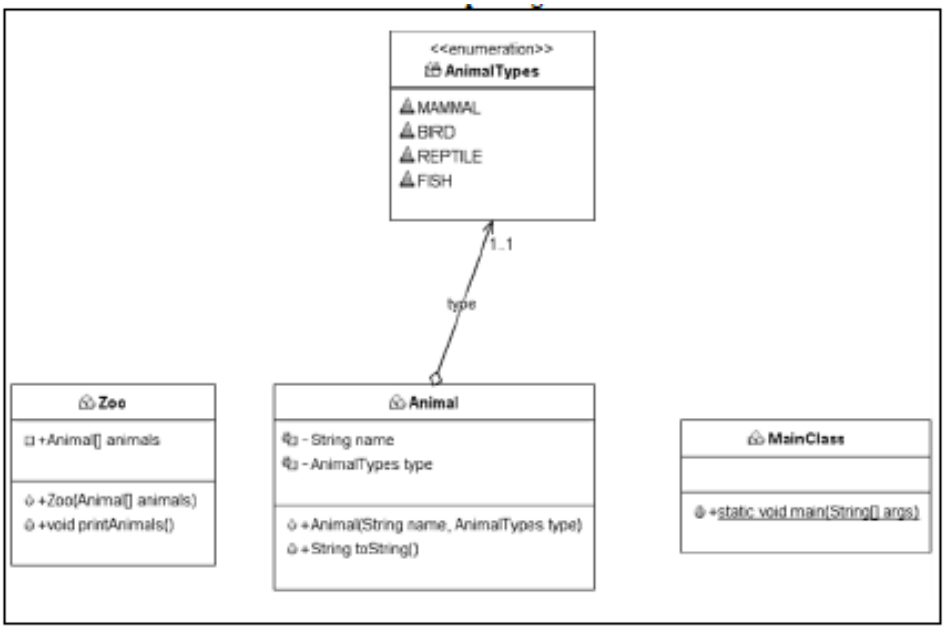
Homework 1

QUESTION 1: Write necessary classes and enum types for given following UML diagram. All the classes are declared in the same package.



a) Create Animal, Zoo Classes and AnimalType according to the given UML class diagrams.

b) In the main method, create 4 objects from Animal class, each will be different animal type. Put them all in a array of type Animal called "animals".

c) Create a myZoo object. Print all animals in the zoo using printAnimal() method.

[a) Verilen UML sınıf diyagramlarına göre Animal ve Zoo Sınıflarını ve AnimalType enum-ını oluşturun.
b) Main Metodda, her biri farklı AnimalTypedan olmak uzere 4 Animal nesnesi oluşturun. Hepsini "animals" adında bir Animal dizisine koyun.
c) myZoo nesnesi oluşturun. printAnimal () metodunu kullanarak Zoo'daki tüm Animal'lari yazdırın.]

QUESTION.2: Write necessary lines of code to perform followings

a. Create a class called **EncapsulatedStudent**, which applies data encapsulation (data hiding) to **Student** class.

```
public class Student {
    public int midterm;   // midterm cannot be less than 0 or greater than 100
    public int finalExam; // finalExam cannot be less than 0 or greater than 100
}
```

b. **EncapsulatedStudent** class should throw an exception (with an error message) **if any illegal arguments** (**midterm and finalExams**) are set.

c. Inside the MainClass, create an instance (student1) from **EncapsulatedStudent**, set attributes for it, handle any exception, and print the error message on the screen. Use getAverage() method to return the average score, then print it on the screen. [**averageScore= midTerm *0.4 + finalExam*0.6;**]

QUESTION.3: Write outputs in the given table. All the classes are declared in the same package.

```
public class Worker {
    String name;
    public static String compName,
    public static int number;
    public int id;

    public Worker(String n){
        this(n, number);
        compName ="Iron and Steel Inc.";
    }

    private Worker(String n, int num){
        number = num;
        number++;
        this.name =n;
        this.id =number;
    }
} //-----------------------------------------------
public class Company {
    static int index;
    Worker[] workers;

   public Company(int num){
       workers = new Worker[num];
   }

    void addWorker(Worker w){
       workers[index] = w;
       index++;
    }

    public void showData(Worker w){
     System.out.println(w.id +":"+w.compName);
    }
}
```

```
public class MainClass {
  public static void main(String[] args) {

    Worker w1 = new Worker("Ahmet");
    System.out.println(Worker.compName);   //(1)

    Worker w2 = new Worker("Mert");
    Worker w3 = new Worker("Cenk");
    System.out.println(Worker.number);     //(2)
    Worker w4 = new Worker("Okan");
    System.out.println(w2.id);             //(3)

    Company comp = new Company(4);
    comp.addWorker(w1);
    comp.addWorker(w2);
    comp.addWorker(w3);
    comp.addWorker(w4);

     System.out.println(Company.index);    //(4)
     Worker.compName ="Just Steel Inc.";
     comp.showData(w3);                    //(5)
 }
}
```

(1).	
(2).	
(3).	
(4).	
(5).	