Sunil Kumar Anjanappa

SunilKumar.Anjanappa@aptean.com

Abstract

Exercises on Interfaces and Abstractions

C# Assignments

Abstraction and Interfaces

## Problem 1.

Define an interface**IPerson** with properties for **Name** and **Age**. Define a class **Citizen**which implements **IPerson** and has a constructor which takes a **string** name and an **int** age.

Add the following code to your main method and submit it to Judge.



If you defined the interface and implemented it correctly, the test should pass.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Raj Kumar  25 | 2  Raj Kumar  25 |

## Problem 2.

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.

Add the following code to your main method and submit it to Judge.



If you defined the interfaces and implemented them, the test should pass.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Raj Kumar  25  9105152287  15/05/1994 | 1  String  1  String |

## Problem 3.

Create the following class hierarchy:

* **Soldier** – general class for soldiers, holding **id**, **first name** and **last name.**
  + **Private** – lowest base soldier type, holding the field **salary**(double).
    - **LeutenantGeneral** – holds a set of **Privates** under his command.
    - **SpecialisedSoldier –**general class for all specialised soldiers – holds the **corps** of the soldier. The corps can only be one of the following: **Airforces**or **Marines**.
      * **Engineer** – holds a set of **repairs**. A **repair**holds a **part name** and **hours worked**(int).
      * **Commando** – holds a set of **missions**. A mission holds **code name** and a **state** (***inProgress*** or ***Finished***). A mission can be finished through the method **CompleteMission()**.
    - **Spy** – holds the **code number**of the spy.

Extract **interfaces** for each class. (e.g. **ISoldier**, **IPrivate**, **ILeutenantGeneral**, etc.) The interfaces should hold their public properties and methods (e.g. **Isoldier** should hold **id**, **first name** and **last name**). Each class should implement its respective interface. Validate the input where necessary (corps, mission state) - input should match **exactly** one of the required values, otherwise it should be treated as **invalid**. In case of **invalid** **corps** the entire line should be skipped, in case of an **invalid** **missionstate** only the mission should be skipped.

Define proper constructors. Avoid code duplication through abstraction. Override **ToString()** in all classes to print detailed information about the object.

Below are **all** the Concrete class in the following formats:

* Private: “**Private <id> <firstName> <lastName> <salary>**”
* LeutenantGeneral: “**LeutenantGeneral <id> <firstName> <lastName> <salary> <private1Id> <private2Id> … <privateNId>**” where privateXId will **always** be an **Id** of a private already received through the input.
* Engineer: “**Engineer <id> <firstName> <lastName> <salary> <corps> <repair1Part> <repair1Hours> … <repairNPart> <repairNHours>**” where repairXPart is the name of a repaired part and repairXHours the hours it took to repair it (the two parameters will always come paired).
* Commando: “**Commando <id> <firstName> <lastName> <salary> <corps> <mission1CodeName> <mission1state> … <missionNCodeName> <missionNstate>**” a missions code name, description and state will always come together.
* Spy: “**Spy <id> <firstName> <lastName> <codeNumber>**”



Note: Zoom image to view details

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Private 1 Pesho Peshev 22.22 Commando 13 Stamat Stamov 13.1 Airforces  Private 222 Toncho Tonchev 80.08  LeutenantGeneral 3 Joro Jorev 100 222 1  End | Name: Pesho Peshev Id: 1 Salary: 22.22  Name: Stamat Stamov Id: 13 Salary: 13.10  Corps: Airforces  Missions:  Name: Toncho Tonchev Id: 222 Salary: 80.08  Name: Joro Jorev Id: 3 Salary: 100.00  Privates:    Name: Toncho Tonchev Id: 222 Salary: 80.08   Name: Pesho Peshev Id: 1 Salary: 22.22 |
| Engineer 7 Pencho Penchev 12.23 Marines Boat 2 Crane 17  Commando 19 Penka Ivanova 150.15 Airforces HairyFoot finished Freedom inProgress  End | Name: Pencho Penchev Id: 7 Salary: 12.23  Corps: Marines  Repairs:    Part Name: Boat Hours Worked: 2    Part Name: Crane Hours Worked: 17  Name: Penka Ivanova Id: 19 Salary: 150.15  Corps: Airforces Missions:    Code Name: Freedom State: inProgress |