More Exercises: OOP

You can check your solutions in **Judge system**: https://judge.softuni.bg/Contests/3166/OOP-More-Exercises

1. Program Loyal Customer

You are hired to create a program that monitors and encourages loyal customers of a store. For this you need a **`public Customer`** class. Each customer has:

- **`CustomerId`** – int
- **`Name`** – string
- **`Age`** – int
- **`Email`** – string
- **`BonusPoints`** – int

The class should have a constructor. Each customer must have customer id, name, age, email and starts with default **10 bonus points**.

The class **`Customer`** should have the following **methods**:

- **`public void AddBonusPoints(int points)`** – This method should **increase** the bonus points with the submitted ones and **print** the accumulated points in the following format: **`"You have { accumulated points } bonus points."`**.
- **`public void ExchangePoints(int points)`** – This method checks if there are enough points accumulated to complete the exchange. If there are not enough points **print: `"You do not have enough bonus points."`**. Otherwise, it reduces the accumulated points and **prints** the remaining points in the following format: **`"You have { points } points left."`**.

You will receive on the first line the **customer information** divided **by single space: `", "`**. You will first receive the **customer id**, followed by the **name**, **age** and **email**.

Until you receive the "**End**" command, you will be given the following commands: **`Bonus Points`** or **`Exchange`**.

Input	Output
33217854, Philip Reid, 32, pr@mail.com Bonus Points 25 Exchange 15 Bonus Points 45 End	You have 35 bonus points. You have 20 points left. You have 65 bonus points.
21223454, Clark Green, 23, kg@mail.com Exchange 20 Bonus Points 5 Exchange 15 Bonus Points 45 End	You do not have enough bonus points. You have 15 bonus points. You have 0 points left. You have 45 bonus points.

2. Golf & Spa Resort

You have the task to write a program that will serve the reception of a golf and spa resort. The policy of the complex does not allow the presence of people who are not registered. For this reason, the resort in its system wants to divide **people** into: **employees** and **customers**, and customers can be of two types: **members** or **guests**. For the program you will need:

`IPerson`:

- string **`FirstName`**
- string **`LastName`**

`Customer`:

Each **customer** is a **person** which is why he has **`FirstName`** and **`LastName`**:

`Guest`:

Each **guest** is a **customer** which is why he has **`FirstName`** and **`LastName`** and also:

- reuses the constructor from the base class
- **`NewGuest()`** – returns the following message: **`"Mr/Ms/Mrs {first name} {last name} registers as a guest."`**

`Member`:

Each **member** is a **customer** which is why he has **`FirstName`** and **`LastName`** and also:

- **`int MembershipId`**
- reuses part of the constructor from the base class
- **`GetMemberCard(string freeAccess)`** – returns the following message: **`"Member {first name} {last name} with membership number {membership id} gets free access to the {free access}."`**

`Employee`:

Each **employee** is a **person** which is why he has **`FirstName`** and **`LastName`** and also:

- string **`Department`**
- int **`EmployeeId`**
- **`StartWorkingDay()`** – returns the following message: **`"{first name} {last name} with id {employee id} starts a new working day in the department {department}."`**

You will receive from the console an unknown amount of lines. Until the command "**End**" is received, each line will contain information for a **guest**, **member** or **employee**. If a **member** is an **even** number in a row he gets **free access** to the **fitness** otherwise to the **spa**. Print the appropriate message when registering a person.

Input	Output
Lewis Scott 254566 Alice Potter Rick Ford Maintenance 33345 Terry Bell 349643 End	Member Lewis Scott with membership number 254566 gets free access to the spa. Mr/Ms/Mrs Alice Potter registers as a guest. Rick Ford with id 33345 starts a new working day in the department Maintenance. Member Terry Bell with membership number 349643 gets free access to the fitness.

3. Company Hierarchy

You have a company with **employees from three departments**. Your task is to organize employees into classes that store their information and calculate their salary.

You will receive from the console an unknown amount of lines. Until the command "**End**" is received, each line will contain information for employee from one of the three departments. For your program you will need:

`IEmployee` that has:

- string **`FirstName`**
- string **`LastName`**
- string **`Department`**
- decimal **`Salary`**
- **`GetSalary()`**

`SalesEmployee` is **employee** that has:

- string **`FirstName`**
- string **`LastName`**
- string **`Department`**
- decimal **`Salary`** – the **default** salary is **1000**
- decimal **`Profits`**
- **`GetSalary()`** – salary is calculated as the **sum** of **10%** of the **profit** plus the **default salary**
- **`ToString()`** – returns the following message: **`"{first name} {last name} from {department} has {profits} profits."`**

`Engineer` is **employee** that has:

- string **`FirstName`**
- string **`LastName`**
- string **`Department`**
- decimal **`Salary`** – the **default** salary is **1300**
- int **`YearsService`**
- **`GetSalary()`** – salary is calculated as the sum of the default salary and bonus 90 for each year of service
- **`ToString()`** – returns the following message: **`"{first name} {last name} from {department} has {years of service} years of service."`**

`Junior` is **employee** that has:

- string **`FirstName`**
- string **`LastName`**
- string **`Department`**
- decimal **`Salary`** – the **default** salary is **900**
- **`ToString()`** – returns the following message: **`"{first name} {last name} is {department} engineer."`**

After the "**End**" command, **print** each **employee** and on a new line print his **salary** in the following format: **`"Receives a salary {employee salary}."`**

Input	Output
`Alexia Evans Sales 5000`	`Alexia Evans from Sales has 5000 profits.`

Harold Tucker Engineering 5 Paul Baker Junior End	Receives a salary 1500.0. Harold Tucker from Engineering has 5 years of service. Receives a salary 1750. Paul Baker is Junior engineer. Receives a salary 900.