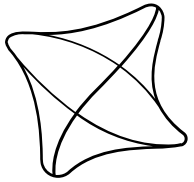
Jazyk C# a .NET

cvičenie

2



ŽILINSKÁ UNIVERZITA V ŽILINE
Fakulta riadenia
a informatiky

Práca s triedou

Ing. **Štefan Toth**, PhD.

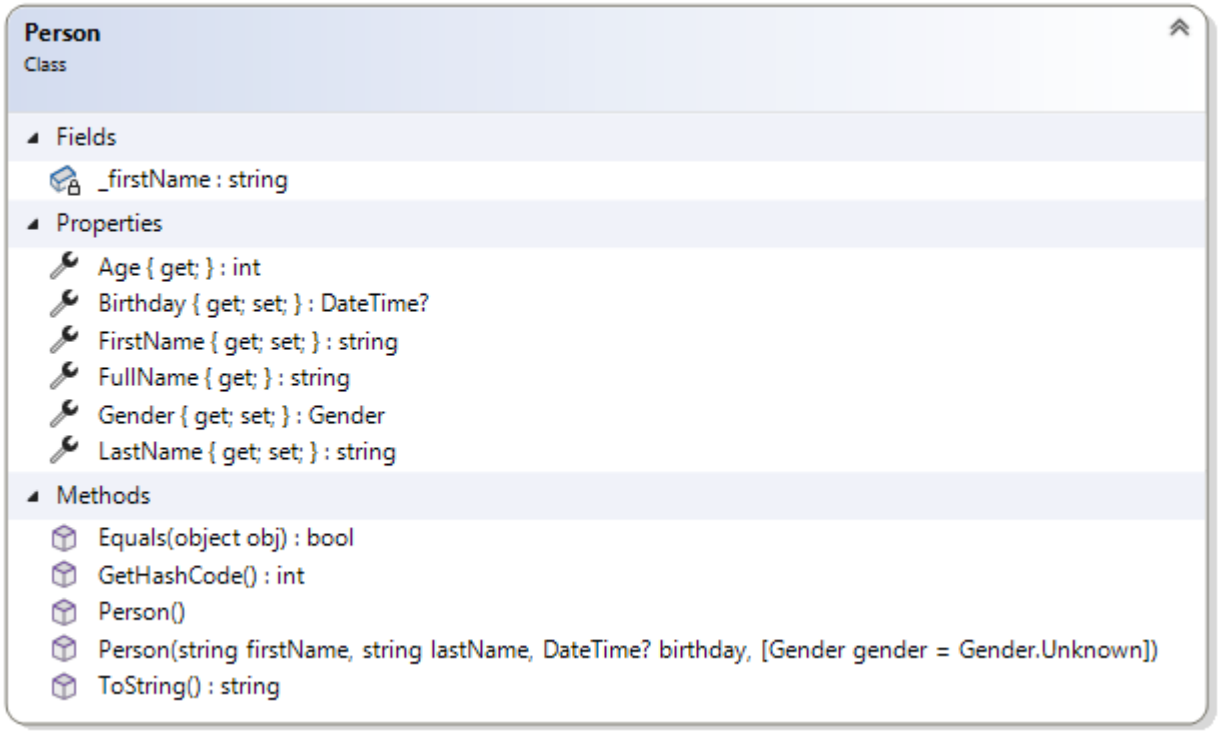
KST Katedra softvérových technológií
Fakulta riadenia a informatiky, Žilinská univerzita v Žiline

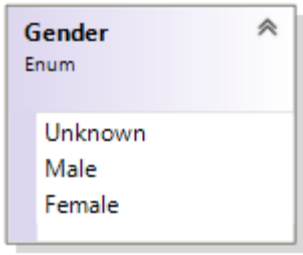
Obsah cvičenia

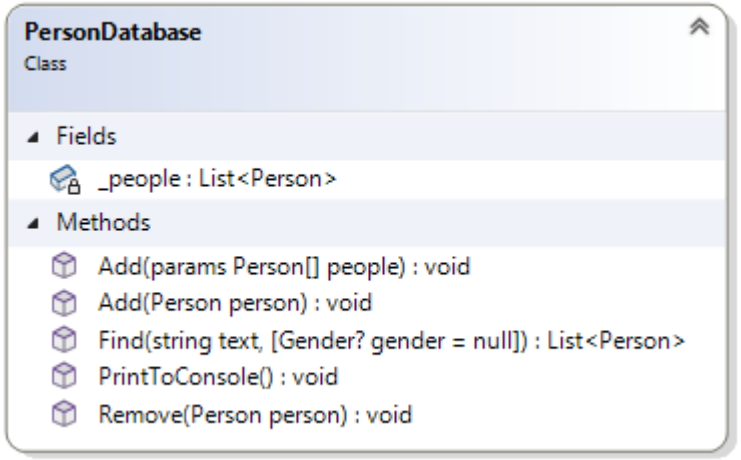
- Vytvorenie **triedy** (class)
- Používanie **vlastností** (properties)
- **Preťažovanie** (overloading) konštruktorov a metód
- **Voliteľné argumenty** (optional arguments) konštruktorov a metód
- **Prekrytie** (overriding) metód **ToString()**, **Equals()**, **GetHashchode()**

Úloha 1

- **Vytvorte konzolovú aplikáciu**, v ktorej implementujte postupne nasledujúce typy:





