

# C# - Generic Collections



# Co jsou to kolekce ?



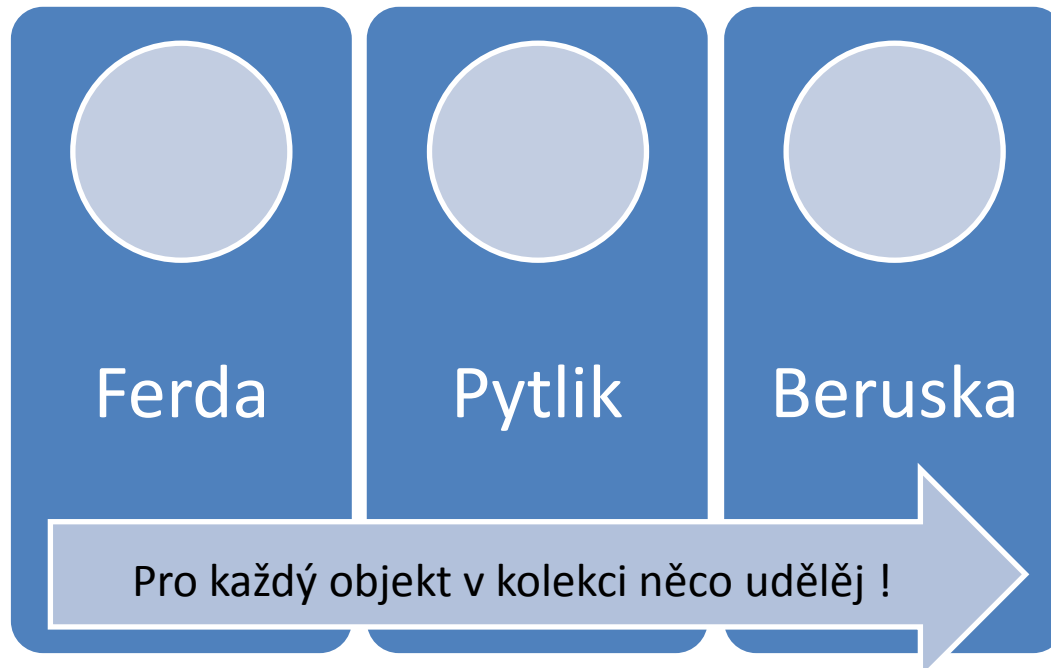
- Pro ukládání různých druhů objektů můžete využít Vám dobře známé pole **Array**
- Knihovna **System.Collections** ,ale nabízí daleko širší spektrum dynamických datových struktur pro ukládání více objektů např.:
  - Seznamy
  - Hašovací tabulky
  - Slovníky (*Dictionary*)
  - Stromy

[Indian tutor explaining the Collections in C#](#)  
[ArrayList explained on web functionx](#)

# Proč ?



- Máme 3 objekty typu Customer  
Customer Ferda = new Customer („Ferda“,...)  
Customer Pytlík = new Customer („Pytlík“,...)  
Customer Beruska = new Customer(„Beruska“,...)



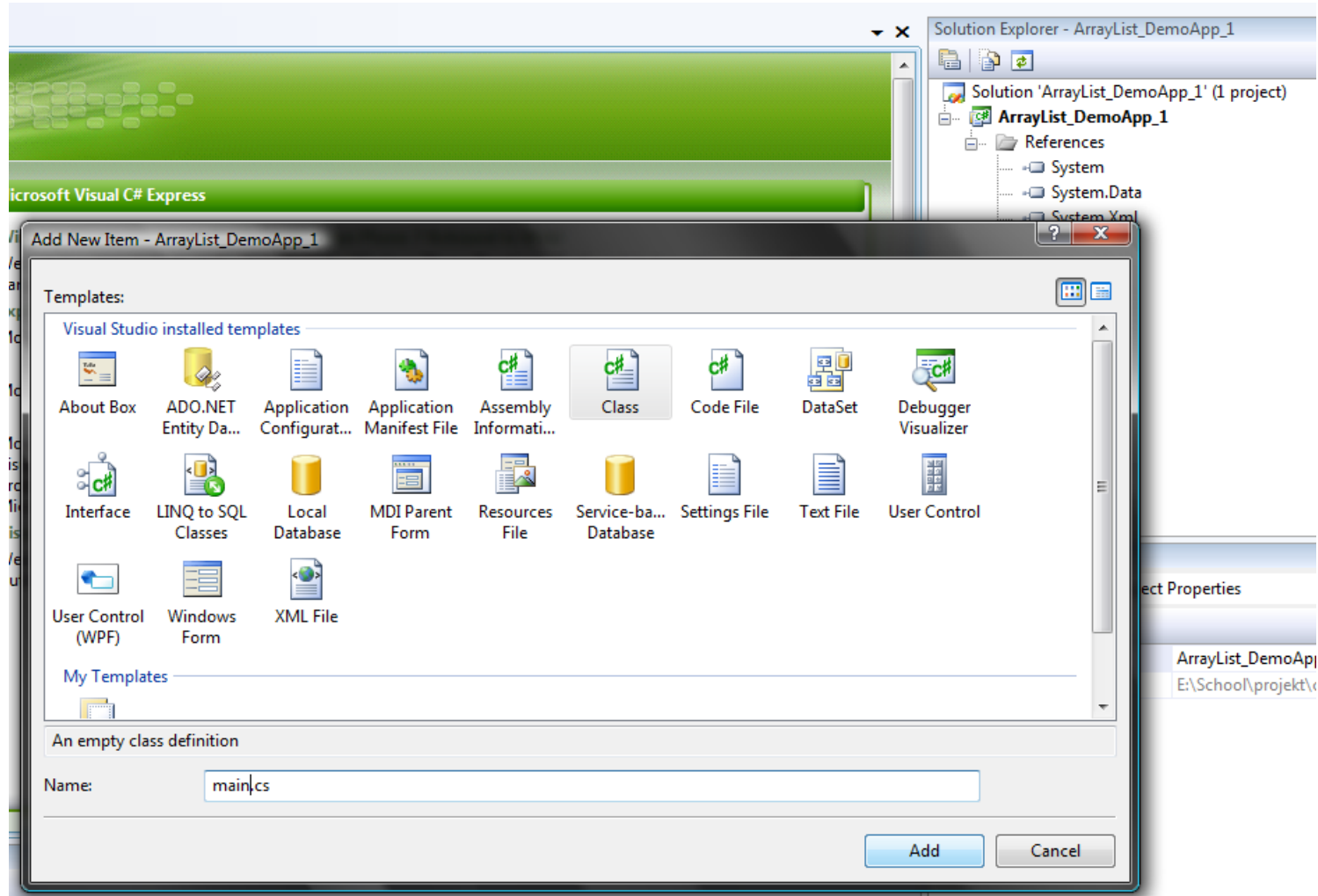
# ArrayList Step by Step application



- Návod na vytvoření aplikace najdete zde [Step by Step Array List demo](#)
- Vytvořte nový projekt typu **Empty project** a uložte ho pod názvem: **ArrayList\_demoApp\_1**

# Step 1

- Do projektu přidejte **class** a nazvěte ho **main**



# Step 2



- Projekt bude využívat knihovnu pro práci s kolekcemi

```
main.cs* Start Page Object Browser
Project1.main
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
//přidáme knihovnu pro práci s kolekcemi
//Namespace, který obsahuje i ArrayList
using System.Collections;

namespace Project1
{
    class main
    {
    }
}
```

# Step 3



- Vložíme metodu **Main** – vstupní bod naší aplikace a nadeklarujeme a nadefinujeme objekt typu **ArrayList**, který nazveme **myList**

```
//přidáme knihovnu pro práci s kolekcemi
//Namespace, který obsahuje i ArrayList
using System.Collections;

namespace Project1
{
    class main
    {
        static void Main(string[] args)
        {
            ArrayList myList; //deklarace myList
            myList = new ArrayList(); //vytvoření myList (definice)
        }
    }
}
```

## Step 4



- Do seznamu přidáme prvek pomocí metody Add a vypíšeme ho na konzoli.

```
static void Main(string[] args)
{
    ArrayList myList; //deklarace myList
    myList = new ArrayList(); //vytvoření myList (definice)

    //přidání hodnoty do seznamu
    myList.Add(1000);
    //vypsání prvku seznamu na konzoli
    Console.WriteLine("Prvek s indexem 0 v seznamu je " + myList[0]);
    Console.Read();
}
```





```
file:///E:/School/projekt/c sharp/Priklady/ArrayList_DemoApp_1/ArrayList_DemoApp_1/bin/Debug/ArrayList_DemoApp_1.EXE
Prvek s indexem 0 v seznamu je 1000
```

