Name : Ferdinand Jacques

Class : L2BC

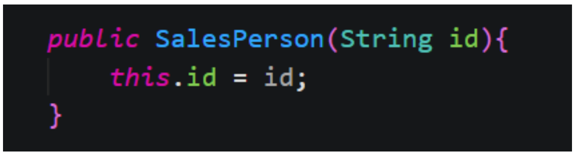
NIM : 2501982600

Course : COMP6699001 – Object Oriented Programming

Github link : https://github.com/Jacques7103/FE_OOP

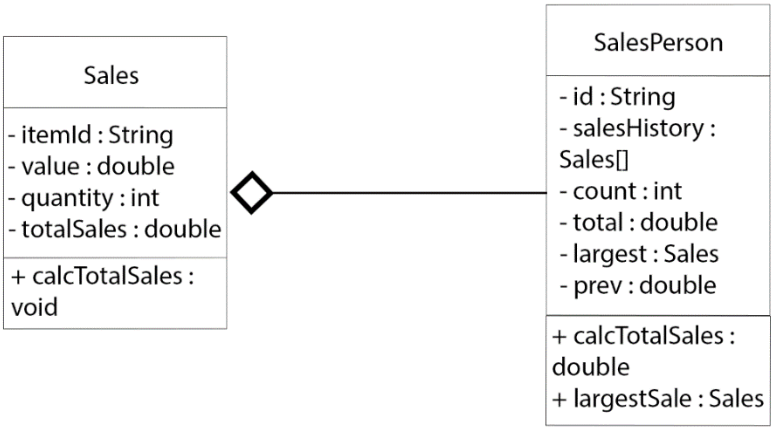
Case 1 :

1. A. So, class is a template that has their own methods, constructor, destructor, and objects. While instantiation of a class is the way we call the constructor of the class.
1. B We can use inheritance to store the common things that is contained in the subclasses. For example when we want to count the sales for a product, we can just create a class Item with the function calculateTotal. And then we just need to create the subclasses based on the name of the item that are available.
1. C. With the availability of built in library, it can save us more time since we don't have to create the function that is already available from the library. And with that more time, we could expand our program with more functions.
2. A.



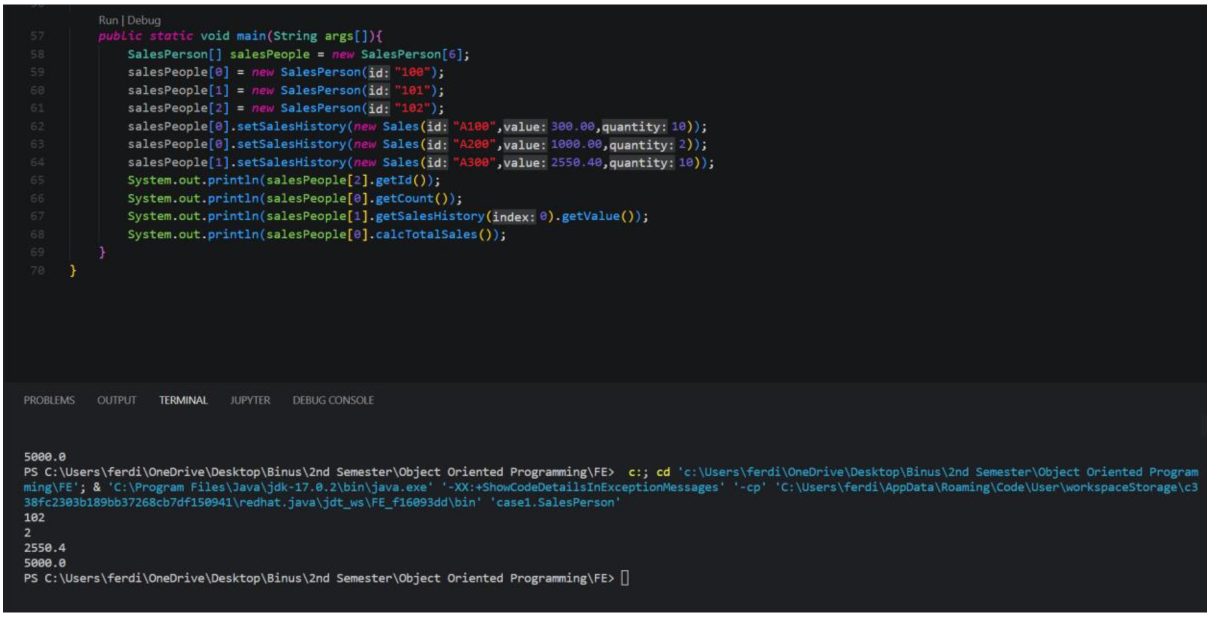
```
public SalesPerson(String id){
    this.id = id;
}
```