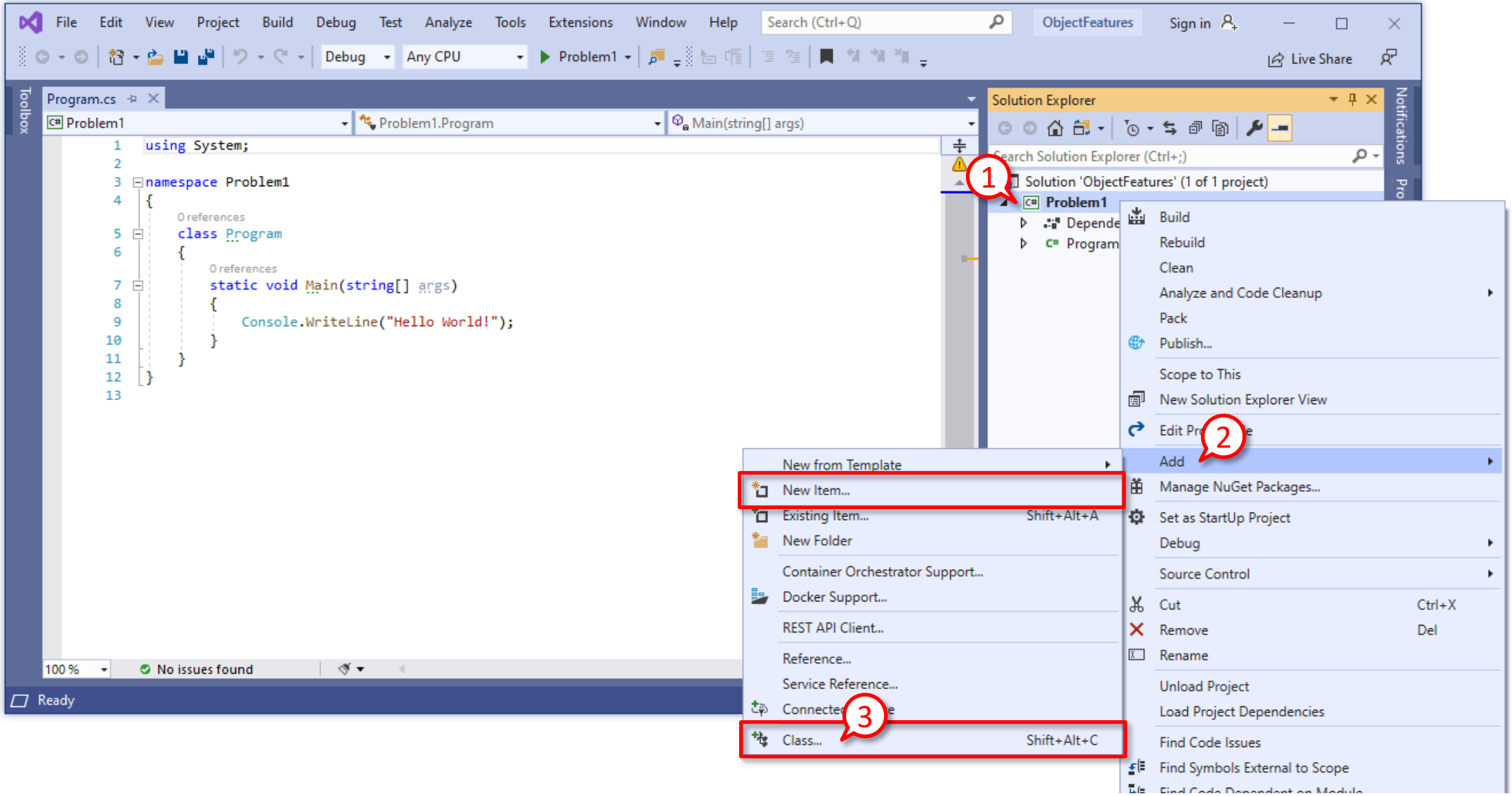


Úloha 1.1 (trieda Person)

