Worksheet 06

CTEC 22043
Object Oriented Programming

Student No: CT/2021/002

Faculty of Computing and Technology
University of Kelaniya
Sri Lanka

Q1.
Code:

Pet.java

```java
package Q_01;

abstract class Pet {
    private String name;

    public String getName(){
        return name;
    }

    public void setName (String petName){
        name = petName;
    }

    public String speak(){
        return "I'm your cuddly pet.";
    }
}
```

Cat.java

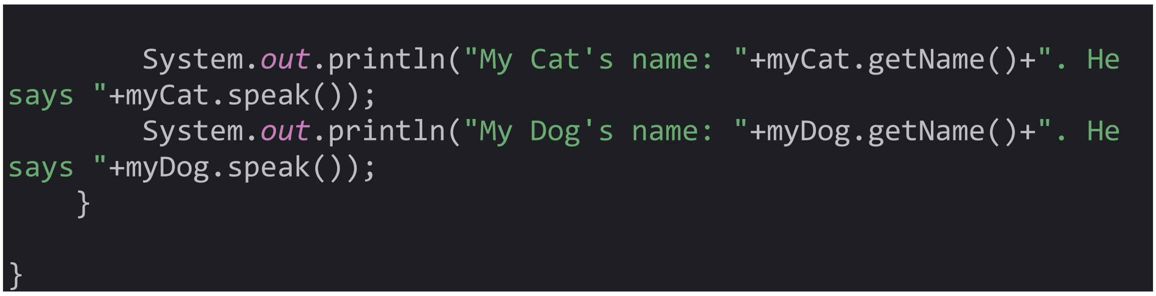
```java
package Q_01;


public class Cat extends Pet{
    @Override
    public String speak(){
        return "";
    }
}
```

Dog.java

```java
package Q_01;


public class Dog extends Pet {
    @Override
    public String speak(){
        return "";
    }

}
```

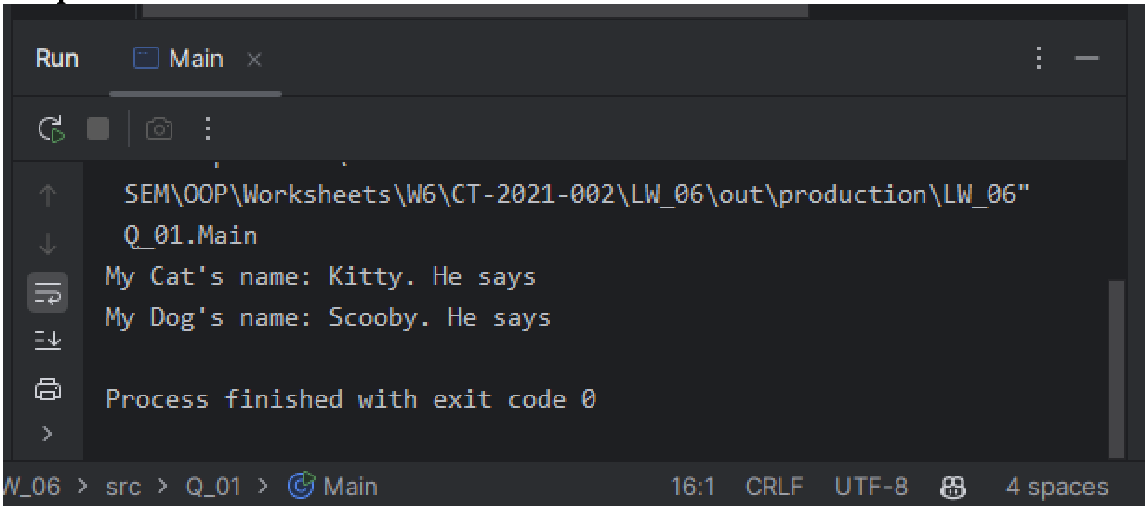
Main.java

```java
package Q_01;

public class Main {
    public static void main(String[] args) {
        Cat myCat = new Cat();
        Dog myDog = new Dog();

        myCat.setName("Kitty");
        myDog.setName("Scooby");
```



```
        System.out.println("My Cat's name: "+myCat.getName()+". He
says "+myCat.speak());
        System.out.println("My Dog's name: "+myDog.getName()+". He
says "+myDog.speak());
    }

}
```

Output:



```
 SEM\OOP\Worksheets\W6\CT-2021-002\LW_06\out\production\LW_06"
 Q_01.Main
My Cat's name: Kitty. He says
My Dog's name: Scooby. He says

Process finished with exit code 0
```

Q2.
Code:

Pet.java

```java
package Q_02;

abstract class Pet {
    private final String name;

    public Pet(String name) {
        this.name = name;
    }

    public String getName(){
        return name;
    }
    public  abstract String getType();

}
```

Cat.java

```java
package Q_02;


public class Cat extends Pet {
    public Cat(String name){
        super(name);
    }

    @Override
    public String getType(){
        return "Cat";
    }


}
```

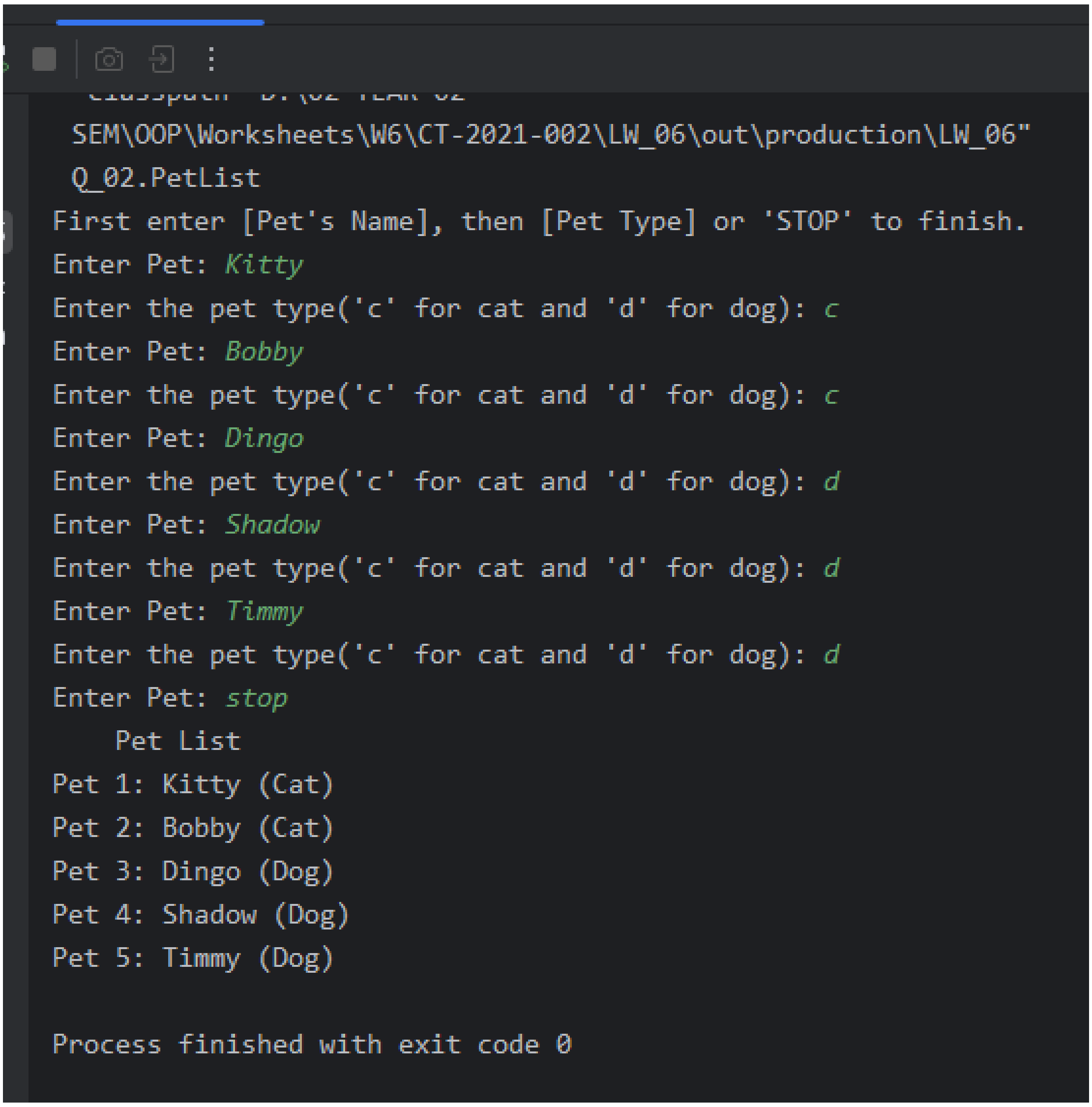
Dog.java

```java
package Q_02;

public class Dog extends Pet {
    public Dog(String name){
        super(name);
    }

    @Override
    public String getType(){
        return "Dog";
    }

}
```

