

## Lab worksheet 6: Inheritance & Polymorphism

Q1.

Code:

### **Pet class**

```
package Q_01;
```

```
public 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 little pet.";  
    }  
}
```

### **Dog class**

```
package Q_01;
```

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

## Cat class

```
package Q_01;

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

## Main class

```
package Q_01;

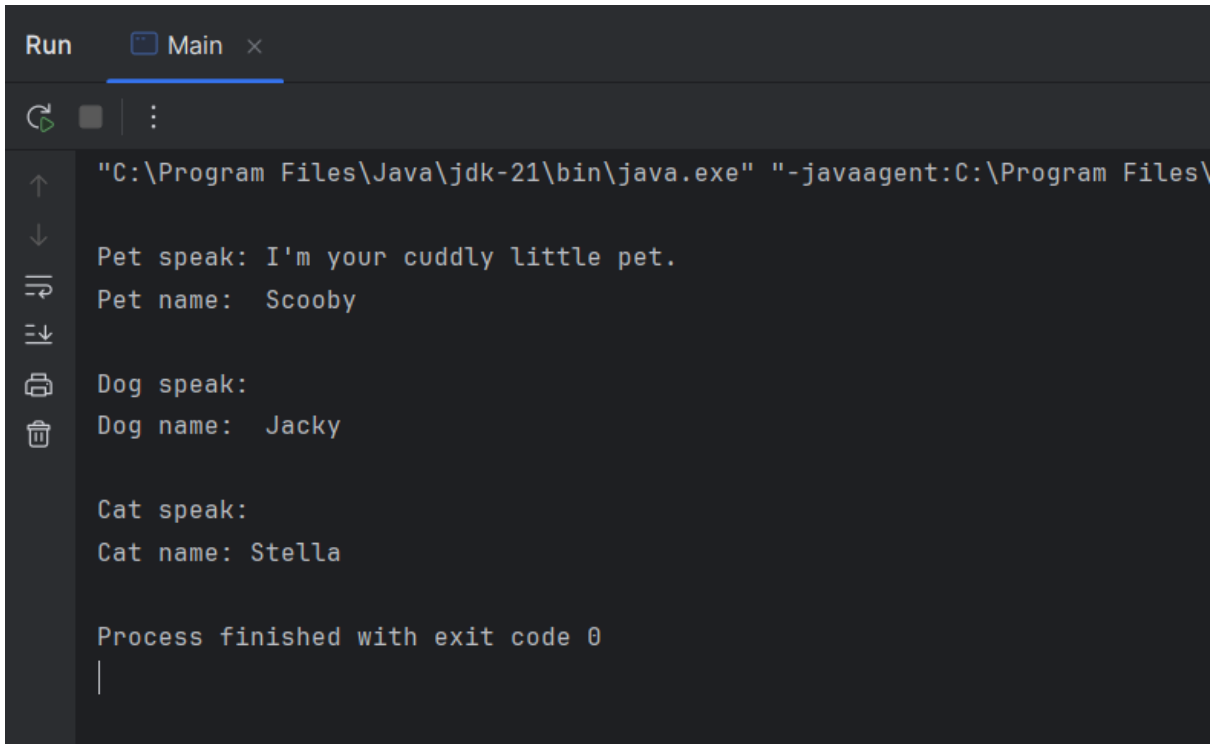
public class Main {
    public static void main(String[] args) {
        Pet pet = new Pet();
        System.out.println("\nPet speak: "+pet.speak());
        pet.setName("Scooby");
        System.out.println("Pet name: "+pet.getName());

        Dog dog = new Dog();
        dog.setName("Jacky");
        System.out.println("\nDog speak: " +dog.speak());
        System.out.println("Dog name: "+dog.getName());

        Cat cat = new Cat();
        cat.setName("Stella");
        System.out.println("\nCat speak: "+cat.speak());
        System.out.println("Cat name: "+cat.getName());

    }
}
```

Output:



```
Run Main x
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\
Pet speak: I'm your cuddly little pet.
Pet name: Scooby

Dog speak:
Dog name: Jacky

Cat speak:
Cat name: Stella

Process finished with exit code 0
```

Q2.

Code:

```
package Q_02;

import Q_01.Cat;
import Q_01.Dog;
import Q_01.Pet;
import java.util.Scanner;

public class Q_02 {
    public static void main(String[] args){
        Scanner scanner = new Scanner(System.in);
        Pet[] pets = new Pet[100]; // Array to hold up to
100 pets
        int count = 0;
```

```

        while (true) {
            System.out.print("Enter pet name (or 'STOP'
to finish): ");
            String name = scanner.nextLine();
            if (name.equalsIgnoreCase("STOP")) {
                break;
            }

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

            Pet pet;
            if (type == 'c') {
                pet = new Cat();
            } else if (type == 'd') {
                pet = new Dog();
            } else {
                System.out.println("Invalid type. Please
enter 'c' for cat or 'd' for dog.");
                continue; // Skip to the next iteration
            }

            pet.setName(name);
            pets[count] = pet;
            count++;