2. B. Accessor methods in SalesPerson class are important because it is used to get the private variable that can't be access normally.
2. C (i)



2. C (ii)

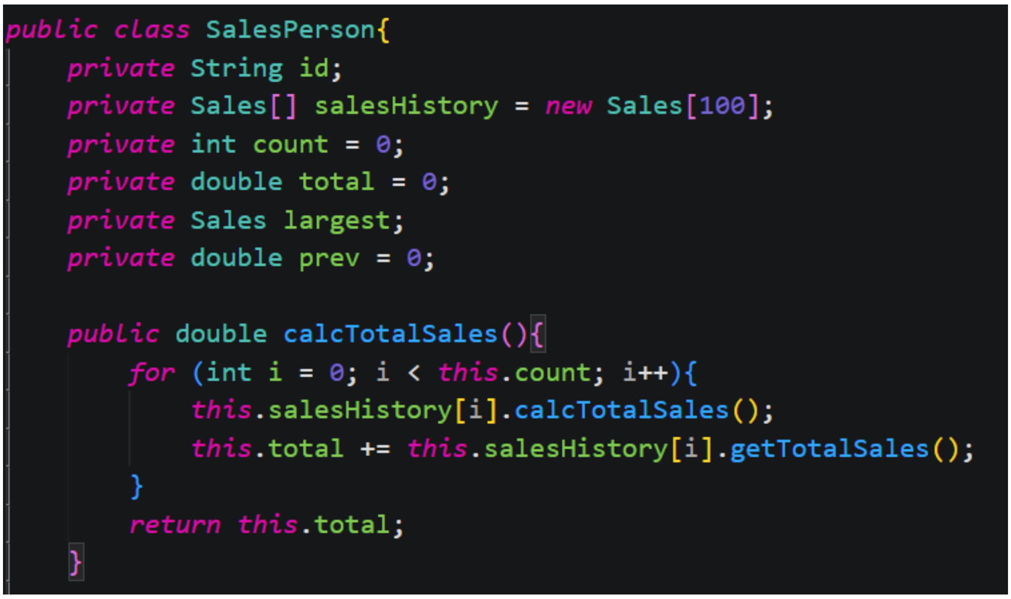
2. D.



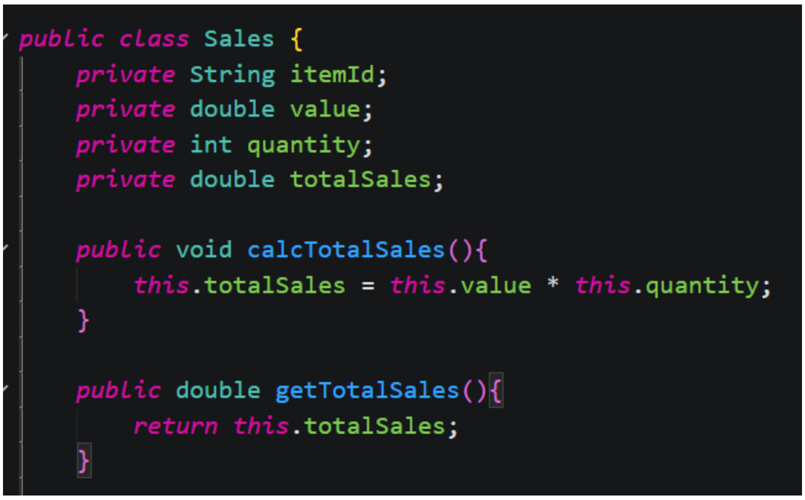
```
    public static void main(String args[]){
        SalesPerson[] salesPeople = new SalesPerson[6];
        salesPeople[0] = new SalesPerson(id: "100");
        salesPeople[1] = new SalesPerson(id: "101");
        salesPeople[2] = new SalesPerson(id: "102");
        salesPeople[0].setSalesHistory(new Sales(id: "A100",value: 300.00,quantity: 10));
        salesPeople[0].setSalesHistory(new Sales(id: "A200",value: 1000.00,quantity: 2));
        salesPeople[1].setSalesHistory(new Sales(id: "A300",value: 2550.40,quantity: 10));
        System.out.println(salesPeople[2].getId());
        System.out.println(salesPeople[0].getCount());
        System.out.println(salesPeople[1].getSalesHistory(index: 0).getValue());
        System.out.println(salesPeople[0].calcTotalSales());
    }
}
```

```
5000.0
PS C:\Users\ferdi\OneDrive\Desktop\Binus\2nd Semester\Object Oriented Programming\FE>  c:; cd 'c:\Users\ferdi\OneDrive\Desktop\Binus\2nd Semester\Object Oriented Program
ming\FE'; & 'C:\Program Files\Java\jdk-17.0.2\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ferdi\AppData\Roaming\Code\User\workspaceStorage\c3
38fc2303b189bb37268cb7df150941\redhat.java\jdt_ws\FE_f16093dd\bin' 'case1.SalesPerson'
102
2
2550.4
5000.0
PS C:\Users\ferdi\OneDrive\Desktop\Binus\2nd Semester\Object Oriented Programming\FE> 
```

2. E

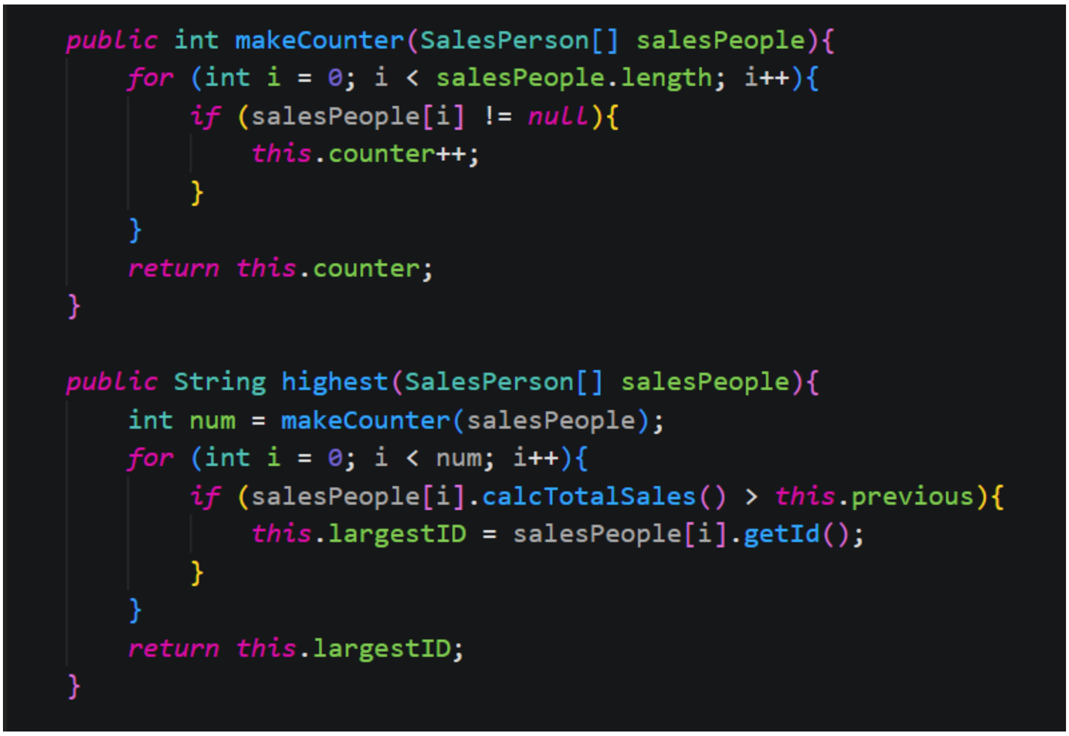


```
public class SalesPerson{
    private String id;
    private Sales[] salesHistory = new Sales[100];
    private int count = 0;
    private double total = 0;
    private Sales largest;
    private double prev = 0;

    public double calcTotalSales(){
        for (int i = 0; i < this.count; i++){
            this.salesHistory[i].calcTotalSales();
            this.total += this.salesHistory[i].getTotalSales();
        }
        return this.total;
    }
```



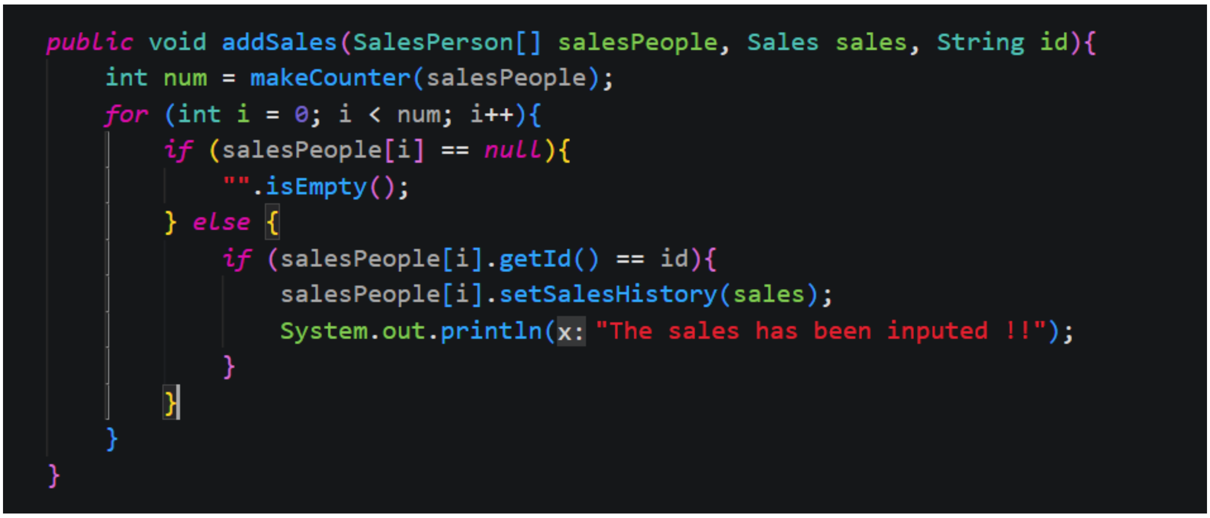
```
public class Sales {
    private String itemId;
    private double value;
    private int quantity;
    private double totalSales;

    public void calcTotalSales(){
        this.totalSales = this.value * this.quantity;
    }

    public double getTotalSales(){
        return this.totalSales;
    }
```

2. F.



```
public int makeCounter(SalesPerson[] salesPeople){
    for (int i = 0; i < salesPeople.length; i++){
        if (salesPeople[i] != null){
            this.counter++;
        }
    }
    return this.counter;
}

public String highest(SalesPerson[] salesPeople){
    int num = makeCounter(salesPeople);
    for (int i = 0; i < num; i++){
        if (salesPeople[i].calcTotalSales() > this.previous){
            this.largestID = salesPeople[i].getId();
        }
    }
    return this.largestID;
}
```

2. G.

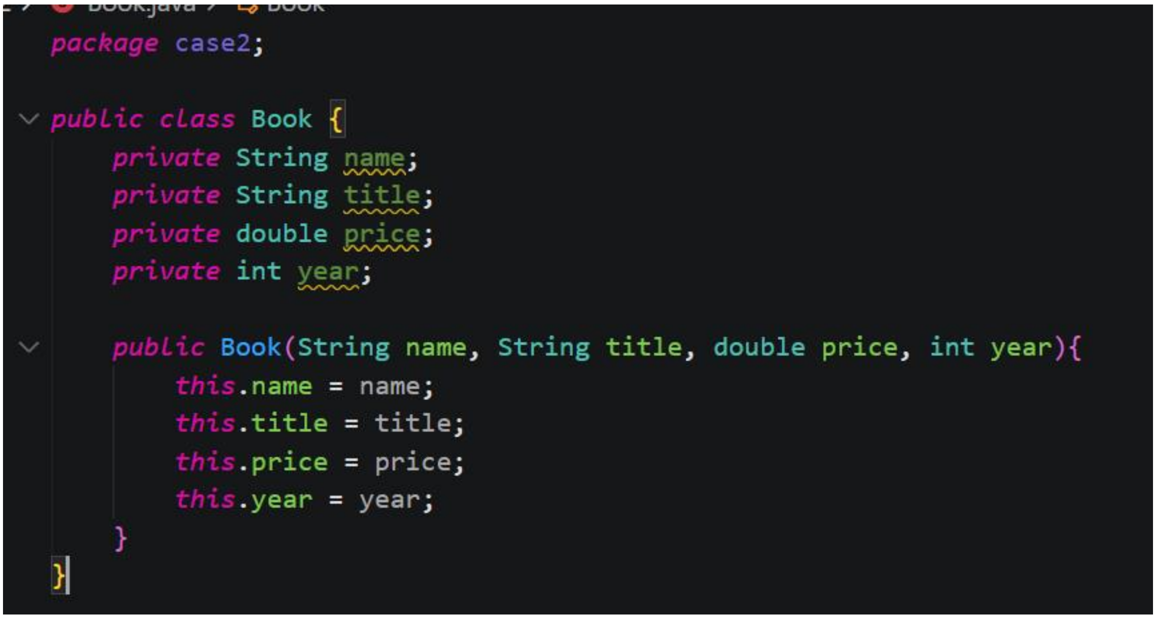


```
public void addSales(SalesPerson[] salesPeople, Sales sales, String id){
    int num = makeCounter(salesPeople);
    for (int i = 0; i < num; i++){
        if (salesPeople[i] == null){
            "".isEmpty();
        } else {
            if (salesPeople[i].getId() == id){
                salesPeople[i].setSalesHistory(sales);
                System.out.println(x: "The sales has been inputed !!");
            }
        }
    }
}
```