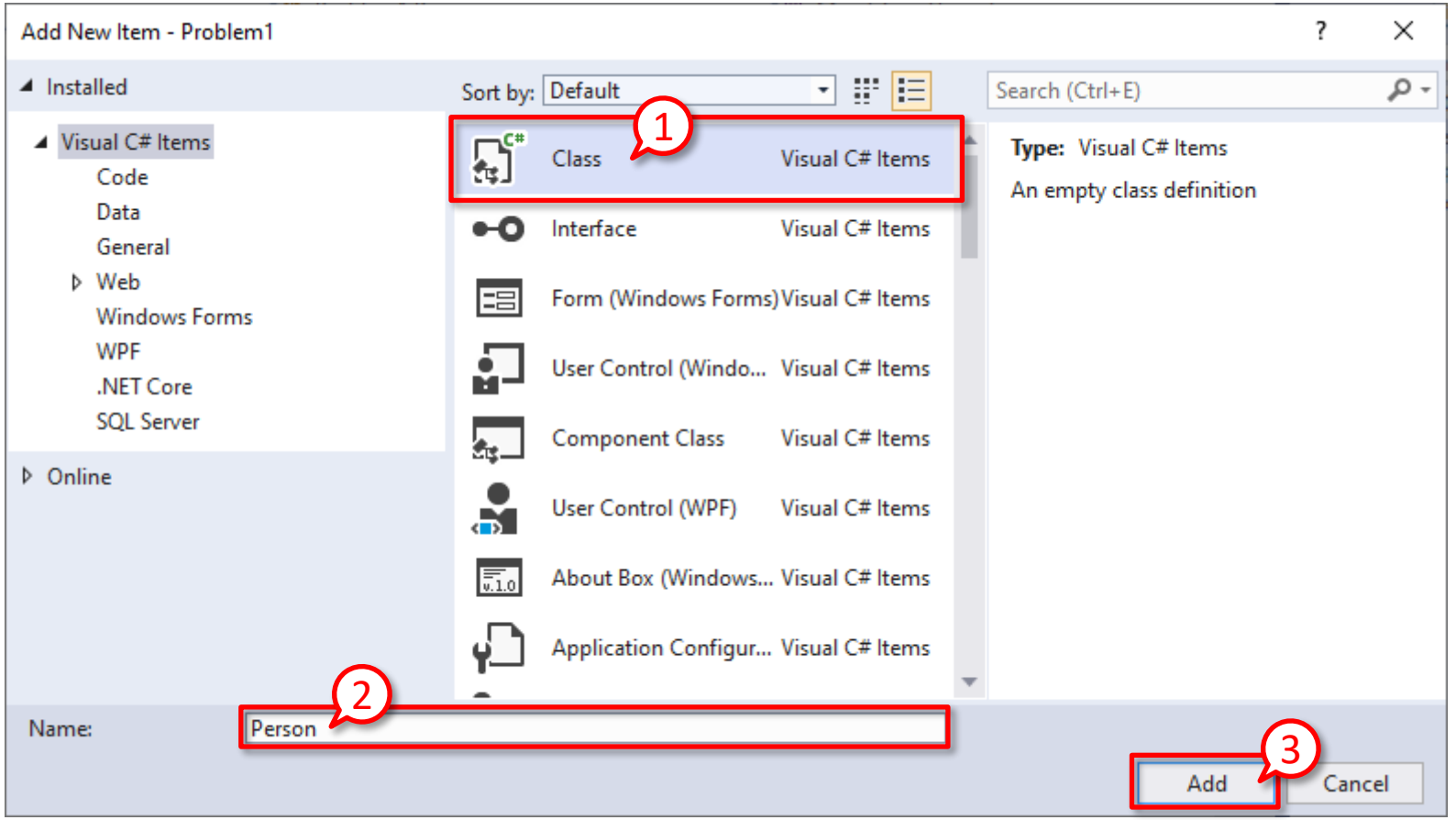
- V projekte **pridajte triedu Person** (Osoba), ktorá bude **obsahovať** tieto **vlastnosti** (property):
  - **FirstName** (Meno)
  - **LastName** (Priezvisko)
  - **FullName** (Meno a priezvisko) – len na čítanie bez dátového člena
  - **Birthday** (Dátum narodenia)
  - **Age** (Vek) – bude sa automaticky vypočítavať z dátumu narodenia a aktuálneho času

- Vyskúšajte si rôzne spôsoby tvorby vlastností (s dát. členmi, automatické, ...): https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/properties

Pridanie novej triedy (cez New Item... alebo Class...)



Pridanie novej triedy (Class – názov - Add)



Pridanie vlastnosti

- Snippet **prop** vytvorí v triede **automatickú vlastnosť**, tabulátorom sa prepíname medzi typom a názvom vlastnosti, ktoré môžeme modifikovať

```
1   using System;
2   using System.Collections.Generic;
3   using System.Text;
4
5   namespace Problem1
6   {
        0 references
7       class Person
8       {
            0 references
9           public int MyProperty { get; set; }
10      }
11  }
12
```

Príklady definovania vlastnosti FirstName

Vlastnosť (full property):

```
// Súkromný dátový člen (field)
private string _firstName;

// Verejná vlastnosť zaobalujúca prístup
k dátovému členu (staršia syntax):
public string FirstName
{
    get { return _firstName; }
    set { _firstName = value; }
}

// Alebo je možné jej get a set časť
definovať aj cez lambda operátor (=>)
public string FirstName
{
    get => _firstName;
    set => _firstName = value;
}
```

Automatická vlastnosť (auto property):

```
// Verejná vlastnosť na čítanie i zapisovanie
public string FirstName { get; set; }
```

```
// Verejná vlastnosť na čítanie, avšak
súkromná na zapisovanie (iba vo vnútri triedy)
public string FirstName { get; private set; }
```

```
// Verejná vlastnosť iba na čítanie,
nastavovať a zapisovať do nej sa dá iba v
konštruktore:
public string FirstName { get; }

// alebo aj priamo pri inicializovaní:
public string FirstName { get; } = "Jano";
```

Vlastnosti – rôzne zápisy (vývoj jazyka C#)

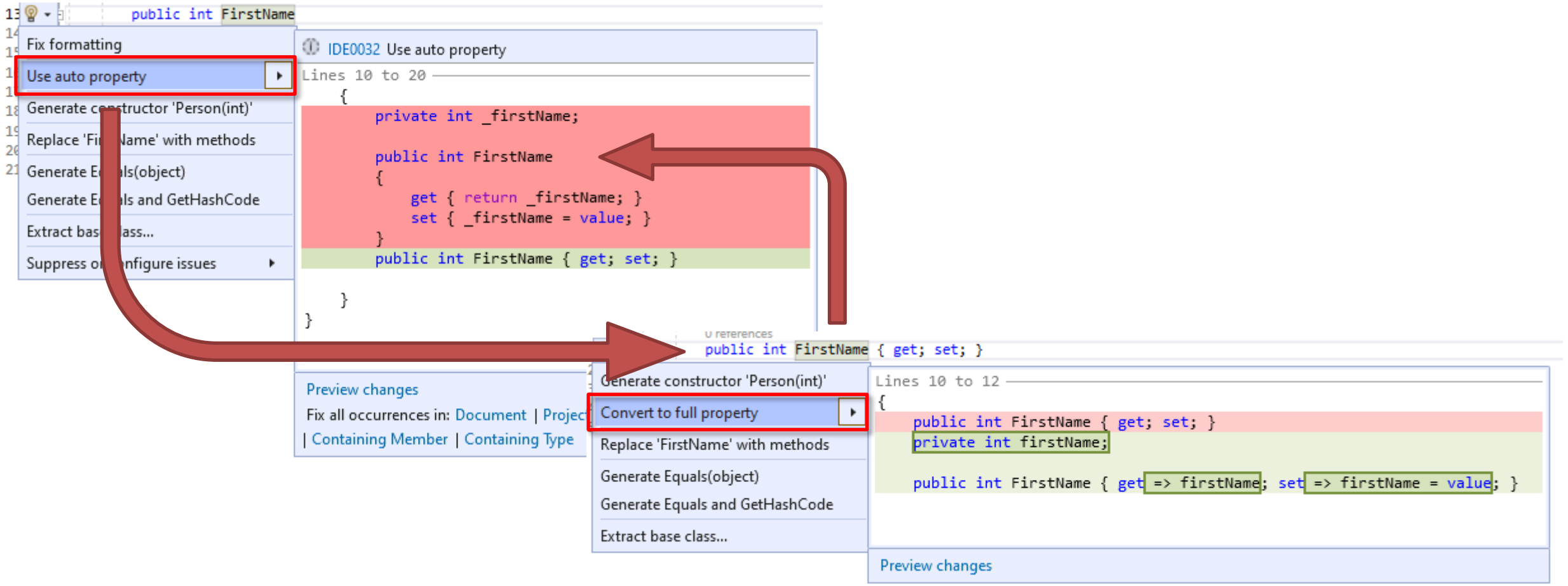
Vlastnosť FullName definovaná 3 rôznymi spôsobmi (všetky sú ekvivalentné):

```
// Súkromný dátový člen (field)
private string _fullName;

// (1. spôsob) Verejná vlastnosť
// iba na čítanie:
public string FullName
{
    get { return _fullName; }
}

// (2. spôsob) Cez lambda operátor (=>),
return a zátvorky { } sa vynechávajú
public string FullName
{
    get => _fullName;
}

// (3. spôsob) Vynechanie get
public string FullName => _fullName;
```

```
public string FirstName { get; set; }

public string LastName { get; set; }

// Bez použitia dátového člena _fullName, výsledkom
je reťazec zložený z mena a priezviska
public string FullName => $"{FirstName} {LastName}";
```

Rýchle akcie (Quick Actions)

- Klávesová skratka **Ctrl + .** (alebo **Alt + Enter**) nad identifikátorom vlastnosti:



Úloha 1.2 (vytvorenie inštancie Person)

- V metóde Main() **vytvorte ľubovoľný objekt typu Person**, napr.:

```
var jano1 = new Person();
jano1.FirstName = "Ján";
jano1.LastName = "Mrkvička";
jano1.Gender = Gender.Male;
```

alebo **pomocou inicializátora objektu** (volá konštruktor + nastavuje vlastnosti):

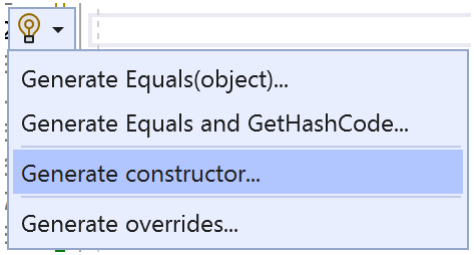
```
var jano2 = new Person { FirstName = "Ján",
                         LastName = "Mrkvička",
                         Gender = Gender.Male };
```

- Dokumentácia pre inicializátory: https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/object-and-collection-initializers

Úloha 1.3 (parametrický konštruktor)

- **Pridajte parametrický konštruktor**, ktorý bude inicializovať objekt s parametrami mena, priezviska, dátumu narodenia a pohlavia, aby ste mohli vytvoriť objekt jednoduchšie cez konštruktor:

```
var jano3 = new Person("Ján", "Mrkvička", new DateTime(1985, 12, 31), Gender.Male);
```



- Vytvorený objekt **osoby vypíšte na obrazovku** pomocou **formátovacieho reťazca** v tvare "{FullName}, {Birthday:dd.MM.yyyy}, age: ({Age}), gender: {Gender}"

Úloha 1.4 (voliteľný parameter a nullable)

- Upravte konštruktor tak, aby:
  - parameter ***pohlavie*** **bol voliteľný** a implicitne nastavený na neznáme pohlavie,
  - parameter ***dátum narodenia*** **mohol nadobúdať** ***null*** **hodnoty**, navyše ak bude parameter mať hodnotu *null*, nastavte vlastnosť na aktuálny dátum.

```
var jano4 = new Person("Ján", "Mrkvička", null);
```

- Ďalšie informácie v dokumentácii:
  - Pomenované a voliteľné argumenty: https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/named-and-optional-arguments
  - Nullable hodnotové typy: https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/builtin-types/nullable-value-types

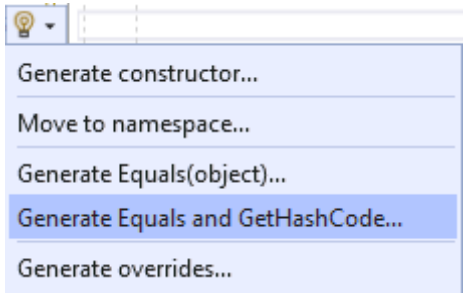
Úloha 1.5 (Equals)

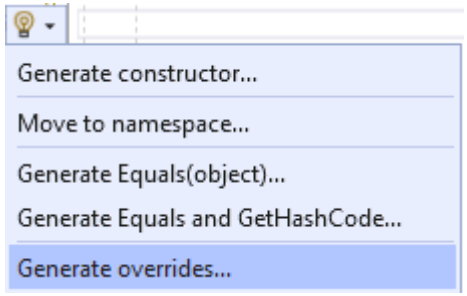
- Pridajte do triedy metódu **Equals** pre zistenie, či je objekt rovný s iným objektom:

```
var jano2 = new Person
{
    FirstName = "Ján",
    LastName = "Mrkvička",
    Gender = Gender.Male,
    Birthday = DateTime.Parse("31.12.1985", new CultureInfo("sk"))
};

var jano3 = new Person("Ján", "Mrkvička", new DateTime(1985, 12, 31), Gender.Male);

bool result = jano2.Equals(jano3); // result == False (Prečo?)
```

Úloha 1.6 (prekryte ToString, Equals, GetHashCode)

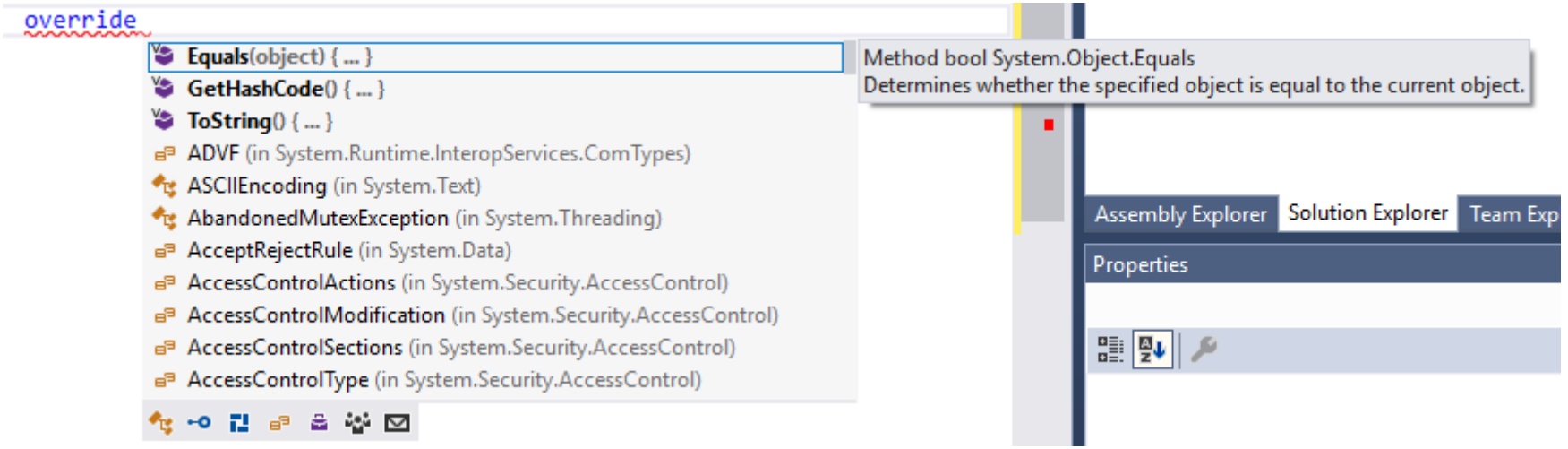
- **Prekryte** (override) v triede Person **nasledujúce metódy** predka `object` (`System.Object`):
  - **ToString()** – bude vypisovať osobu vo formáte „Meno Priezvisko (Vek)“
  - **Equals()** – bude porovnávať objekty typu Person (objekty budú rovné, ak sa zhodujú vo všetkých hodnotách vlastností). S metódou je nutné prekryť aj metódu **GetHashCode()**, ktorá vracia číslo identifikujúca objekty, najmä pre hash kolekcie
- Na vygenerovanie môžete použiť Visual Studio – na prázdnom riadku použite **Ctrl + .** (alebo **Alt + Enter**):