PetList.java

```
package Q_02;

import java.util.Scanner;

public class PetList {
    public static final int MAX_VALUE = 10;
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        Pet[] pets = new Pet[MAX_VALUE];
        int petCount = 0;

        //input the data to array
        System.out.println("First enter [Pet's Name], then [Pet
Type] or 'STOP' to finish.");

        while(true){
            System.out.print("Enter Pet: ");
            String name = scan.nextLine();

            if(name. equalsIgnoreCase ("stop")){
                System.out.println("Exiting...");
                break;
            }

            System.out.print("Enter the pet type('c' for cat and
'd' for dog): ");
            char type = scan.nextLine().charAt(0);

            if(type == 'c'){
                pets[petCount] = new Cat(name);
            } else if (type == 'd'){
                pets[petCount] = new Dog(name);
            }else{
                System.out.print("Invalid Input.!");
            }
            petCount = petCount + 1;
        }

        //print the array
        System.out.println("\tPet List");
        for(int i = 0; i < petCount; i++){
            System.out.println("Pet "+(i+1)+ ":
"+pets[i].getName()+" ("+pets[i].getType()+")");
        }
    }
}
```

Output:



```
  -classpath "D:\02 YEAR 02
  SEM\OOP\Worksheets\W6\CT-2021-002\LW_06\out\production\LW_06"
  Q_02.PetList
First enter [Pet's Name], then [Pet Type] or 'STOP' to finish.
Enter Pet: Kitty
Enter the pet type('c' for cat and 'd' for dog): c
Enter Pet: Bobby
Enter the pet type('c' for cat and 'd' for dog): c
Enter Pet: Dingo
Enter the pet type('c' for cat and 'd' for dog): d
Enter Pet: Shadow
Enter the pet type('c' for cat and 'd' for dog): d
Enter Pet: Timmy
Enter the pet type('c' for cat and 'd' for dog): d
Enter Pet: stop
    Pet List
Pet 1: Kitty (Cat)
Pet 2: Bobby (Cat)
Pet 3: Dingo (Dog)
Pet 4: Shadow (Dog)
Pet 5: Timmy (Dog)

Process finished with exit code 0
```

Q3.
Code:

Pet.java

```
package Q_03;

abstract class Pet {
    private final String name;

    public Pet(String name) {
        this.name = name;
    }

    public String getName(){
        return name;
    }
    public  abstract String getType();

}
```

Cat.java

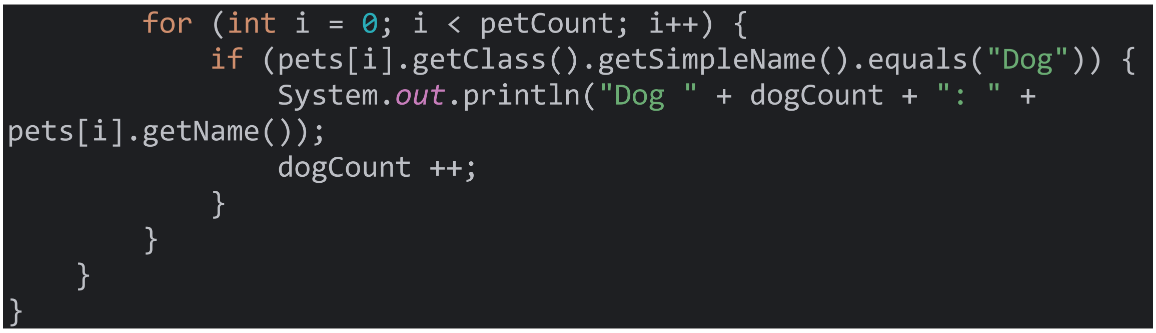
```
package Q_03;

public class Cat extends Pet {
    public Cat(String name){
        super(name);
    }

    @Override
    public String getType(){
        return "Cat";
    }


}
```

Dog.java

```
package Q_03;

public class Dog extends Pet {
    public Dog(String name){
        super(name);
    }

    @Override
    public String getType(){
        return "Dog";
    }

}
```

