# Lab: Enumerations and Attributes

Problems for exercises and homework for the <https://softuni.bg/courses/csharp-oop-advanced-high-quality-code>.

You can check your solutions here: [https://judge.softuni.bg/Contests/Practice/Index/708#0](https://judge.softuni.bg/Contests/Practice/Index/708).

# Part I: Enumerations

## Weekdays

Create Enum Weekday with the days from **Monday** through **Sunday**.

Create a class WeeklyCalendar that should have at least the methods:

* void AddEntry(string weekday, string notes)
* IEnumerable<WeeklyEntry> WeeklySchedule

Create a class WeeklyEntrywhich should have constructor:

* WeeklyEntry(string weekday, string notes)
* Implement IComparable<WeeklyEntry> - Entries have to be compare first by their week day and then by note

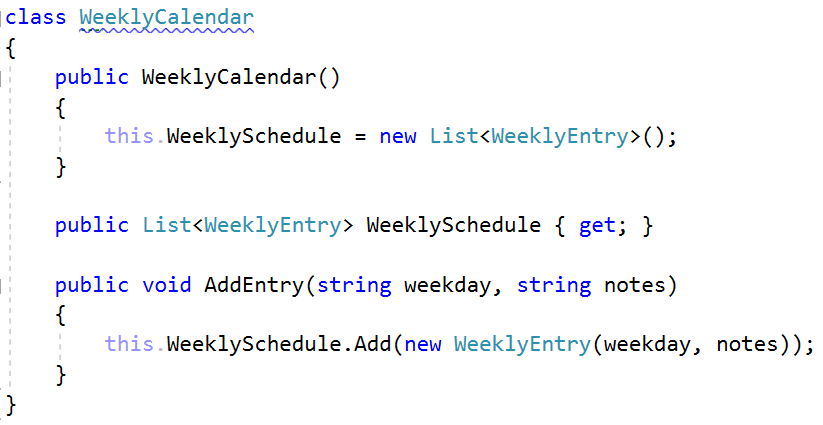
OverrideToString()ofWeeklyEntry – "{weekday} - {notes}" (e.g. "Monday - sport", "Sunday - sleep")

### Examples

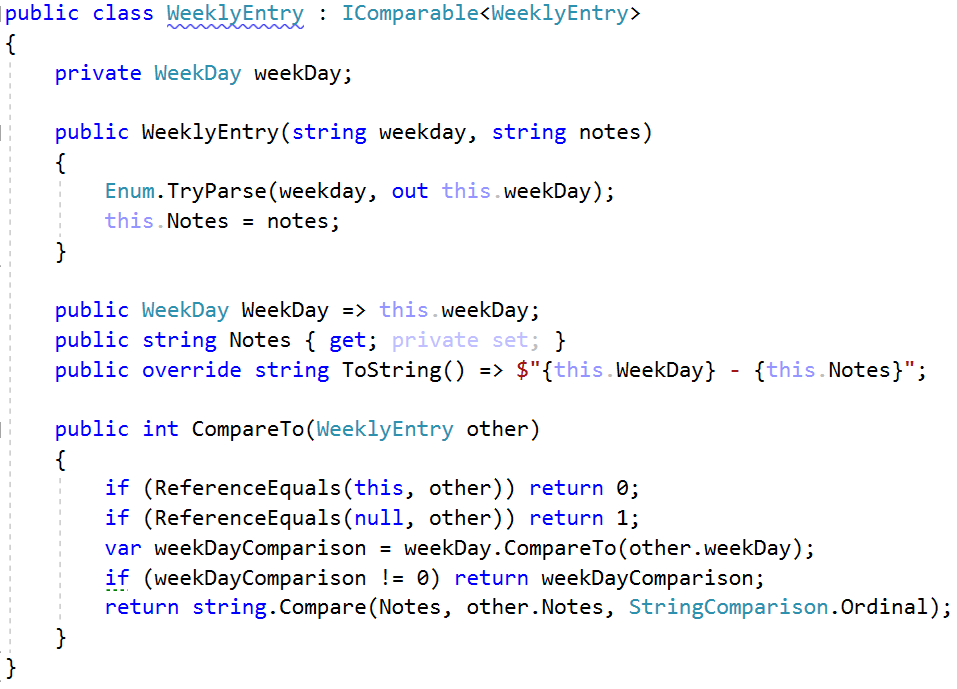
|  |
| --- |
| StartUp.cs |
| WeeklyCalendar calendar = new WeeklyCalendar();  calendar.AddEntry("Monday", "Internal meeting");  calendar.AddEntry("Tuesday", "Create presentation");  calendar.AddEntry("Tuesday", "Create lab and exercise");  calendar.AddEntry("Thursday", "Enum Lecture");  calendar.AddEntry("Monday", "Second internal meeting");  var ordered = calendar.WeeklySchedule.OrderBy(n => n).ToList();  foreach (var weeklyEntry in ordered)  {  Console.WriteLine(weeklyEntry);  } |