# Step 5



- Přidáme několik dalších prvků, setřídíme seznam a vypíšeme ho

```
//přidání hodnoty do seznamu
myList.Add(1000);
//vypsání prvku seznamu na konzoli
Console.WriteLine("Prvek s indexem 0 v seznamu je " + myList[0]);
Console.Read();

myList.Add(0);
myList.Add(100);
myList.Add(10000);
myList.Add(10);
myList.Add(1000);
//setřídíme seznam
myList.Sort(); //setřídí seznam sestupně

//vypíšeme obsah našeho seznamu
//Property Count udává počet prvků v našem seznamu
for (int i = 0; i < myList.Count; i++)
{
    Console.WriteLine(i + " : " + myList[i]);
}
Console.Read();
```



```
ca. file:///E:/School/projekt/c sharp/Priklady/ArrayList_DemoApp_1/ArrayList_DemoApp_1/bin/Debug/ArrayList_DemoApp_1.EXE
Prvek s indexem 0 v seznamu je 1000
0 : 0
1 : 10
2 : 100
3 : 1000
4 : 1000
5 : 10000
```

## Step 6



- Vymažeme všechny prvky seznamu pomocí metody **Clear**

```
//vypíšeme obsah našeho seznamu
//Property Count udává počet prvků v našem seznamu
for (int i = 0; i < myList.Count; i++)
{
    Console.WriteLine(i + " : " + myList[i]);
}

myList.Clear(); //metoda vymaže všechny prvky seznamu a uvolní paměť
myList.Add(100); //toto je teď jediný prvek našeho seznamu, který stále existuje

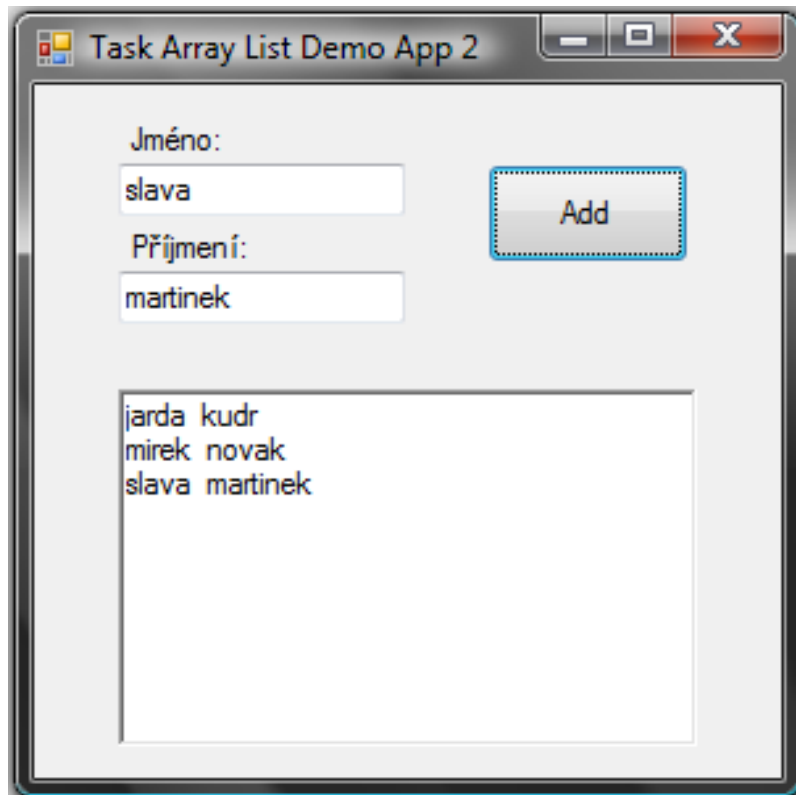
Console.Read();
```



**Konec Step by Step Demo aplikace 1**

# Úkol

- Naprogramujte vlastní formulářovou aplikaci, která bude využívat *ArrayList*



- Vytvořte **class** *Person*, který bude mít dvě property *FirstName* a *SurName*
- Na tlačítko **Add** se vytvoří instance objektu *Person* s údaji zadanými v textbozech aplikace, tato osoba se přidá do seznamu a celý seznam se vypíše do *RichTextboxu* tak, jak to vidíte na obrázku.
- Pro výpis seznamu použijte příkaz *foreach*

# Step by Step Demo Application 2

## Collections

### ArrayList

### Shopping Basket



# Aplikace Shopping basket



- Video prezentaci na vytvoření vzorové aplikace najdete zde :

[Shopping basket Demo App - video](#)

- Vstupní znalosti potřebné pro vytvoření aplikace:
  - Práce s objekty
  - Dědičnost
  - Práce se seznamy – viz. předchozí úkol
- Co se naučíte ?
  - Vytvořit aplikaci, která umí do seznamu ukládat různé objekty
  - Seřadit seznam podle různých parametrů a vytvořit vlastní srovnávací kritéria
  - Poznáte jinou strukturu pro ukládání objektů než je **ArrayList** a to je fronta - **Queue**