## Military Elite

Create the following class hierarchy:

* **Soldier** - general class for **Soldiers**, holding **id**, **first name** and **last name.**
  + **Private** - lowest base **Soldier** type, holding the **salary**(**decimal**).
    - **LieutenantGeneral** - 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** (**int**).

Extract **interfaces** for each class. (e.g. **ISoldier**, **IPrivate**, **ILieutenantGeneral**, 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** **mission** **state** only the mission should be **skipped**.

You will receive from the console an unknown amount of lines containing information about soldiers until the command "**End**" is received. The information will be in one of the following formats:

* **Private**: "**Private <id> <firstName> <lastName> <salary>**"
* **LeutenantGeneral**: "**LieutenantGeneral <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>**"

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

* **Privates:**  
  Name: <firstName> <lastName> Id: <id> Salary: <salary>
* **Spy:**  
  Name: <firstName> <lastName> Id: <id>  
  Code Number: <codeNumber>
* **LieutenantGeneral:**  
  Name: <firstName> <lastName> Id: <id> Salary: <salary>  
  Privates:  
   <private1 ToString()>  
   <private2 ToString()>  
   …  
   <privateN ToString()>
* **Engineer:**  
  Name: <firstName> <lastName> Id: <id> Salary: <salary>  
  Corps: <corps>  
  Repairs:  
   <repair1 ToString()>  
   <repair2 ToString()>  
   …  
   <repairN ToString()>
* **Commando:**  
  Name: <firstName> <lastName> Id: <id> Salary: <salary>  
  Corps: <corps>  
  Missions:  
   <mission1 ToString()>  
   <mission2 ToString()>  
   …  
   <missionN ToString()>
* **Repair:**Part Name: <partName> Hours Worked: <hoursWorked>
* **Mission:**  
  Code Name: <codeName> State: <state>

**NOTE: Salary** should be printed rounded to **two decimal places** after the separator.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Private 1 Pesho Peshev 22.22 Commando 13 Stamat Stamov 13.1 Airforces  Private 222 Toncho Tonchev 80.08  LieutenantGeneral 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 |