DN8b

Zürcher Hochschule für Angewandte Wissenschaften

Arbeitsblatt: DNET1

Name: _____ **Kurznamen:** _____

1. Reflection - Copy Corresponding

Die Aufgabe Daten von einem Object in ein anderes zu übertragen tritt häufig auf. In der Regel werden dafür einzelne Feldwerte aus dem ersten Objekt gelesen und dann dem Feld des zweiten zugewiesen.

```
o2.value = o1.value
```

Falls jedoch viele Felder übertragen werden müssen, wird der Code unübersichtlich und es können sich auch leicht Fehler einschleichen, wenn Felder vergessen gehen, wenn z.B. die Klassen nachträglich erweitert werden.

Ein Spezialfall ist, wenn es sich zwar um Objekte unterschiedlicher Klassen handelt, deren Felder aber den gleichen Namen haben: die beiden Klassen haben also korrespondierende Felder.

```
class Source {
    public string S1;
    public int I1;
    public int X;
}

class Target {
    public string S1;
    public int I1;
    public string X;
}
```

Für diesem Fall kann mittels Reflection eine Methode `copyCorresponding` geschrieben werden, die das Übertragen sämtlicher Werte übernimmt, sofern der Typ passt.

```
void copyCorresponding(Object source, Object target) {
```

Aufgabe 1

Vervollständigen Sie die Methode copyCorresponding wie oben beschrieben.

Abgabe:

Praktikum: DN8.3

Datei: CopyCorresponding.cs

2. Common Intermediate Language

Es kann auch direkt Code in der sogenannten CIL geschrieben und in ausführbare Programme übersetzt werden. Das Übersetzungsprogramm `ilasm` ist der .NET Framework Installation i.d.R. beigefügt. Der Instruktionssatz ähnelt einem traditionellen Assembler. Der generierte intermediate Code wird beim Laden durch den JIT in Maschinencode übersetzt.

Aufgabe 2

Sie haben folgendes einfaches C# Programm zur Berechnung der Fakultät gegeben.

```
using System;
public class Factorial {
    public static void Main(string[] s) {
        int n = 5;
        int k = 1;
        while (n > 1) {
            k *= n; n--;
        }
        Console.WriteLine(k);
    }
}
```

Schreiben Sie dasselbe Programm direkt in CIL.

Hinweise:

- Obiges übersetztes Programm mittels `ildasm` in CIL umwandeln und versuchen zu verstehen bzw. als Grundlage für Ihr eigenes Programm nehmen.
- Sie werden sehen, dass der Compiler nicht optimalen Code erstellt (z.B. NOPs einfügt); optimieren Sie den Code.
- Linux mono:
  - installation: `sudo apt install mono-complete`
  - compiler: `mcs <filename>.cs`
  - assembler: `ilasm <filename>.il`
  - disassembler: `monodis <filename>.exe`
  - execute: `mono <filename>.exe`

- Linux/Windows .net core: `<filename>.ilproj`

```
<Project Sdk="Microsoft.NET.Sdk.IL/6.0.0">
  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net6.0</TargetFramework>
    <ProduceReferenceAssembly>false</ProduceReferenceAssembly>
  </PropertyGroup>
</Project>
```

- Online `https://sharplab.io/`

Abgabe:

Praktikum: DN8.4
Datei: Factoral.IL