Ambientes Virtuais de Execução

(2º Semestre 2010/2011)

2ª Aula Prática

1. Considere os ficheiros presentes na directoria **Questao1** do zip anexo a este enunciado. Compile os ficheiros e analise, recorrendo à ferramenta ildasm, o IL produzido.

2. Considere o seguinte programa:

```
using  System;
public struct Ponto {
  public int x, y;

  public static void Main() {
    Ponto p;
    p.x = p.y = 1;
    Object o = p;
    p = (Point) o;
    p.x = 2;
    o=p;
  }
}
```

Indique quais as operações de *box* e *unbox* realizadas no método Main, justificando a sua existência.

3. Considere o seguinte programa:

```
using System;
public struct A { public int x; }
public struct C :A { public int y; }
public class B { public int w; }
public class D : B { public int z;}
public class Program{
  public static void Main(){
    A a;
    Object o1 = a;
    Object o2 = a;
    B b= new D();
    D d= new B();
    o1.x = 2;
    ( (A) o2 ). x = 4;
     A k = o2;
    ((D) b).z=3;
    Console.WriteLine(a);
    Console.WriteLine(k);
    Console.WriteLine(o1);
    Console.WriteLine(o2);
    Console.WriteLine(b);
    Console.WriteLine(d);
 }
}
```

a) Altere, justificando, o que for necessário para que deixem de existir erros de compilação.
b) Quais foram as conversões e coersões realizadas?

4. Considere o seguinte programa:

```
using System;

class B1{
  public override bool Equals(object o){
        B1 b1 = o as B1;
        if(b1==null) return false;
        return true;
  }
}
class B2{
  public override bool Equals(object o){
    if(o ==null || o.GetType() != typeOf(B2) ) return false;
        return true;
  }
}
class B3 {
  public override bool Equals(object o){
    if(o ==null || o.GetType() != GetType() ) return false;
        return true;
        }
}
class B11 : B1{   }
class B22 : B2{   }
class B33 : B3{   }
class Program{
  public static void Main(){
   B11 a =new B11(); B1 b = new B1();
   B22 c = new B22(); B2 d = new B2();
   B3 e = new B3(); B33 f = new B33();
   Console.WriteLine(a.Equals(b)); Console.WriteLine(c.Equals(d)); Console.WriteLine(e.Equals(f));
  }
}
```

a) Altere, justificando, o que for necessário para que deixem de existir erros de compilação.
b) Qual é a implementação do método Equals que está correcta? Justifique.

5. Considere o seguinte programa

```
using System;
struct S{
        public int x;
}
class Program{
  public static void Main(){
     S a=new S();
     a.x=1;
     S b=new S();
     b.x=1;
     Console.WriteLine(a==b);
     Console.WriteLine(a.Equals(b));
     Console.WriteLine(ReferenceEquals(a,b));
  }
}
```

a) Altere o que for necessário para que deixem de existir erros de compilação.
b) Comente o output produzido.

6. Considere o seguinte programa:

```
using System;
class A{
  public  void  F(){ Console.WriteLine("A.F"); }
  public virtual void G(){ Console.WriteLine( " A.G");}
 }
class B: A{
 public void F(){ Console.WriteLine("B.F");}
  public override void G() { Console.WriteLine("B.G");}
   }
class Test{
static void Main(){
  B b = new B();
  A a = new B();
  a.F();
  b.F();
  a.G();
  b.G();
  }
}
```