

## Zkouška na test – pro samostatné procvičení

Navrhněte třídu **A**, která bude obsahovat:

- Vlastnost jen pro čtení `int` `cislo`
- Parametrický konstruktor, který nastaví `cislo` na hodnotu 10
- Virtuální metodu, která vytiskne na obrazovku jméno třídy (**A**)

Od třídy **A** odvodte třídu **B**, která bude obsahovat:

- Implementaci virtuální metody – taktéž vytiskne jméno třídy (zde **B**)

V hlavním programu:

- vytvořte instanci obou tříd
- na konzoli vytiskněte hodnotu vlastnosti `Cislo` jedné libovolné instance

### Třída A:

```
using System;
using System.Collections.Generic;
using System.Text;

class A
{
    private int cislo; // Privátní členská proměná

    // Kontruktor třídy A
    // Obsahuje nějaký parametr
    public A(int nejakeCislo)
    {
        this.cislo = 10; // přiřazení hodnoty 10 do proměné cislo
    }

    // Vlastnost Cislo
    public int Cislo
    {
        get { return cislo; } // Pouze pro čtení
    }

    // Virtuální metoda, která tiskne jméno třídy na obrazovku
    public virtual void TiskJmena()
    {
        Console.WriteLine("A"); // Tisk jména třídy
    }
}
```

#### Třída B:

```
using System;
using System.Collections.Generic;
using System.Text;

class B : A // Třída B je zděděná od třídy A
{
    // Konstruktor třídy B
    public B(int nejakeCislo)
        : base(nejakeCislo) // Zavolání konstruktoru bázové třídy
    {
        // Tady se nic dělat nemusí...
    }

    // Metoda, která vytiskne jméno třídy na obrazovku
    public override void TiskJmena()
    {
        Console.WriteLine("B"); // Tisk jména
    }
}
```

#### Hlavní program:

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Test
{
    class Program
    {
        static void Main(string[] args)
        {
            A tridaA = new A(20); // Vytvoření instance třídy A
            B tridaB = new B(1234); // Vytvoření instance třídy B

            Console.WriteLine(tridaA.Cislo); // Tisknutí Císla z tridaA
        }
    }
}
```