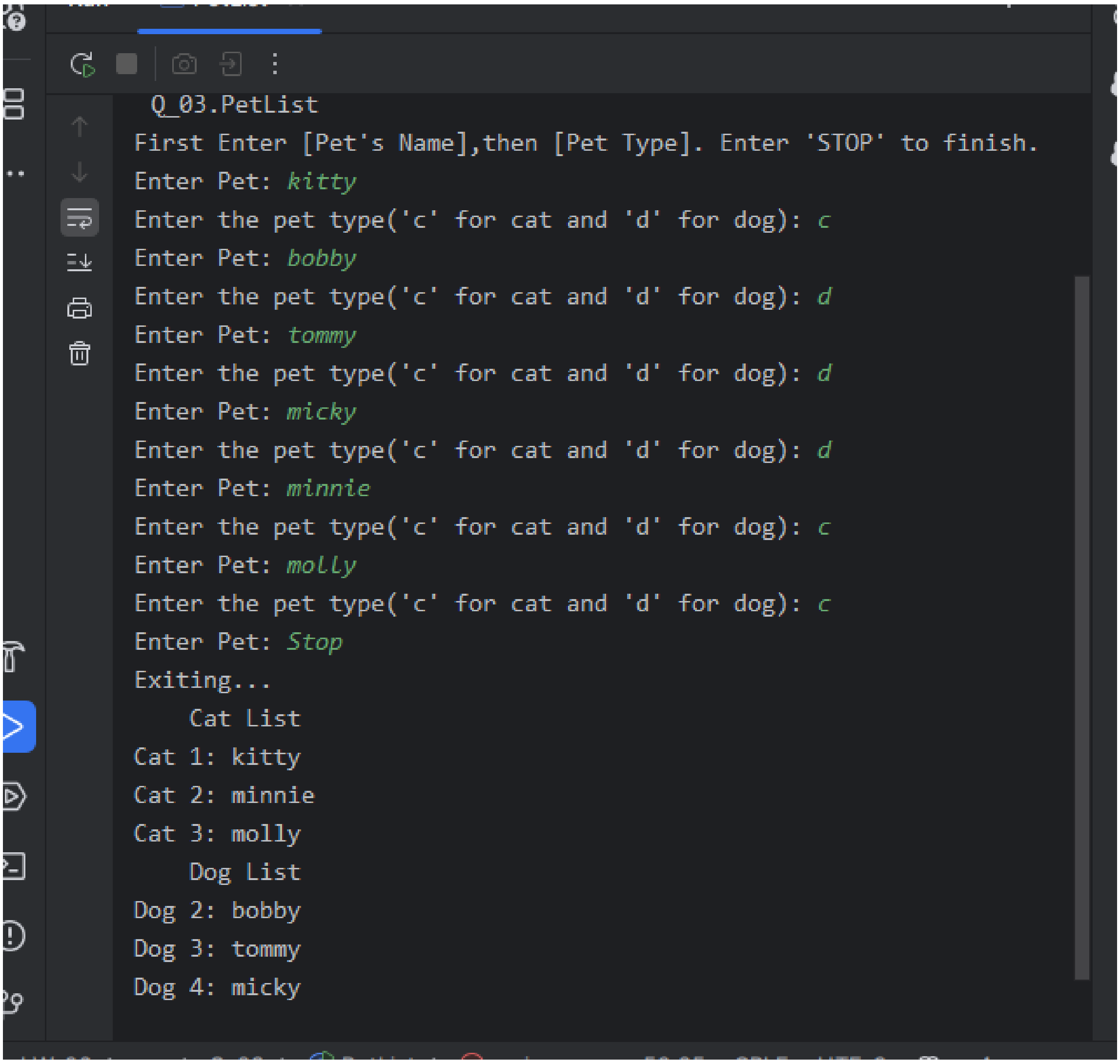
PetList.java

```
package Q_03;


import java.util.Scanner;

public class PetList {
    public static final int MAX_VALUE = 10;

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        Pet[] pets = new Pet[MAX_VALUE];
        int petCount = 0;

        //input the data to array
        System.out.println("First Enter [Pet's Name],then [Pet
Type]. Enter 'STOP' to finish.");


        while (true) {
            System.out.print("Enter Pet: ");
            String name = scan.nextLine();

            if (name.equalsIgnoreCase("stop")) {
                System.out.println("Exiting...");
                break;
            }

            System.out.print("Enter the pet type('c' for cat and
'd' for dog): ");
            char type = scan.nextLine().charAt(0);

            if (type == 'c') {
                pets[petCount] = new Cat(name);
            } else if (type == 'd') {
                pets[petCount] = new Dog(name);
            } else {
                System.out.print("Invalid Input.!");
            }
            petCount = petCount + 1;
        }

        //print the array
        System.out.println("\tCat List");
        int catCount = 0;
        for (int i = 0; i < petCount; i++) {
            if (pets[i].getClass().getSimpleName().equals("Cat")) {
                System.out.println("Cat " + catCount + ": " +
pets[i].getName());
                catCount ++;
            }
        }

        System.out.println("\tDog List");
        int dogCount = 0;
```



```
        for (int i = 0; i < petCount; i++) {
            if (pets[i].getClass().getSimpleName().equals("Dog")) {
                System.out.println("Dog " + dogCount + ": " +
pets[i].getName());
                dogCount ++;
            }
        }
    }
}
```

Output:



```
  Q_03.PetList
First Enter [Pet's Name],then [Pet Type]. Enter 'STOP' to finish.
Enter Pet: kitty
Enter the pet type('c' for cat and 'd' for dog): c
Enter Pet: bobby
Enter the pet type('c' for cat and 'd' for dog): d
Enter Pet: tommy
Enter the pet type('c' for cat and 'd' for dog): d
Enter Pet: micky
Enter the pet type('c' for cat and 'd' for dog): d
Enter Pet: minnie
Enter the pet type('c' for cat and 'd' for dog): c
Enter Pet: molly
Enter the pet type('c' for cat and 'd' for dog): c
Enter Pet: Stop
Exiting...
    Cat List
Cat 1: kitty
Cat 2: minnie
Cat 3: molly
    Dog List
Dog 2: bobby
Dog 3: tommy
Dog 4: micky
```

Q4.
Code:

Pet.java

```
package Q_04;

abstract class Pet {
    private final String name;

    public Pet(String name) {
        this.name = name;
    }

    public String getName(){
        return name;
    }

    //public  abstract String getType();

}
```

Dog.java

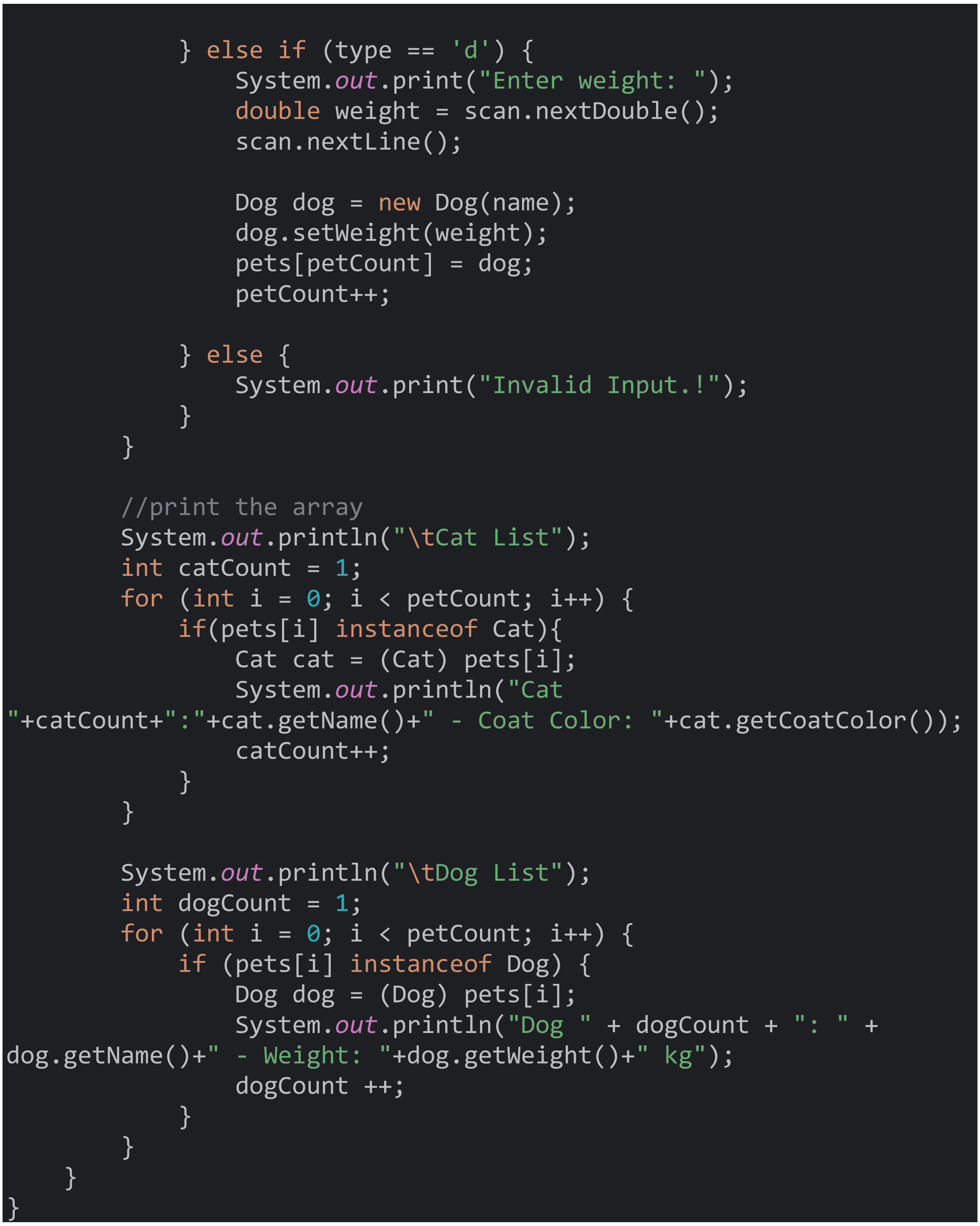
```
package Q_04;

public class Dog extends Pet {
    private double weight;

    public Dog(String name){
        super(name);
    }

    public double getWeight() {
        return weight;
    }

    public void setWeight(double weight) {
        this.weight = weight;
    }

//    @Override
//    public String getType(){
//        return "Dog";
//    }

}
```

