**Exercise: Interfaces and Abstraction**

Problems for exercise and homework for the ["C# OOP" course @ SoftUni"](https://softuni.bg/trainings/3585/csharp-oop-february-2022).

You can check your solutions here: <https://judge.softuni.org/Contests/1502/Interfaces-and-Abstraction-Exercise>

* **Define an Interface IPerson**

**NOTE**: You need a public **StartUp** class with the namespace **PersonInfo**.

Define an **interface IPerson** with properties for **Name** and **Age**. Define a class **Citizen** that implements **IPerson** and has two properties **string** **name** and an **int** **age.** The **Citizen** should accept **name** and **age** upon initialization.

Try to create a new **Person** like this:

|  |
| --- |
| string name = Console.ReadLine();  int age = int.Parse(Console.ReadLine());  IPerson person = new Citizen(name, age);  Console.WriteLine(person.Name);  Console.WriteLine(person.Age); |

**Examples**

|  |  |
| --- | --- |
| **Input** | **Output** |
| Peter  25 | Peter  25 |

* **Multiple Implementation**

**NOTE**: You need a public **StartUp** class with the namespace **PersonInfo**.

Using the code from the previous task, define an **interface** **IIdentifiable** with a **string** property **Id** and an **interface** **IBirthable** with a **string** property **Birthdate** and implement them in the **Citizen** class. Rewrite the **Citizen** constructor to accept the new parameters.

Test your class like this:

|  |
| --- |
| string name = Console.ReadLine();  int age = int.Parse(Console.ReadLine());  string id = Console.ReadLine();  string birthdate = Console.ReadLine();  IIdentifiable identifiable = new Citizen(name, age,id, birthdate);  IBirthable birthable = new Citizen(name, age, id, birthdate);  Console.WriteLine(identifiable.Id);  Console.WriteLine(birthable.Birthdate); |

**Examples**

|  |  |
| --- | --- |
| **Input** | **Output** |
| Peter  25  9105152287  15/05/1991 | 9105152287  15/05/1991 |