2. H. I suggest to add the date to keep the track of the date in order to know when to calculate the salary.

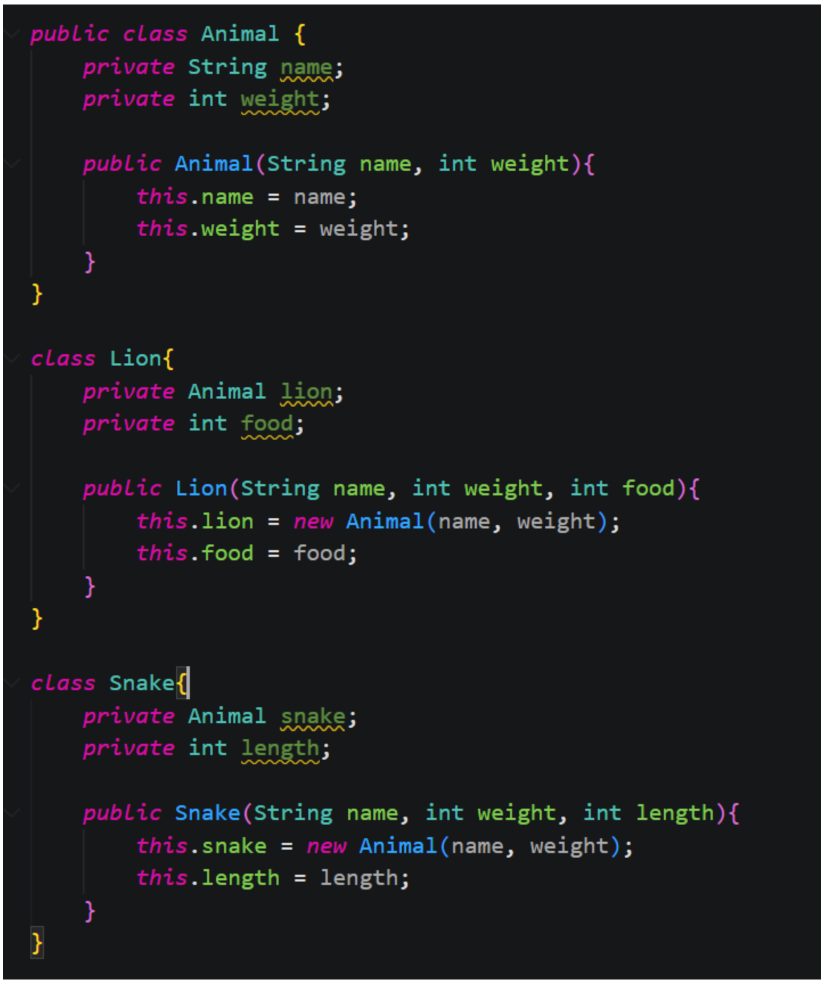
Case 2 :

1. By “new Book ("Ida Bagus","The Young Doctor",900,1974)” means that we are instantiating a new class with the author’s name “Ida Bagus”, the book’s title “The Young Doctor”, the price of the book “900” and the year of publication “1974”.

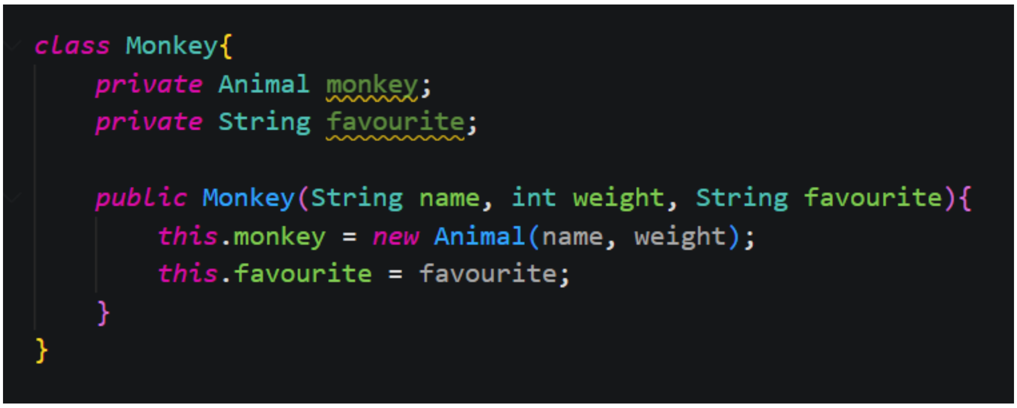


```java
package case2;

public class Book {
    private String name;
    private String title;
    private double price;
    private int year;

    public Book(String name, String title, double price, int year){
        this.name = name;
        this.title = title;
        this.price = price;
        this.year = year;
    }
}
```