### Hints

Create the WeeklyCalendar:



Create WeeklyEntry and think about a solution to compare entries:



## Coffee Machine

Create a class CoffeeMachine, with methods:

* void BuyCoffee(string size, string type)
* void InsertCoin(string coin)
* IEnumerable<**CoffeeType**> CoffeesSold { get; }

Create enum CoffeeType - Espresso, Latte, Irish

Create enum Coin - 1, 2, 5, 10, 20, 50 (constants should be named One, Two, Five, etc.)

Create enum CoffeePrice - Small (50 c), Normal (100 c), Double (200 c)

CoffeeMachine should **clear all coins after each successful coffee sold**.

When you receive "End" command, print all sold coffees.

### Examples

|  |
| --- |
| StartUp.cs |
| CoffeeMachine machine = new CoffeeMachine();  string input;  while ((input = Console.ReadLine()) != "End")  {  string[] inputArgs = input.Split();  if (inputArgs.Length == 1)  {  machine.InsertCoin(inputArgs[0]);  }  else  {  machine.BuyCoffee(inputArgs[0], inputArgs[1]);  }  }  foreach (var coffeeType in machine.CoffeesSold)  {  Console.WriteLine(coffeeType);  } |

|  |  |
| --- | --- |
| **Input** | **Output** |
| Ten  Twenty  Twenty  Small Espresso  End | Espresso |
| Ten  Twenty  Small Espresso  Twenty  Small Espresso  End | Espresso  //Comment: first try - not enough coins |

# Part II: Attributes

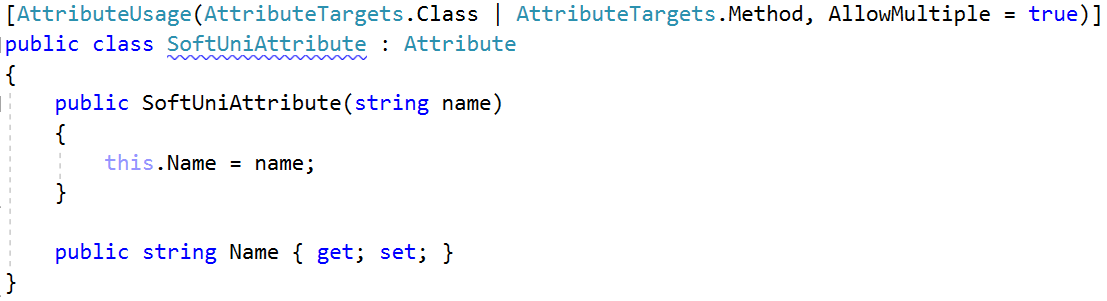
## Create Attribute

Create attribute SoftUni with a string element called **name**, that**:**

* Can be used over classes and methods
* Allow multiple attributes of same type

### Examples

|  |
| --- |
| StartUp.cs |
| [SoftUni("Ventsi")]  class StartUp  {  [SoftUni("Gosho")]  static void Main(string[] args)  {  }  } |



## Coding Tracker

Create a class **Tracker** with a method:

* static void printMethodsByAuthor()

### Examples

|  |
| --- |
| StartUp.cs |
| [SoftUni("Ventsi")]  class StartUp  {  [SoftUni("Gosho")]  static void Main(string[] args)  {  var tracker = new Tracker();  tracker.PrintMethodsByAuthor();  }  } |

