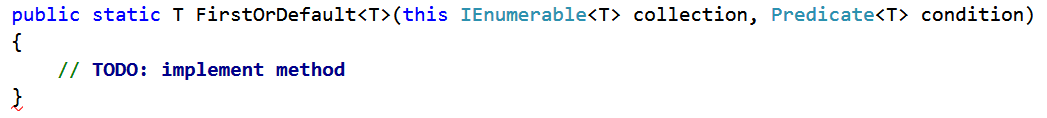
# Exercise: Delegates and Events

This document defines an in-class exercise from the ["OOP" Course @ Software University](https://softuni.bg/courses/oop/).

## Predicates

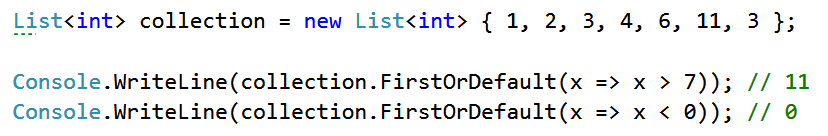
Task: Write a LINQ Extension method **FirstOrDefault(Predicate<T>)**. It should return the first element meeting the criterion provided or the default value of T if no such element exists.

**Hints**: Remember that extension methods are defined in a **static** class, the methods themselves are also **static** and the first parameter is the type you want to extend. LINQ extension methods extend **IEnumerable<T>** collections, so the signature of the method would look like this:



The first parameter is the type being extended (**IEnumerable<T>**), the second parameter is the condition, which is a **Predicate<T>**. In the method body you need to test each element in the collection and if you find one which meets the criterion – return it. If you don't find such element, return the default value of **T** – you can use **default(T)** to achieve this. Therefore, the return type of the method is **T**.

Example:



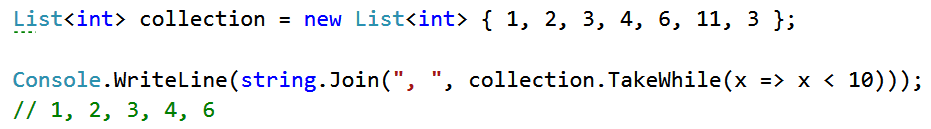
## Func

Task: Write a method **TakeWhile(Func<T, bool> predicate)** which takes elements from a collection while a condition is true.

**Func<T, bool>** is by nature a **Predicate<T>**. It takes an object of type **T** as an argument and returns true or false.

Example: In the collection int[] nums = { 1, 2, 3, 6, 4, 11 } the method **TakeWhile(x => x < 5)** would return an IEnumerable collection containing the elements 1, 2 and 3.

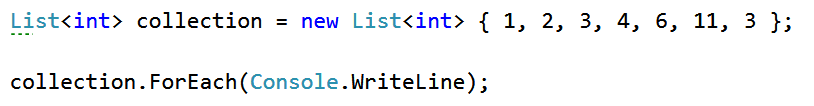
Example:



## Action

Task: Write a method **ForEach(Action<T>)** which takes an element of type T from a collection and applies some action to it. Actions are **void** methods – they don't have a return value. **Console.WriteLine** is an action, it prints on the console.

Example:



Result in the console:

