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 kód, který se bez chyb přeloží:

```
class Prg1 {
  class Person {
    public string Name { get; set; }
    public override string ToString() => Name;
  }

  static void MakeUpper(object[] array) {
    for (int i = 0; i < array.Length; i++) {
      array[i] = array[i].ToString().ToUpper();
  } }

  static void Main() {
    var people = new[] {
      new Person { Name = "Pavel" },
      new Person { Name = "Michal" }
    };
    MakeUpper(people);
    Console.WriteLine(people[0]);
} }
```

Napište, co program vypíše, a detailně vysvětlete proč.
[1 bod]

Otázka č. 2

Detailně vysvětlete, proč je API třídy `Monitor` navržené tak, že pro volání metody `Monitor.Wait(…)` je třeba mít zamčený zámek stejného objektu, na kterém se pokoušíme čekat pomocí `Monitor.Wait`. Uveďte příklad situace, kde by mohlo dojít k problému, pokud by bylo možné volat `Monitor.Wait` bez předchozího volání `Monitor.Enter`.
[1 bod]

Otázka č. 3

Předpokládejte následující kód, který se bez chyb přeloží:

```
class Prg3 {
  static void Main() {
    var list = new List<int> {
      2, 4, 3, 1, 9, 5
    };

    list.OrderBy(x => x).Skip(2).Take(3);

    foreach (var item in list) {
      Console.WriteLine(item);
} } }
```

Napište, co program vypíše, a detailně vysvětlete proč.
[1 bod]

Otázka č. 4

Předpokládejte, že chceme naimplementovat následující třídu `MyCountdown`, jejíž základní API odpovídá standardnímu chování třídy `System.Threading.CountdownEvent`:

```
public class MyCountdown {
   public MyCountdown(int initialCount) {
   }

   /// <summary>Increments the countdowns's current
   /// count by one.</summary>
   public void AddCount() {
   }

   /// <summary>Registers a signal with the
   /// countdown, decrementing the value of current
   /// count.</summary>
   /// <returns>true, if the signal caused the
   /// count to reach zero and the event
   /// was set.</summary>
   public bool Signal() {
   }

   public void Wait() {
   }
}
```

Napište implementaci takové třídy `MyCountdown` bez použití `CountdownEvent` i bez použití `WaitHandle` a jejích libovolných potomků.
[1 bod]

Otázka č. 5

Immutable struktura `System.Drawing.Color` má pouze bezparametrický konstruktor a následující factory metody:

```
public static Color FromArgb(
  int red, int green, int blue
)
public static Color FromArgb(
  int alpha, int red, int green, int blue
)
```

a veřejné getter-only `byte` vlastnosti `A`, `R`, `G`, `B`.

Doplňte následující zdrojový soubor tak, aby se správně přeložil a výsledný program po spuštění vypsal (počítejte s tím, že struktura `Color` má již vhodně overridovanou metodu `ToString`) na standardní výstup `Color [A=105, R=217, G=0, B=0]`. **POZOR: máte zakázáno měnit tělo metody `Main`, její obsah musí zůstat přesně takový, jak je zapsáno níže**:

```
class Prg5 {
  static void Main() {
    Color c = 217.Red().WithAlpha(105);
    Console.WriteLine(c);
  }
}
```

[1 bod]

Otázka č. 6

Předpokládejte následující kód, který se bez chyb přeloží:

```
class B { public static void m()
  => Console.WriteLine(1); }

namespace A.A {
  class B { public static void m()
    => Console.WriteLine(2); }
}

namespace A {
  class B { public static void m()
    => Console.WriteLine(3); }

  class Prg6 {
    static void Main() {
      A.B.m();
} } }
```

Napište, co program vypíše, a detailně vysvětlete proč. Jak by se musela metoda `Main` upravit, aby se vypsaly i zbývající 2 varianty (tj. se zavolaly i zbývající 2 varianty metod `m`)?
[1 bod]

Otázka č. 7

Předpokládejte následující kód, který se bez chyb přeloží:

```
class Prg7 {
  static void Main(string[] args) {
    var list = new List<Action>();

    for (int i = 0; i < 2; i++) {
      int x = 0;
      for (int j = 0; j < 2; j++) {
        int y = 0;
        list.Add(() => Console.WriteLine(
          $"{i} = {++x} {++y}"
        ));
      }
    }
    foreach (Action a in list) a();
} }
```

Napište, co program vypíše, a detailně vysvětlete proč.
[1 bod]

Otázka č. 8

Předpokládejte následující třídu:

```
class X {
   private static Task f(int a) { ... }
   private static Task<int> g(int a, int b) { ... }
   private static int tx(int x) { ... }

   public static async Task<int> m(int x, int y) {
      f(x);
      int r = await g(x + 1, tx(y));
      return r * 2;
} }
```

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

Otázka č. 9

Předpokládejte následující program:

```
using C = System.Console;

interface IW {
    void m1(); void m2(); void m3();
}
class X : IW {
    public virtual void m1() { C.WriteLine(11); }
    public virtual void m2() { C.WriteLine(12); }
    public void m3() { C.WriteLine (13); }
}
class Y : X, IW {
    public virtual void m1() { C.WriteLine(21); }
    public override void m2() { C.WriteLine(22); }
    public virtual void m3() { C.WriteLine(23);  }
    void IW.m3() { C.WriteLine(-23);  }
}
class Z : Y {
    public override void m1() { C.WriteLine(31); }
    public override void m2() { C.WriteLine(32); }
    public override void m3() { C.WriteLine(33); }
}
class Prg9 {
    static void Main() {
        IW iw = new Z();
        X x = (X) iw;
        ((IW) x).m1();
        ((IW) x).m2();
        ((IW) x).m3();
}   }
```

Detailně vysvětlete (nakreslete tabulky VMT), co a proč vypíše po svém spuštění.
[1 bod]

Otázka č. 10

Předpokládejte následující kód, který se bez chyb přeloží:

```
public static IEnumerable<int> Range(
   int from, int to
) {
   if (to < from) {
      throw new ArgumentException(
        $"'{nameof(from)}' has to be less"
        + $" or equal than '{nameof(to)}'."
      );
   }
   for (int i = from; i <= to; i++) {
      yield return i;
}  }
```

Byla by výše uvedená metoda vhodnou implementací standardní metody `Enumerable.Range`? Detailně vysvětlete proč. Pokud odpovíte ne, tak napište vhodnější implementaci.
[1 bod]