Cat.java

```
package Q_04;

public class Cat extends Pet {
    private String coatColor;

    public Cat(String name){
        super(name);
    }
    public String getCoatColor() {
        return coatColor;
    }

    public void setCoatColor(String coatColor) {
        this.coatColor = coatColor;
    }

//    @Override
//    public String getType(){
//        return "Cat";
//    }


}
```

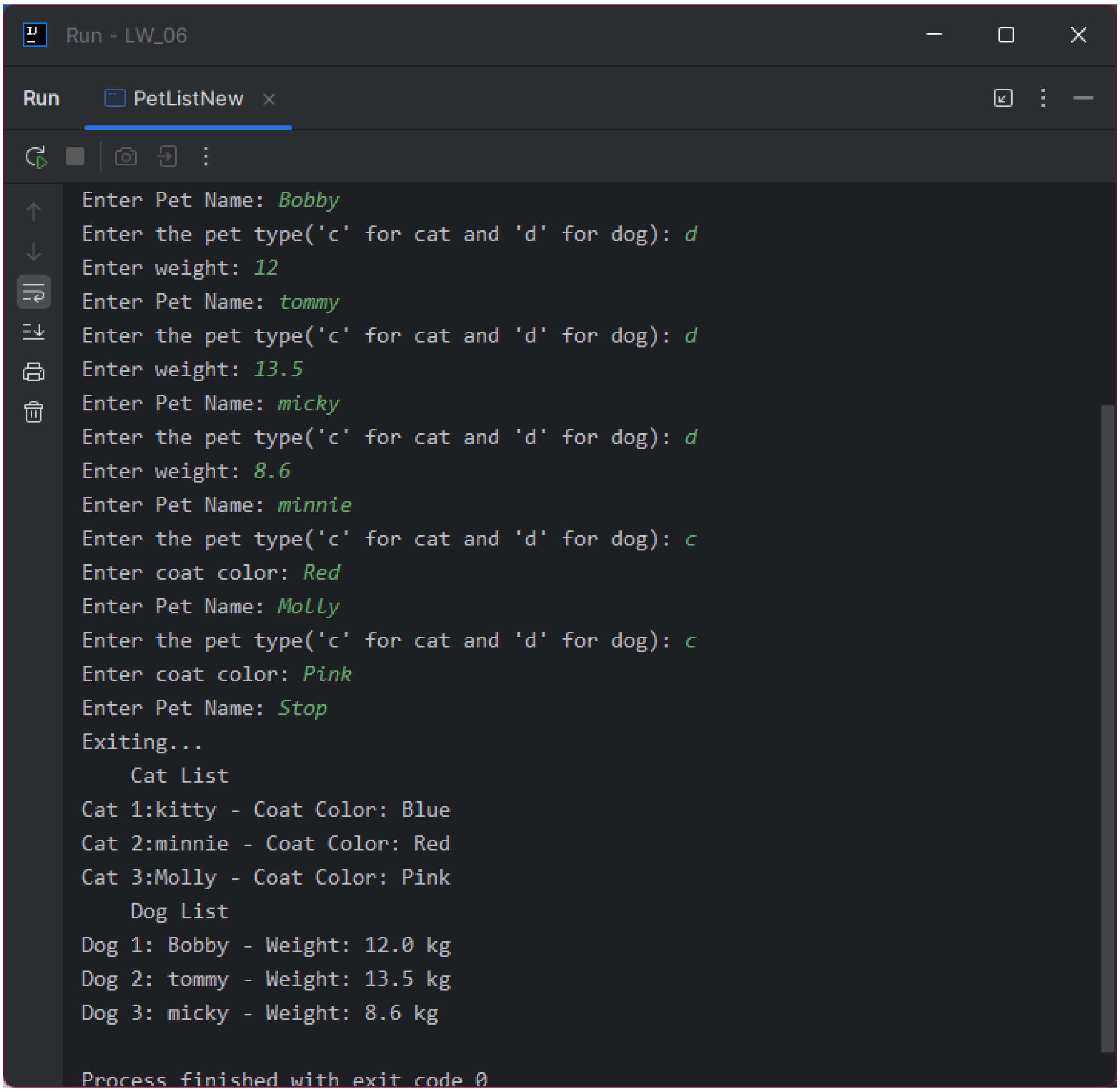
PetListNew.java

```
package Q_04;

import java.util.Scanner;

public class PetListNew {
    public static final int MAX_VALUE = 10;

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        Pet[] pets = new Pet[MAX_VALUE];
        int petCount = 0;

        //input the data to array
        System.out.println("First Enter [Pet's Name],then [Pet
Type]. Enter 'STOP' to finish.");

        while (true) {
            System.out.print("Enter Pet Name: ");
            String name = scan.nextLine();

            if (name.equalsIgnoreCase("stop")) {
                System.out.println("Exiting...");
                break;
            }

            System.out.print("Enter the pet type('c' for cat and
'd' for dog): ");
            char type = scan.nextLine().charAt(0);

            if (type == 'c') {
                System.out.print("Enter coat color: ");
                String coatColor = scan.nextLine();

                Cat cat = new Cat(name);
                cat.setCoatColor(coatColor);
                pets[petCount] = cat;
                petCount++;
```



```
            } else if (type == 'd') {
                System.out.print("Enter weight: ");
                double weight = scan.nextDouble();
                scan.nextLine();

                Dog dog = new Dog(name);
                dog.setWeight(weight);
                pets[petCount] = dog;
                petCount++;

            } else {
                System.out.print("Invalid Input.!");
            }
        }

        //print the array
        System.out.println("\tCat List");
        int catCount = 1;
        for (int i = 0; i < petCount; i++) {
            if(pets[i] instanceof Cat){
                Cat cat = (Cat) pets[i];
                System.out.println("Cat 
"+catCount+":"+cat.getName()+" - Coat Color: "+cat.getCoatColor());
                catCount++;
            }
        }

        System.out.println("\tDog List");
        int dogCount = 1;
        for (int i = 0; i < petCount; i++) {
            if (pets[i] instanceof Dog) {
                Dog dog = (Dog) pets[i];
                System.out.println("Dog " + dogCount + ": " + 
dog.getName()+" - Weight: "+dog.getWeight()+" kg");
                dogCount ++;
            }
        }
    }
}
```