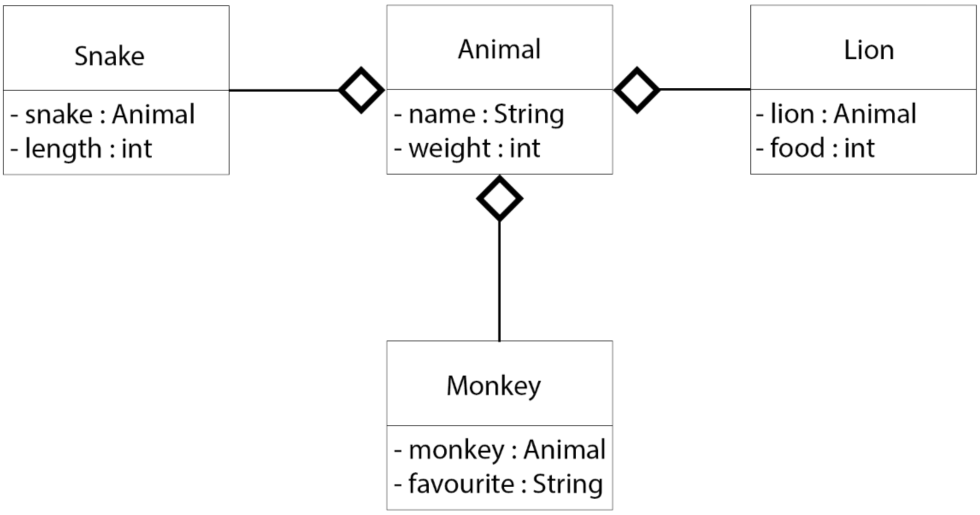
2.



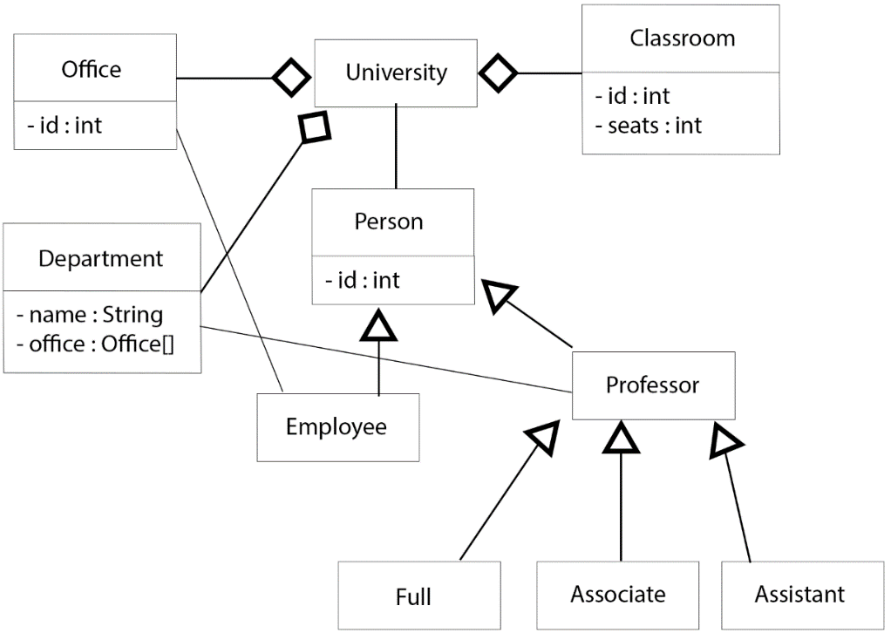
```java
public class Animal {
    private String name;
    private int weight;

    public Animal(String name, int weight){
        this.name = name;
        this.weight = weight;
    }
}

class Lion{
    private Animal lion;
    private int food;

    public Lion(String name, int weight, int food){
        this.lion = new Animal(name, weight);
        this.food = food;
    }
}

class Snake{
    private Animal snake;
    private int length;

    public Snake(String name, int weight, int length){
        this.snake = new Animal(name, weight);
        this.length = length;
    }
}
```