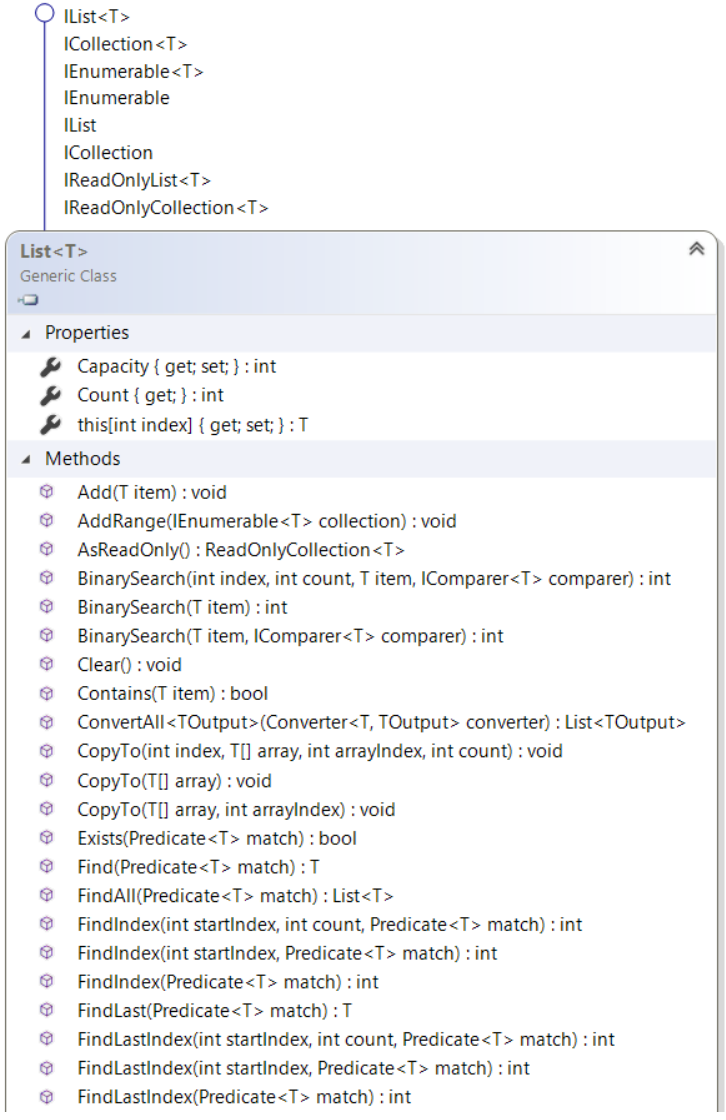


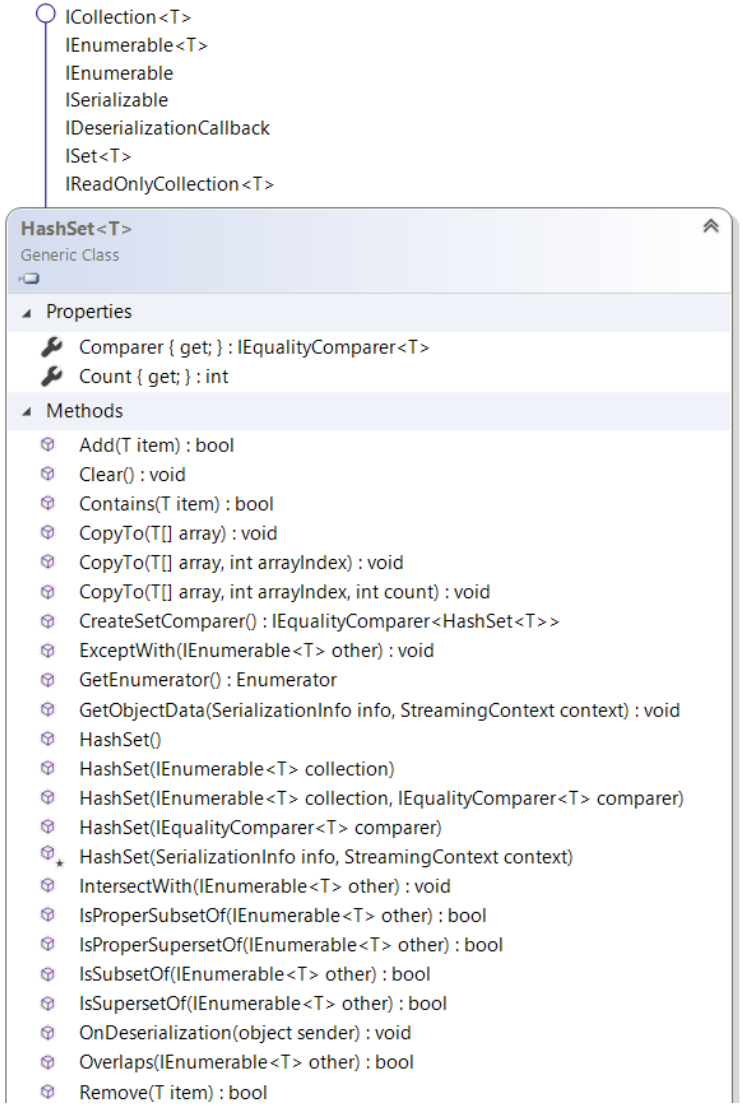
Prekrývanie (Visual Studio – IntelliSense)

