Name:
Student ID:
Day %30 ( ) Day %100 ( ) Night ( ) NYP ( )

SCORE	Q1	Q2	Q3	TOTAL

KARABUK UNIVERSITY | COMPUTER ENGINEERING DEPARTMENT
Object Oriented Programming Midterm | Nesneye Yönelimli Programlama | Fall | 14.11.2024 | 09.00

QUESTION-1 :Write outputs in the given table. All the classes are declared in the same package.
Çıktıları verilen tabloya yazın. Tüm sınıflar aynı pakette yazılmıştır. (30P)

```
enum Job{
Secretary(100,"Scheduling meetings."),
Manager(101,"Implementing business strategies."),
Engineer(102,"Managing projects.");
final private int id;
final private String jobDesc;
    private Job(int id, String jobDesc) {
        this.id = id;
        this.jobDesc = jobDesc;  }
int getID(){ return id;}
String getJobDesc(){ return jobDesc;}
}// end of job enum

public class Employee {
    String name;
    int id;
    static int index;
    Job job;
    public Employee(String name, Job job) {
        this.name = name;
        this.id = ++index;
        this.job = job;
    }
    public String toString(){
      return "Employee "+name+" is working as: "+job;
    }
    void printEmpInfo(){
        System.out.println(id+":"+name);
        System.out.println("Job id:"+job.getID()+
                ", desc:"+job.getJobDesc());
    } }// ends of employee class
```

```
public class MidtermExam2024 {
   public static void main(String[] args) {
   Employee e1=new Employee("Ali", Job.Manager);
   Employee e2=new Employee("Omer", Job.Engineer);
   Employee e3=new Employee("Lala", Job.Secretary);
        System.out.println(e1);          //output 1
        System.out.println(e1.id);       //output 2
        System.out.println(e1.index);    //output 3
        e3.job=e2.job;
        System.out.println(e2);          //output 4
        e3.printEmpInfo();               //output 5
    } }
```

(01)	Employee Ali is working as: Manager
(02)	1
(03)	3
(04)	Employee Omer is working as: Engineer
(05)	3:Lala Job id:102, desc:Managing projects.

QUESTION-2 (30P): In the following table, write the appropriate term given below in front of each code snippet.
Terms are: exception (run time error), exception handling, compilation error, constructor overloading.
Note: you can use one term in more than one location and it's not necessary to use all the terms.

Aşağıdaki tabloda, her kod parçasının önüne aşağıda verilen uygun terimi yazınız.
Not: bir terimi birden fazla yerde kullanabilirsiniz ve tüm terimleri kullanmanız gerekli değildir.

Code snippet	Term
class X{ float a=13.5; int b=4; float method(){ return a/b; } }	compilation error
class Y{ int a=9, b=3; int method(){ b= a/b - b; return a/b; } }	exception (run time error),
class T{ int a; public T(int a){ this.a=a; } public T(float a){ a=10; } }	constructor overloading
class S{ int a; public S(int a){ this.a=a; } public S(String s){ a=10; } }	constructor overloading

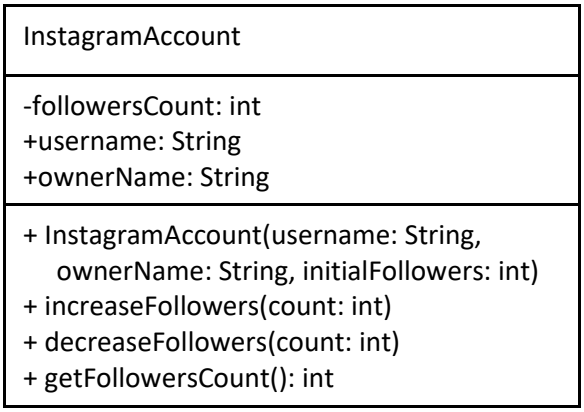
class F{ int i=10,int j=2;
void m(){
try{System.out.println(i/j);}
catch(Exception e){ System.out.println(e.getMessage());} } }
	exception handling

QUESTION-3 (40P): Write the InstagramAccount class according to the UML diagram. Do the followings:

- The constructor should assign username, owner name, and initial numbers of the followers.
- **increaseFollowers** increases the follower count. In this method, increasing followers using negative numbers of followers should cause an exception. Handle the exception in the main.
- **decreaseFollowers** decreases the follower count. In this method, decreasing more followers than avaliable or negaive numbers should cause an exception. Handle the exception in the main.

(TR) Aşağıdaki UML diyagramını kullanarak InstagramAccount sınıfını yazınız. Aşağıdakileri uygulayınız:

- Yapıcı (kurucu), username, owner name ve takipçilerin (followers) ilk sayılarını atamalıdır.
- **increaseFollowers** takipçi sayısını artırır. Bu yöntemde, negatif takipçi sayıları kullanarak takipçileri artırmak bir istisnaya neden olmalıdır. İstisnayı main içinde ele alın.
- **decreaseFollowers** takipçi sayısını azaltır. Bu yöntemde, mevcut olandan daha fazla takipçi artırmak veya negatif sayılar kullanmak bir istisnaya neden olmalıdır. İstisnayı main içinde ele alın.




```
public class InstagramAccount {
    private int followersCount;
    public String username;
    public String ownerName;

    // Constructor
    public InstagramAccount(String username, String ownerName, int initialFollowers) {
        this.username = username;
        this.ownerName = ownerName;
        this.followersCount = initialFollowers;
    }

    // Method to increase followers
    public void increaseFollowers(int count) {
        if (count <= 0) {
            throw new IllegalArgumentException("Followers count to increase must be
positive.");
        }
        this.followersCount += count;
    }

    // Method to decrease followers
    public void decreaseFollowers(int count) {
        if (count <= 0) {
            throw new IllegalArgumentException ("Followers count to decrease must be
positive.");
        }
        if (count > this.followersCount) {
            throw new IllegalArgumentException ("Cannot decrease more followers than
available.");
        }
        this.followersCount -= count;
    }

    // Method to get followers count
    public int getFollowersCount() {
        return this.followersCount;
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        // Create an Instagram account
        InstagramAccount account = new InstagramAccount("alice_mane ", "Alice Mane", 100);

        try {
            // Valid increase
            account.increaseFollowers(50);
            System.out.println("Followers after increase: " + account.getFollowersCount());

            // Invalid increase
            account.increaseFollowers(-10); // This will throw an exception
        } catch (InvalidFollowerOperationException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }

}
```