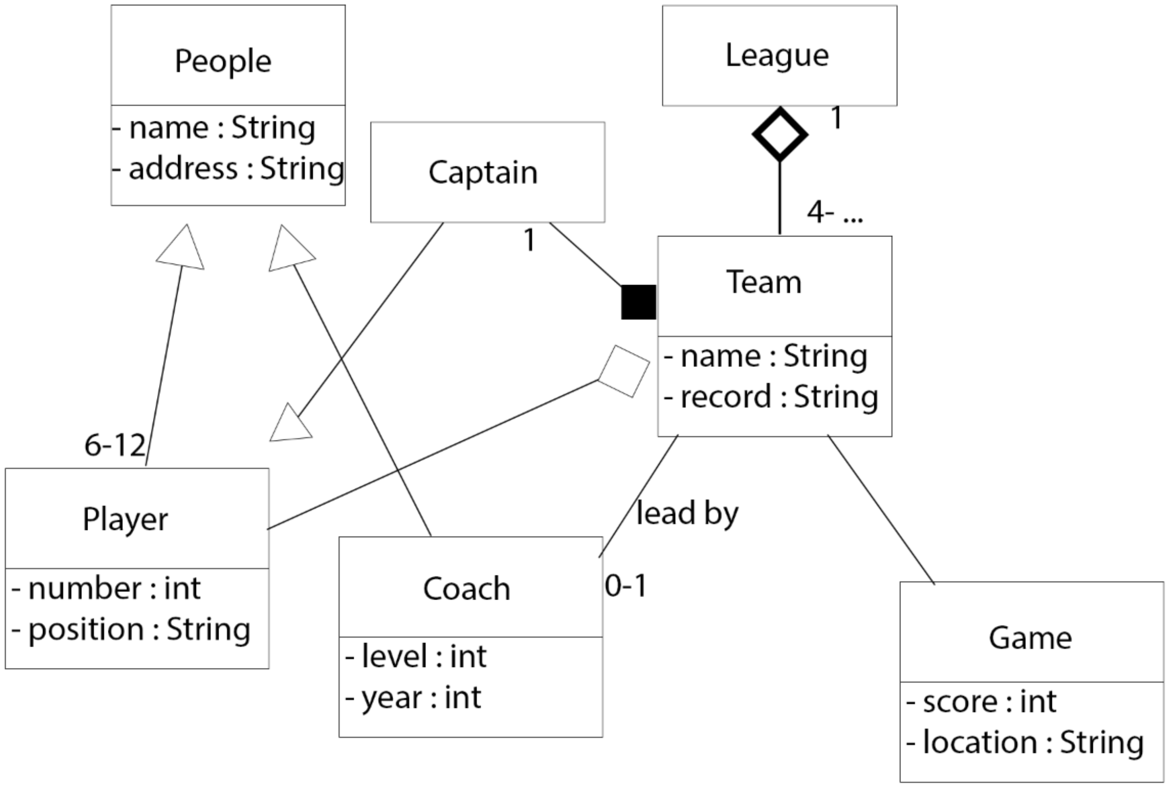
```
class Monkey{
    private Animal monkey;
    private String favourite;

    public Monkey(String name, int weight, String favourite){
        this.monkey = new Animal(name, weight);
        this.favourite = favourite;
    }
}
```



3.



4.


People
- name : String
- address : String
Captain
League
1
4- ...
1
Team
- name : String
- record : String
6-12
Player
- number : int
- position : String
Coach
- level : int
- year : int
lead by
0-1
Game
- score : int
- location : String