Output:



```
Enter Pet Name: Bobby
Enter the pet type('c' for cat and 'd' for dog): d
Enter weight: 12
Enter Pet Name: tommy
Enter the pet type('c' for cat and 'd' for dog): d
Enter weight: 13.5
Enter Pet Name: micky
Enter the pet type('c' for cat and 'd' for dog): d
Enter weight: 8.6
Enter Pet Name: minnie
Enter the pet type('c' for cat and 'd' for dog): c
Enter coat color: Red
Enter Pet Name: Molly
Enter the pet type('c' for cat and 'd' for dog): c
Enter coat color: Pink
Enter Pet Name: Stop
Exiting...
    Cat List
Cat 1:kitty - Coat Color: Blue
Cat 2:minnie - Coat Color: Red
Cat 3:Molly - Coat Color: Pink
    Dog List
Dog 1: Bobby - Weight: 12.0 kg
Dog 2: tommy - Weight: 13.5 kg
Dog 3: micky - Weight: 8.6 kg

Process finished with exit code 0
```

Q5.
Code:

PetQ5.java

```java
package Q_05;

abstract class PetQ5 {
    private final String name;

    public PetQ5(String name) {
        this.name = name;
    }

    public String getName(){
        return name;
    }

}
```

CatQ5.java

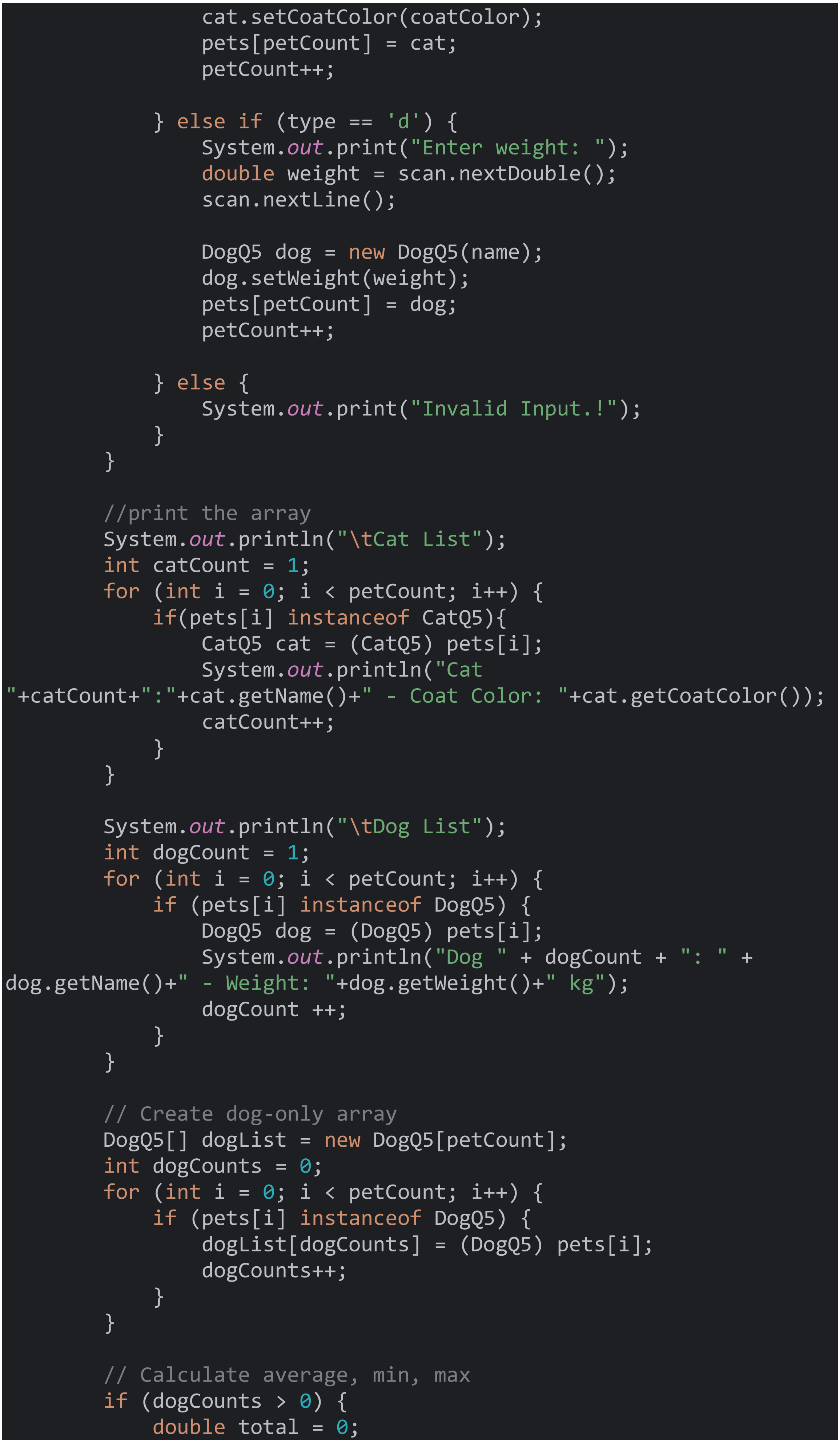
```java
package Q_05;


public class CatQ5 extends PetQ5 {
    private String coatColor;

    public CatQ5(String name){
        super(name);
    }

    public String getCoatColor() {
        return coatColor;
    }

    public void setCoatColor(String coatColor) {
        this.coatColor = coatColor;
    }
}
```

DogQ5.java

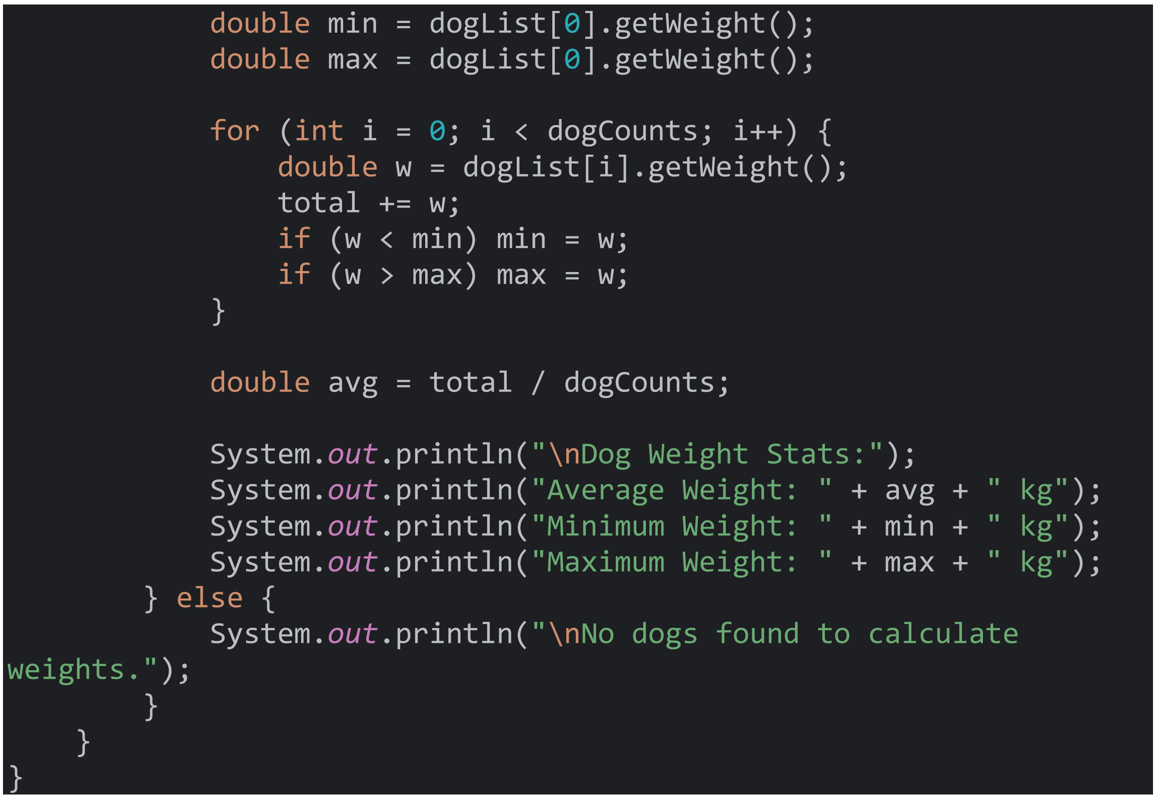
```
package Q_05;

public class DogQ5 extends PetQ5 {
    private double weight;

    public DogQ5(String name){
        super(name);
    }

    public double getWeight() {
        return weight;
    }

    public void setWeight(double weight) {
        this.weight = weight;
    }
}
```