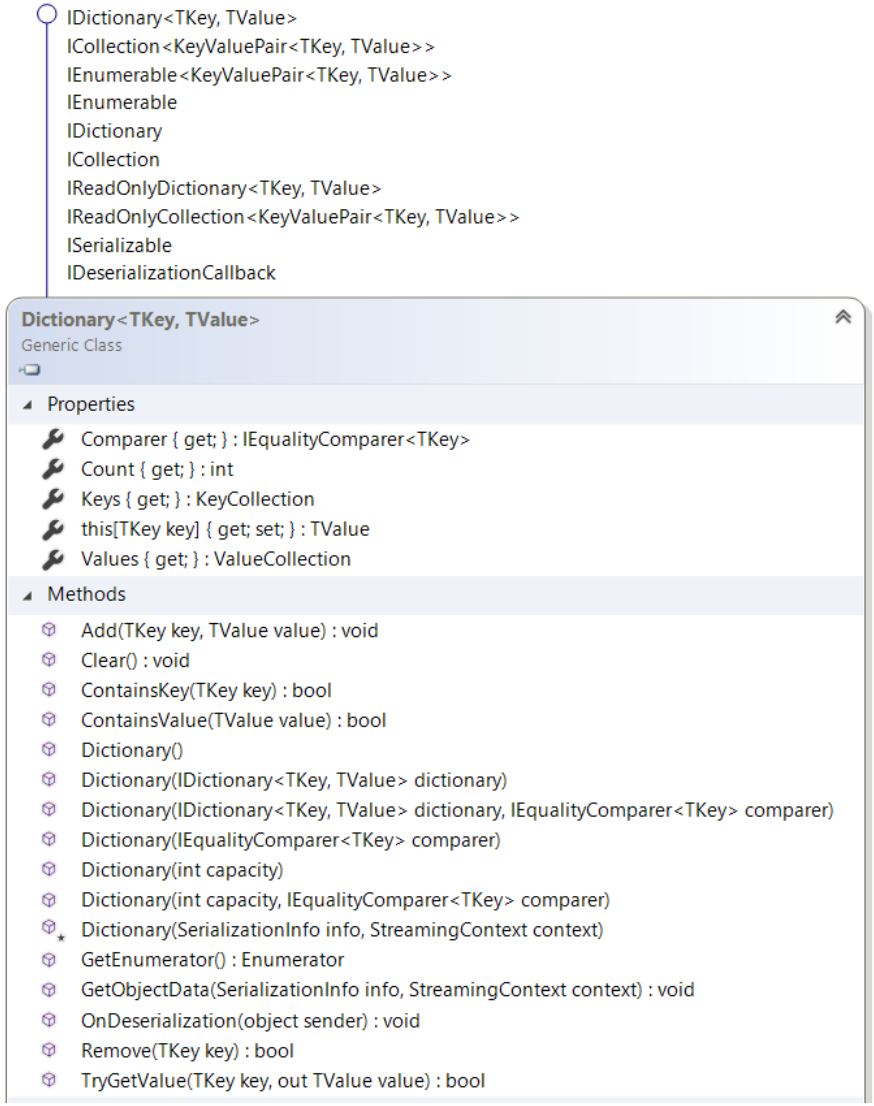
- Ďalším spôsobom, ako prekryť metódy, je napísať kľúčové slovo **override** v triede, IntelliSense následne zobrazí všetky možné metódy, ktoré je možné prekryť z predkov
- Príklad zobrazuje 3 metódy, ktoré sú virtuálne v predkovi (System.Object) a neboli ešte prekryté:



Zoznámte sa s najpoužívanejšími kolekciami

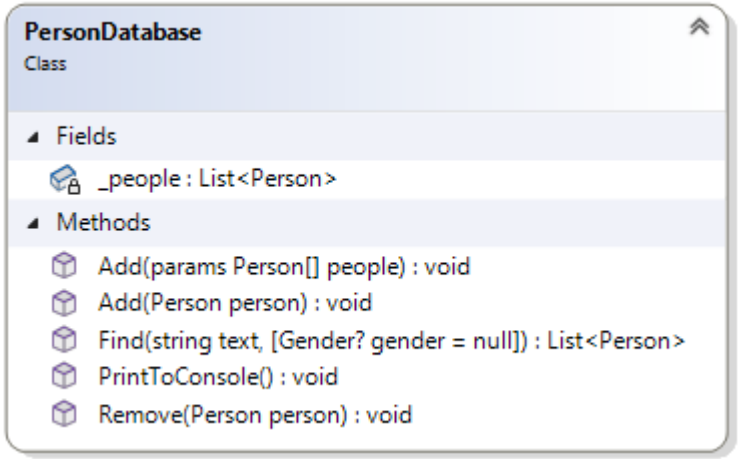






Úloha 1.7 (PersonDatabase)

- **Vytvorte** a implementujte **triedu PersonDatabase**:

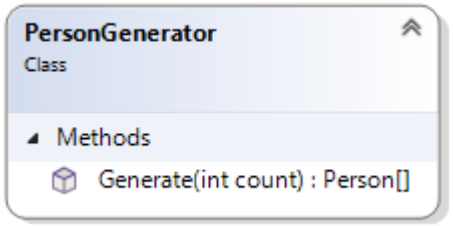


- Pre oboznámenie sa s kolekciami, môžete dátovy člen _people definovať s použitím kolekcie **HashSet<T>** alebo **Dictionary<T>** namiesto **List<T>**

