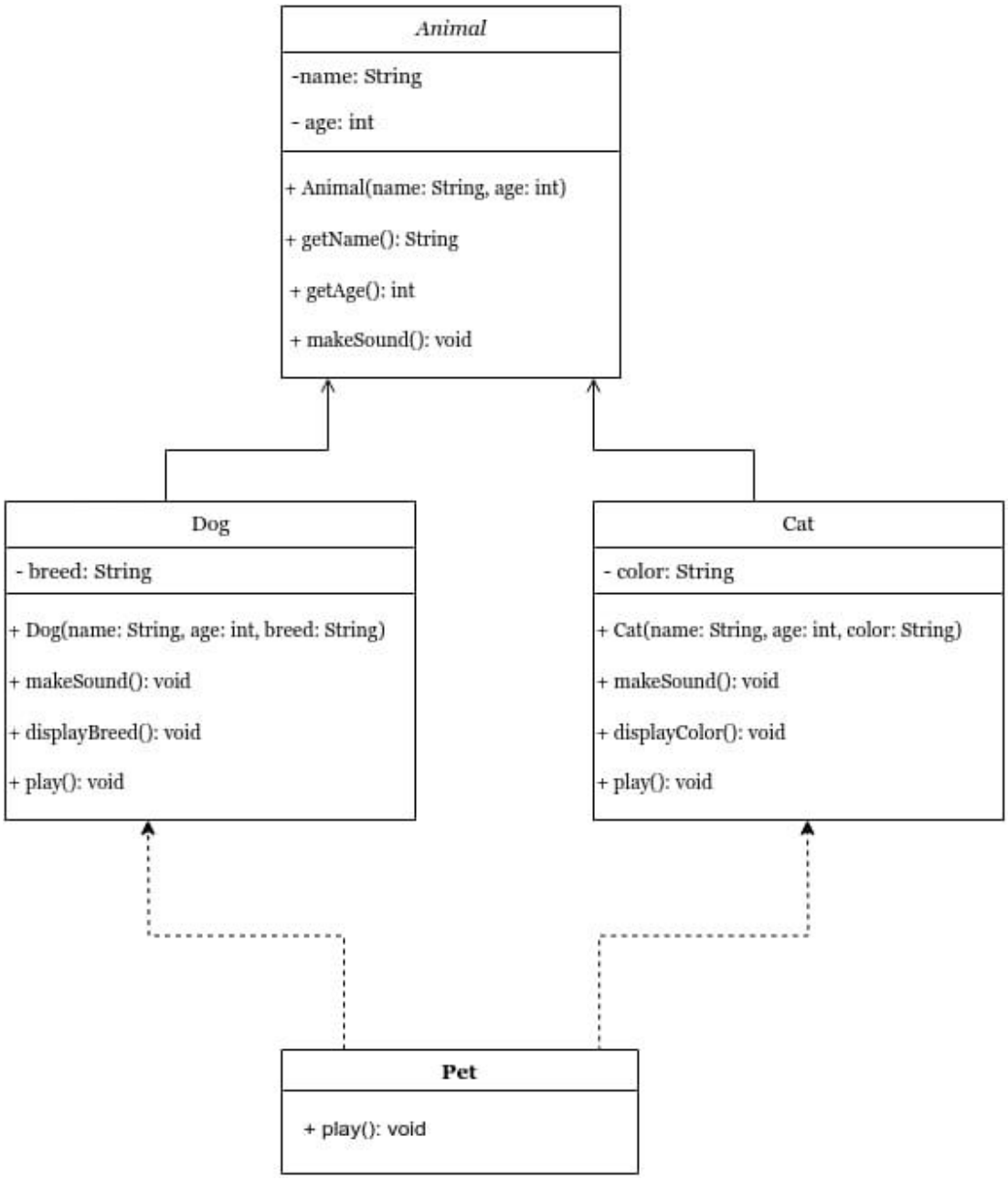
Lab Assignment
Object-Oriented Programming Concepts

Name: Jyotirmay Mondal
Id: 2024-1-60-080
Sec-13

UML DIAGRAM



Animal Class

```
package animaltest;

abstract class Animal {
    private String name;
    private int age;


public Animal (String name, int age){       //constructor of
animal
        this.name = name;
        this.age = age;
}

    public String getName(){            //getter for name
       return name;
}

    public int getAge() {                   //getter for age
       return age;
}

public abstract void makeSound();            //another absract
method

}
```

Dog Class

```
package animaltest;

public class Dog extends Animal implements Pet {
    private String breed;

    public Dog(String name, int age, String breed) {     //
Constructor for Dog's name, age, and breed
        super(name, age);
        this.breed = breed;
    }

//override is happening here
    public void makeSound() {
//giving body to makesound method
        System.out.println(getName() + " says Woof Woof!");
    }

    public void displayBreed() {
//giving body to displabreed method
        System.out.println("Breed: " + breed);
    }


    public void play() {
//override is happening here
        System.out.println(getName() + " is playing fetch with a
ball!");
    }
}
```

Cat Class

```
package animaltest;

public class Cat extends Animal implements Pet {
    private String color;

    public Cat(String name, int age, String color) {     //
Constructor for cat's name, age, and color
        super(name, age);
        this.color = color;

    }

    public void makeSound() {                          //override
is happening here
        System.out.println(getName() + " says Meow Meow!");
    }

    public void displayColor () {                       //giving
body to displaycolor method
        System.out.println ("Color: " + color);

    }
    public void play() {                               //override is
happening here
        System.out.println(getName() + " is playing with rope!");
    }
}
```

Pet Interface

```
package animaltest;


public interface Pet {   //abstract method implemented by any
class that implements pet
    void play();
}
```

AnimalTest Class

```
package animaltest;

public class AnimalTest {

    public static void main(String[] args) {
        // Create an array of Animal objects including dog and cat
        Animal[] ani = new Animal[6];

        ani[0] = new Dog("Tommy", 7 , "Golden Retriever" );
        ani[1] = new Dog("Shelvy", 5 , "Chihuahua" );
        ani[2] = new Dog("Rambo", 6 ,"Siberian Husky" );
        ani[3] = new Cat("Daisy", 2 ,"Golden" );
        ani[4] = new Cat("Jerry", 4 ,"White" );
        ani[5] = new Cat("Lily", 3 ,"Black & White" );

        for (Animal a : ani) {
            System.out.println("Name: " + a.getName());   //shoing
name
            System.out.println("Age: " + a.getAge());
//sjhowing age

            a.makeSound(); // Polymorphic call


            if (a instanceof Dog) {                     // calling
specific methods using downcasting
                ((Dog) a).displayBreed();
            } else if (a instanceof Cat) {
                ((Cat) a).displayColor();
            }

            if (a instanceof Pet) {   //Check if the animal is an
instance of Pet
                ((Pet) a).play();        // If it is a Pet, call
the play() method
            }
                 System.out.println("");   //for a line break
        }

    }
}
```