PetListNew.java

```
package Q_05;


import java.util.Scanner;

public class PetListNew {
    public static final int MAX_VALUE = 10;

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        PetQ5[] pets = new PetQ5[MAX_VALUE];
        int petCount = 0;

        //input the data to array
        System.out.println("First Enter [Pet's Name],then [Pet 
Type]. Enter 'STOP' to finish.");

        while (true) {
            System.out.print("Enter Pet Name: ");
            String name = scan.nextLine();

            if (name.equalsIgnoreCase("stop")) {
                System.out.println("Exiting...");
                break;
            }

            System.out.print("Enter the pet type('c' for cat and 
'd' for dog): ");
            char type = scan.nextLine().charAt(0);

            if (type == 'c') {
                System.out.print("Enter coat color: ");
                String coatColor = scan.nextLine();

                CatQ5 cat = new CatQ5(name);
```

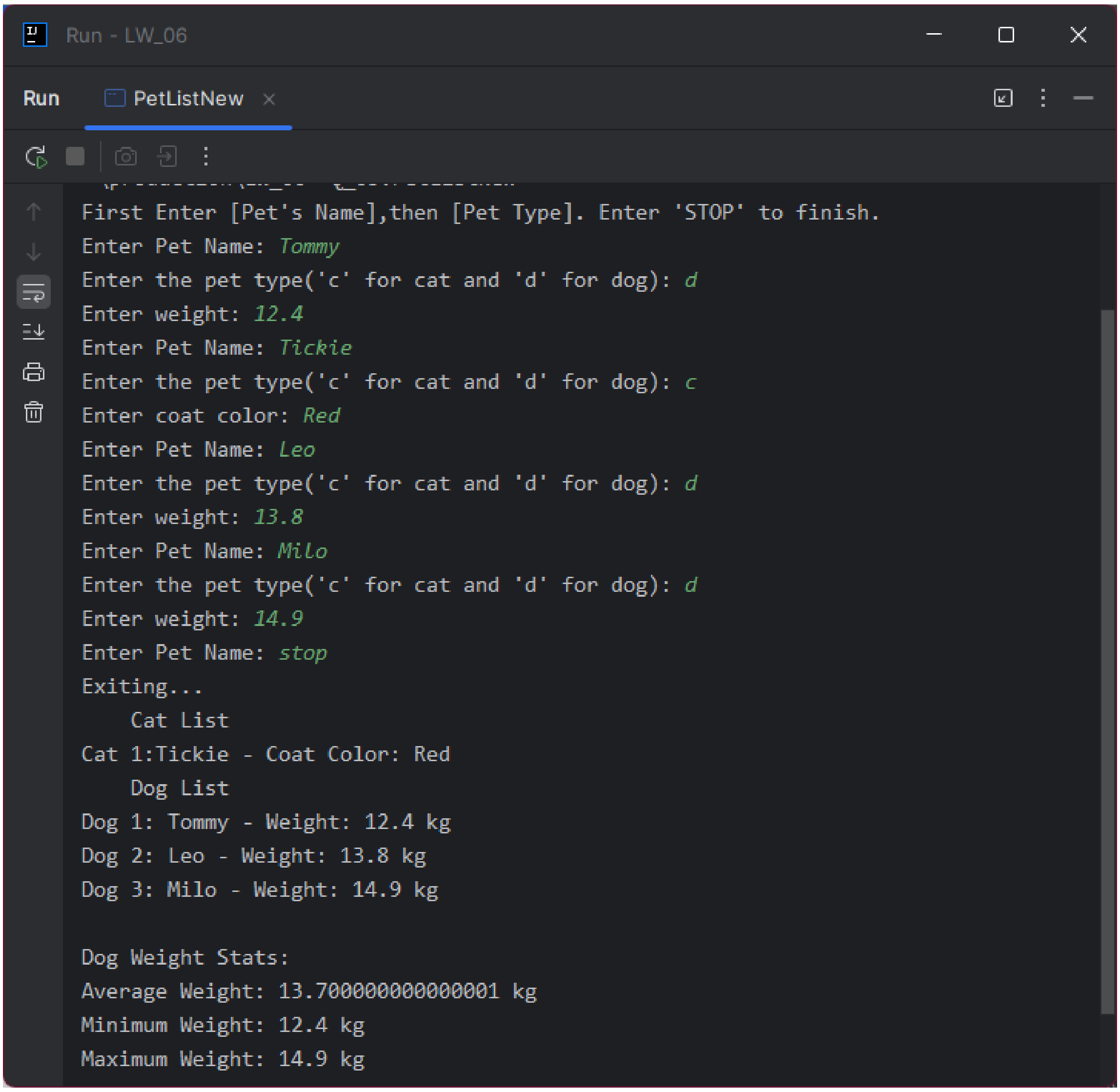


```
                cat.setCoatColor(coatColor);
                pets[petCount] = cat;
                petCount++;

            } else if (type == 'd') {
                System.out.print("Enter weight: ");
                double weight = scan.nextDouble();
                scan.nextLine();

                DogQ5 dog = new DogQ5(name);
                dog.setWeight(weight);
                pets[petCount] = dog;
                petCount++;

            } else {
                System.out.print("Invalid Input.!");
            }
        }

        //print the array
        System.out.println("\tCat List");
        int catCount = 1;
        for (int i = 0; i < petCount; i++) {
            if(pets[i] instanceof CatQ5){
                CatQ5 cat = (CatQ5) pets[i];
                System.out.println("Cat 
"+catCount+":"+cat.getName()+" - Coat Color: "+cat.getCoatColor());
                catCount++;
            }
        }

        System.out.println("\tDog List");
        int dogCount = 1;
        for (int i = 0; i < petCount; i++) {
            if (pets[i] instanceof DogQ5) {
                DogQ5 dog = (DogQ5) pets[i];
                System.out.println("Dog " + dogCount + ": " + 
dog.getName()+" - Weight: "+dog.getWeight()+" kg");
                dogCount ++;
            }
        }

        // Create dog-only array
        DogQ5[] dogList = new DogQ5[petCount];
        int dogCounts = 0;
        for (int i = 0; i < petCount; i++) {
            if (pets[i] instanceof DogQ5) {
                dogList[dogCounts] = (DogQ5) pets[i];
                dogCounts++;
            }
        }

        // Calculate average, min, max
        if (dogCounts > 0) {
            double total = 0;
```



```
            double min = dogList[0].getWeight();
            double max = dogList[0].getWeight();

            for (int i = 0; i < dogCounts; i++) {
                double w = dogList[i].getWeight();
                total += w;
                if (w < min) min = w;
                if (w > max) max = w;
            }

            double avg = total / dogCounts;

            System.out.println("\nDog Weight Stats:");
            System.out.println("Average Weight: " + avg + " kg");
            System.out.println("Minimum Weight: " + min + " kg");
            System.out.println("Maximum Weight: " + max + " kg");
        } else {
            System.out.println("\nNo dogs found to calculate 
weights.");
        }
    }
}
```