Úloha 1.8 (PersonDatabase)

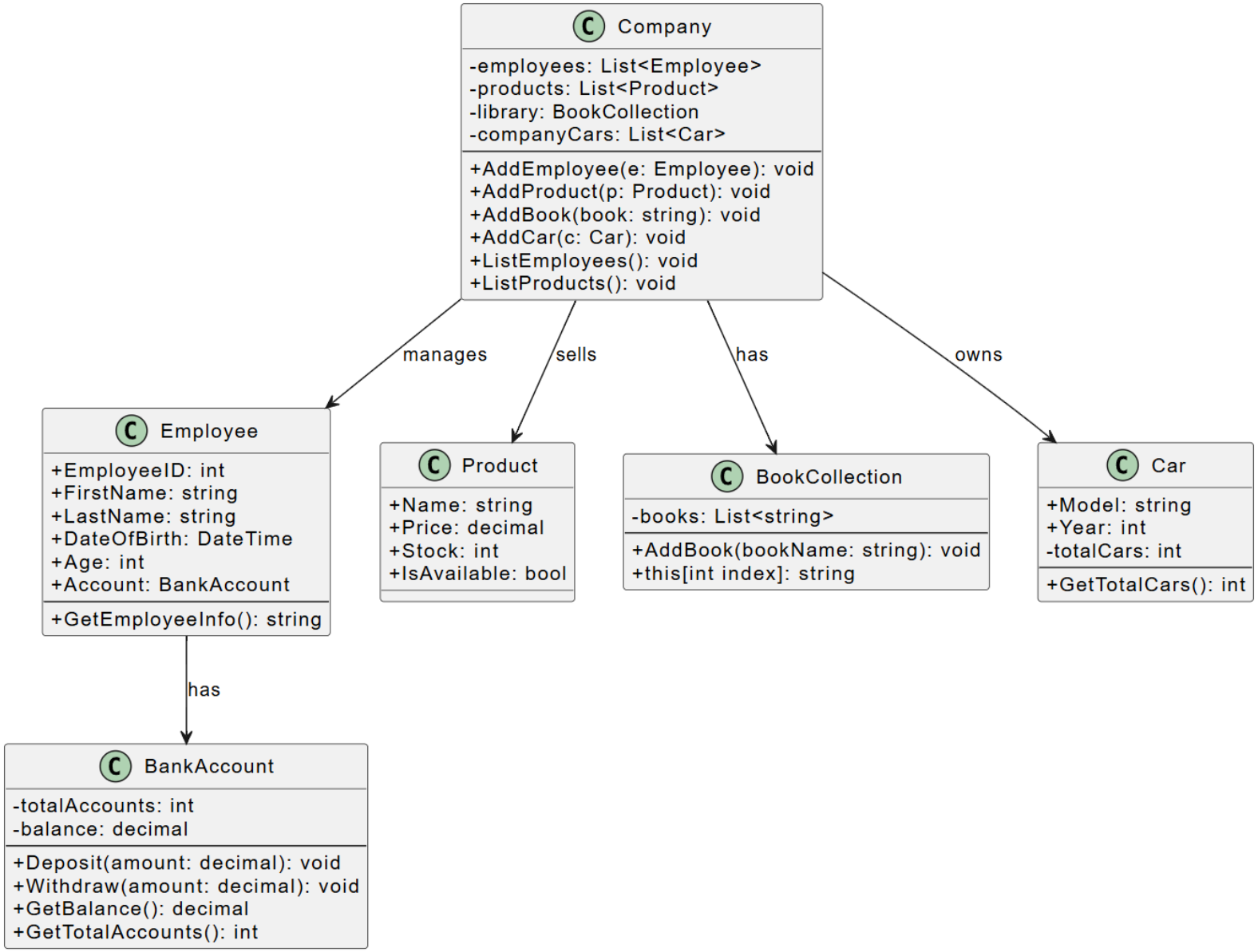
- Vytvorte triedu **PersonGenerator**, ktorá bude generovať osoby napr. z takýchto polí:

```
{ "Ján", "Alexander", "Adam", "Juraj", "Štefan" }
{ "Nový", "Malý", "Veľký", "Chudý", "Vysoký", "Bohatý", "Krásny" }

{ "Silvia", "Františka", "Michaela", "Barbora", "Eva" }
{ "Nová", "Malá", "Veľká", "Chudá", "Vysoká", "Bohatá", "Krásna" }
```



Úloha 2

- Vytvorte si nový projekt konzolovej aplikácie a implementujte jednotlivé triedy podľa zadania



Zdroje

- .NET Documentation
  https://learn.microsoft.com/en-us/dotnet/
- C# documentation
  https://learn.microsoft.com/en-us/dotnet/csharp/
- Visual Studio product family documentation
  https://learn.microsoft.com/en-us/visualstudio/

Upozornenie

- Tieto študijné materiály sú určené výhradne pre študentov predmetu **Jazyk C# a .NET** na Fakulte riadenia a informatiky Žilinskej univerzity v Žiline

- Reprodukovanie, šírenie (i častí) materiálov bez písomného súhlasu autora nie je dovolené

ŽILINSKÁ UNIVERZITA V ŽILINE
Fakulta riadenia
a informatiky

Ing. **Štefan Toth**, PhD.

stefan.toth@uniza.sk