/slovensky).

[1 bod]

Otázka č. 3 (X, Y)

Je uvedená implementace metody `InsertOrAddToEnd` nejvhodnější? Detailně vysvětlete proč. Pokud je vaše odpověď ne, tak napište, jak by měla být naprogramována lépe.

[1 bod]

Otázka č. 4

Předpokládejte níže uvedenou třídu, která se bez chyb přeloží:

```
class Prg4 {
    public static string GetInfoFor(
        string name, int age
    ) {
        return $"Name: {name} Age+1: {age + 1}";
    }
}
```

(A) Detailně vysvětlete, který všechen kód bude schopen volat metodu `GetInfoFor` (pozor na viditelnost třídy `Prg4`). Není chybou, že třída není označena jako `public`?

[1 bod]

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

[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 program, který se bez chyb přeloží:

```
using System;

public class X {
    public void m(int i) {
        Console.WriteLine("X.f(int)");
    }
}

public class Y : X {
    public void m(float f) {
        Console.WriteLine("Y.f(float)");
    }
    public void Test() {
        m(1);
    }
}

class Prg6 {
    static void Main(string[] args) {
        Y y = new Y();
        y.Test();
    }
}
```

Napište, co program po spuštění vypíše na standardní výstup, a detailně vysvětlete proč. Proč je jazyk C# navržen s tímto chováním?

[1 bod]

Otázka č. 7

Následující program se bez chyb přeloží.

```
using Co = System.Console;

abstract class A {
    public virtual void m() => Co.Write("A");
}

class B : A {
    public virtual void m(int i = 42)
        => Co.Write($"B-{i}");
}

class C : B {
    public override void m() => Co.Write("C1");

    public virtual void m(int j = 333)
        => Co.Write($"C2-{j}");
}

class D : C {
    public void m() => Co.Write("D");
}

class Prg7 {
    static void Main() {
        var x = new D();
        x.m();
        (x as B).m();
    }
}
```

(A) Napište, co program po spuštění vypíše na standardní výstup a detailně vysvětlete proč (jako součást vysvětlení nakreslete, jak budou vypadat tabulky virtuálních metod typů A, B, C a D, a na co budou jejich jednotlivé záznamy ukazovat).

[1 bod]

(B) Je v daném příkladu k něčemu užitečné označení třídy A jako `abstract`? Detailně vysvětlete, proč je možné přeložit třídu označenou jako `abstract`, přestože nemá žádné metody označené jako `abstract`.

[1 bod]

Otázka č. 8

Předpokládejte následující třídu, kterou se nám nedaří přeložit:

```
class Prg8 {
    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 'Prg8.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]