Output:



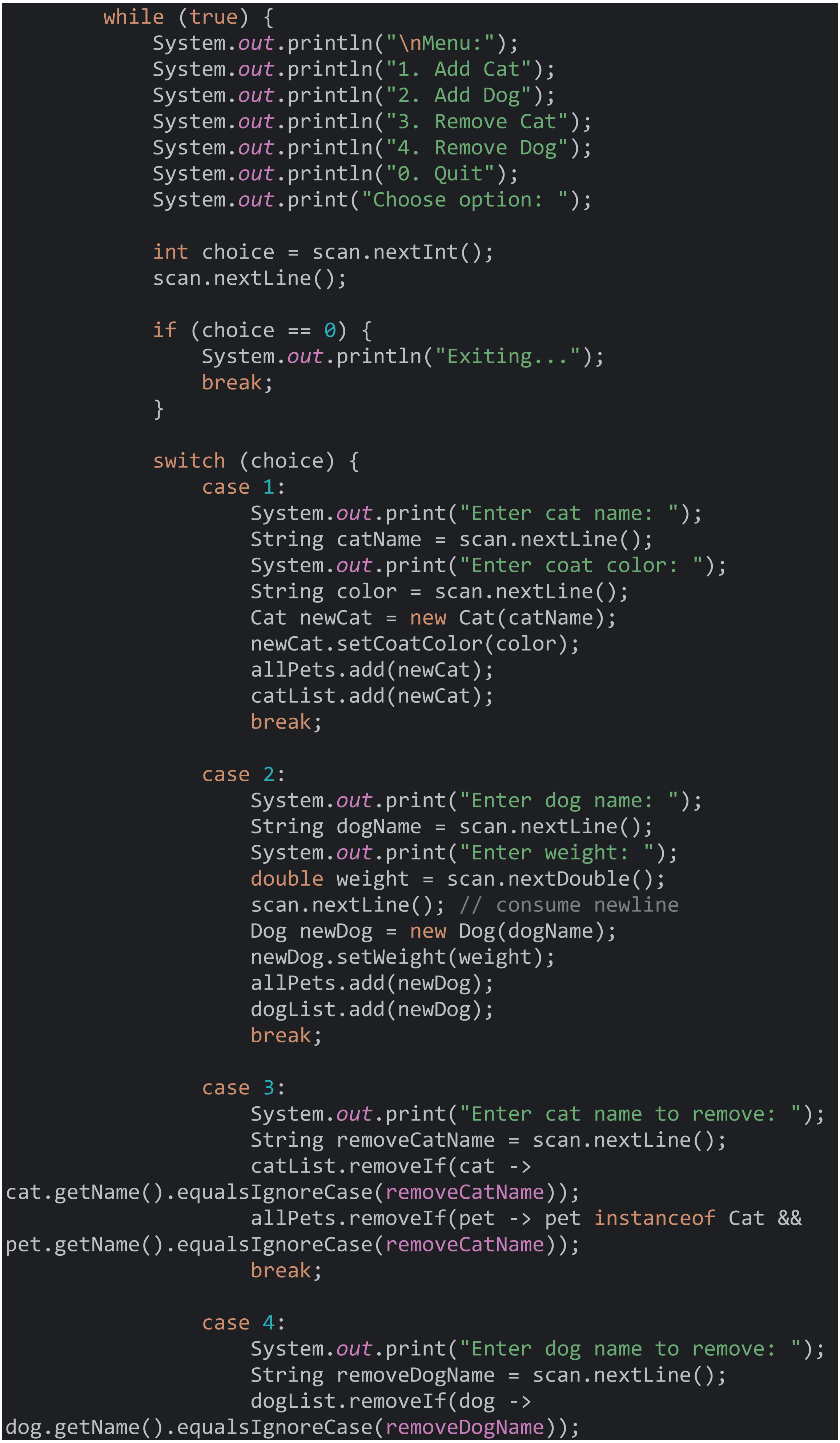
```
First Enter [Pet's Name],then [Pet Type]. Enter 'STOP' to finish.
Enter Pet Name: Tommy
Enter the pet type('c' for cat and 'd' for dog): d
Enter weight: 12.4
Enter Pet Name: Tickie
Enter the pet type('c' for cat and 'd' for dog): c
Enter coat color: Red
Enter Pet Name: Leo
Enter the pet type('c' for cat and 'd' for dog): d
Enter weight: 13.8
Enter Pet Name: Milo
Enter the pet type('c' for cat and 'd' for dog): d
Enter weight: 14.9
Enter Pet Name: stop
Exiting...
    Cat List
Cat 1:Tickie - Coat Color: Red
    Dog List
Dog 1: Tommy - Weight: 12.4 kg
Dog 2: Leo - Weight: 13.8 kg
Dog 3: Milo - Weight: 14.9 kg

Dog Weight Stats:
Average Weight: 13.700000000000001 kg
Minimum Weight: 12.4 kg
Maximum Weight: 14.9 kg
```

Q6.
Code:

Cat.java, Dog.java, Pet.java are same as above.

PetListMenu.java

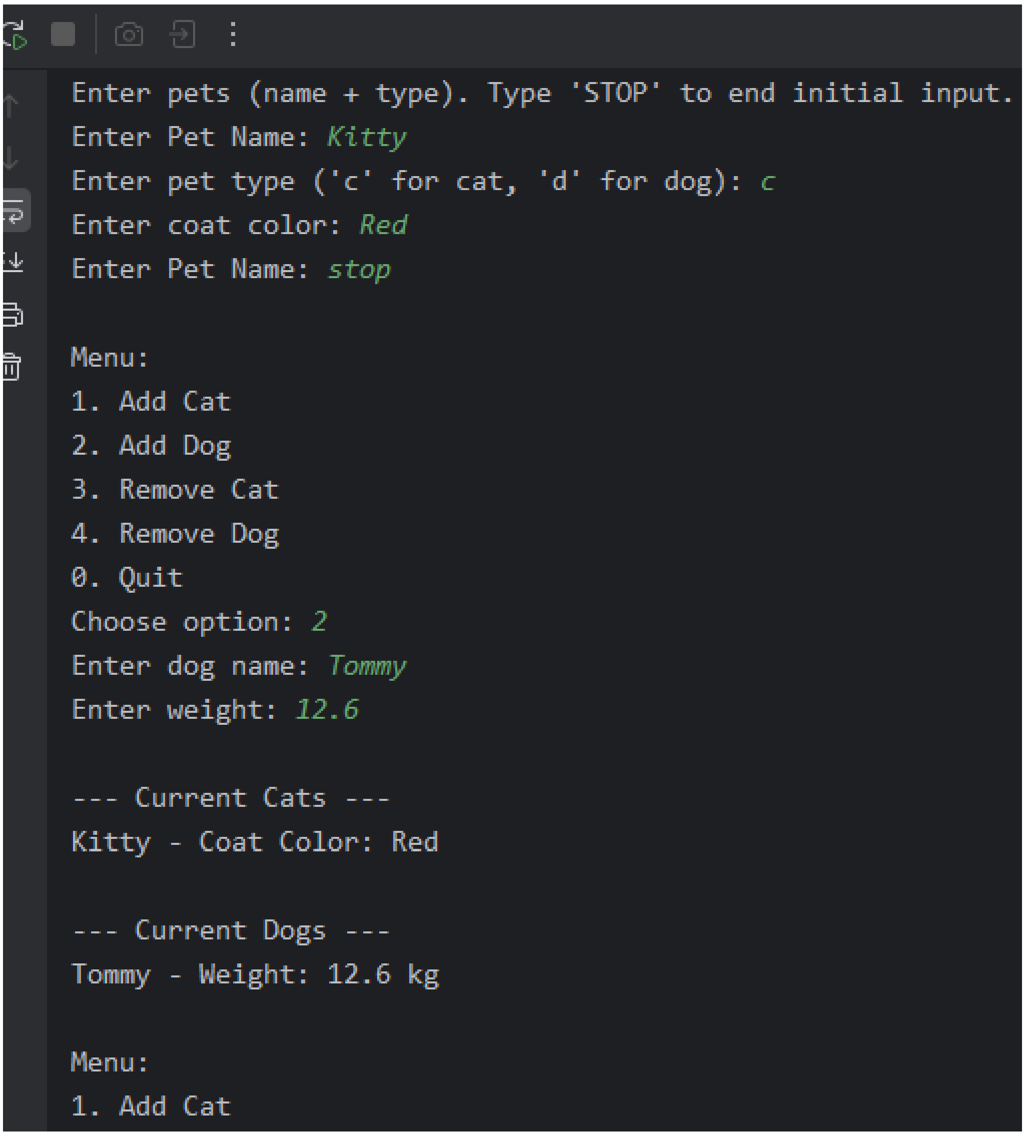
```
package Q_06;

import java.util.ArrayList;
import java.util.Scanner;

public class PetListMenu {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        ArrayList<Pet> allPets = new ArrayList<>();
        ArrayList<Cat> catList = new ArrayList<>();
        ArrayList<Dog> dogList = new ArrayList<>();

        System.out.println("Enter pets (name + type). Type 'STOP'
to end initial input.");
        while (true) {
            System.out.print("Enter Pet Name: ");
            String name = scan.nextLine();
            if (name.equalsIgnoreCase("stop")) break;

            System.out.print("Enter pet type ('c' for cat, 'd' for
dog): ");
            char type = scan.nextLine().toLowerCase().charAt(0);

            if (type == 'c') {
                System.out.print("Enter coat color: ");
                String color = scan.nextLine();

                Cat cat = new Cat(name);
                cat.setCoatColor(color);

                allPets.add(cat);
                catList.add(cat);

            } else if (type == 'd') {
                System.out.print("Enter weight: ");
                double weight = scan.nextDouble();
                scan.nextLine();

                Dog dog = new Dog(name);
                dog.setWeight(weight);

                allPets.add(dog);
                dogList.add(dog);

            } else {
                System.out.println("Invalid type.");
            }
        }
```



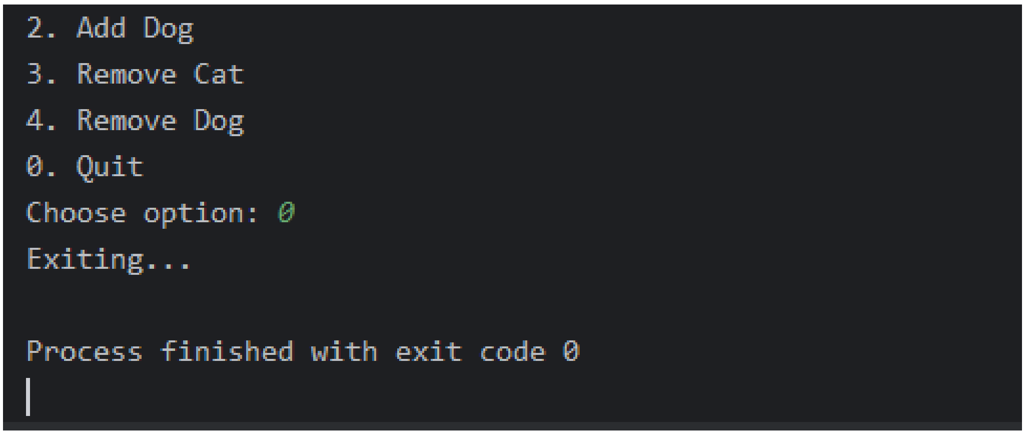
```
        while (true) {
            System.out.println("\nMenu:");
            System.out.println("1. Add Cat");
            System.out.println("2. Add Dog");
            System.out.println("3. Remove Cat");
            System.out.println("4. Remove Dog");
            System.out.println("0. Quit");
            System.out.print("Choose option: ");

            int choice = scan.nextInt();
            scan.nextLine();

            if (choice == 0) {
                System.out.println("Exiting...");
                break;
            }

            switch (choice) {
                case 1:
                    System.out.print("Enter cat name: ");
                    String catName = scan.nextLine();
                    System.out.print("Enter coat color: ");
                    String color = scan.nextLine();
                    Cat newCat = new Cat(catName);
                    newCat.setCoatColor(color);
                    allPets.add(newCat);
                    catList.add(newCat);
                    break;

                case 2:
                    System.out.print("Enter dog name: ");
                    String dogName = scan.nextLine();
                    System.out.print("Enter weight: ");
                    double weight = scan.nextDouble();
                    scan.nextLine(); // consume newline
                    Dog newDog = new Dog(dogName);
                    newDog.setWeight(weight);
                    allPets.add(newDog);
                    dogList.add(newDog);
                    break;

                case 3:
                    System.out.print("Enter cat name to remove: ");
                    String removeCatName = scan.nextLine();
                    catList.removeIf(cat ->
cat.getName().equalsIgnoreCase(removeCatName));
                    allPets.removeIf(pet -> pet instanceof Cat &&
pet.getName().equalsIgnoreCase(removeCatName));
                    break;

                case 4:
                    System.out.print("Enter dog name to remove: ");
                    String removeDogName = scan.nextLine();
                    dogList.removeIf(dog ->
dog.getName().equalsIgnoreCase(removeDogName));
```

```
                    allPets.removeIf(pet -> pet instanceof Dog &&
pet.getName().equalsIgnoreCase(removeDogName));
                    break;

                default:
                    System.out.println("Invalid option!");
            }

            System.out.println("\n--- Current Cats ---");
            for (Cat c : catList) {
                System.out.println(c.getName() + " - Coat Color: "
+ c.getCoatColor());
            }

            System.out.println("\n--- Current Dogs ---");
            for (Dog d : dogList) {
                System.out.println(d.getName() + " - Weight: " +
d.getWeight() + " kg");
            }
        }

        scan.close();
    }
}
```

Output:



```
Enter pets (name + type). Type 'STOP' to end initial input.
Enter Pet Name: Kitty
Enter pet type ('c' for cat, 'd' for dog): c
Enter coat color: Red
Enter Pet Name: stop

Menu:
1. Add Cat
2. Add Dog
3. Remove Cat
4. Remove Dog
0. Quit
Choose option: 2
Enter dog name: Tommy
Enter weight: 12.6

--- Current Cats ---
Kitty - Coat Color: Red

--- Current Dogs ---
Tommy - Weight: 12.6 kg

Menu:
1. Add Cat
```



```
2. Add Dog
3. Remove Cat
4. Remove Dog
0. Quit
Choose option: 0
Exiting...

Process finished with exit code 0
```