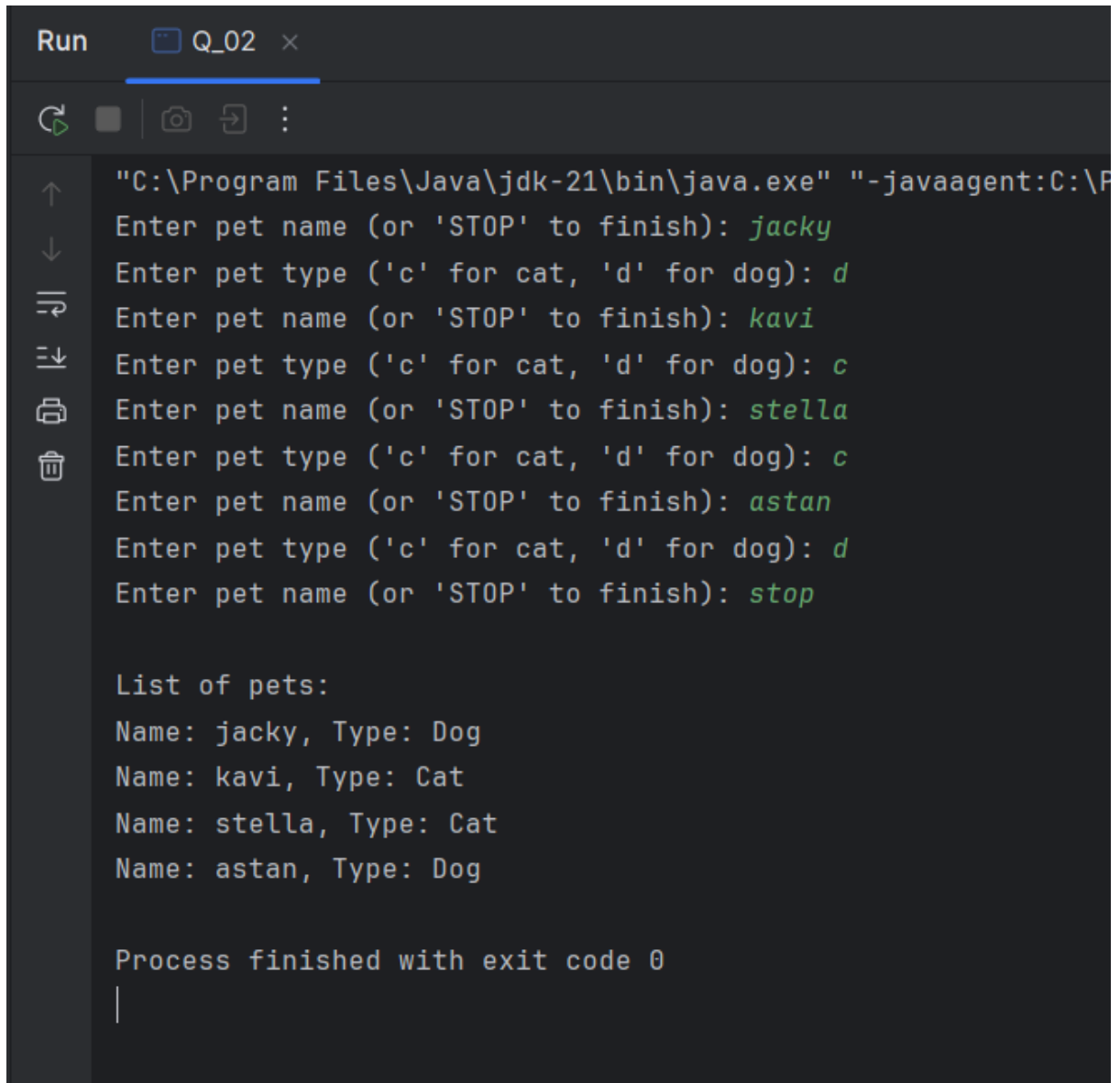
        }

        System.out.println("\nList of pets:");
        for (int i = 0; i < count; i++) {
            Pet pet = pets[i];
            String type = (pet instanceof Cat) ? "Cat" :
"Dog";
            System.out.println("Name: " + pet.getName() + ",
Type: " + type);
        }

        scanner.close();
    }
}

```

Output:



```
Run Q_02 x
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\P
Enter pet name (or 'STOP' to finish): jacky
Enter pet type ('c' for cat, 'd' for dog): d
Enter pet name (or 'STOP' to finish): kavi
Enter pet type ('c' for cat, 'd' for dog): c
Enter pet name (or 'STOP' to finish): stella
Enter pet type ('c' for cat, 'd' for dog): c
Enter pet name (or 'STOP' to finish): astan
Enter pet type ('c' for cat, 'd' for dog): d
Enter pet name (or 'STOP' to finish): stop

List of pets:
Name: jacky, Type: Dog
Name: kavi, Type: Cat
Name: stella, Type: Cat
Name: astan, Type: Dog

Process finished with exit code 0
|
```

Q3.

Code:

```
package Q_03;

import Q_01.Cat;
import Q_01.Dog;
import Q_01.Pet;

import java.util.Scanner;

public class Q_03 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Pet[] pets = new Pet[100]; // Array to hold up to 100
        pets
        int count = 0;

        while (true) {
            System.out.print("Enter pet name (or 'STOP' to
            finish): ");
            String name = scanner.nextLine();
            if (name.equalsIgnoreCase("STOP")) {
                break;
            }

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

            Pet pet;
            if (type == 'c') {
                pet = new Cat();
            } else if (type == 'd') {
                pet = new Dog();
            } else {
                System.out.println("Invalid type. Please
                enter 'c' for cat or 'd' for dog.");
                continue; // Skip to the next iteration
            }
        }
    }
}
```

```

        pet.setName(name);
        pets[count] = pet;
        count++;
        // Check if the array is full
        if (count >= pets.length) {
            System.out.println("The pet array is full. No
more pets can be added.");
            break;
        }
    }
    // Output the list of pets after input is complete
    System.out.println("\nList of pets:\n");

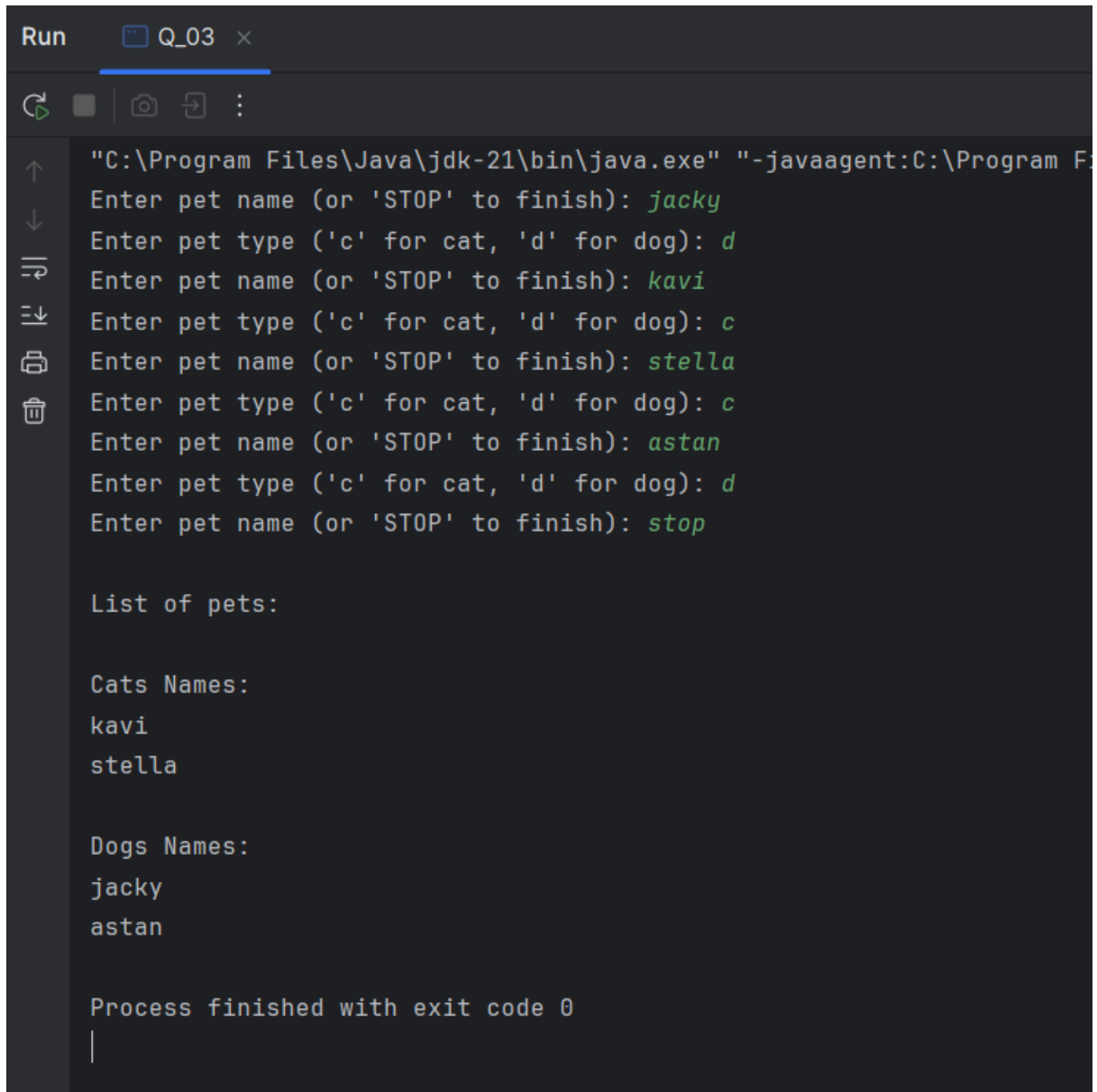
    // Print all cats first
    System.out.println("Cats Names:");
    boolean foundCat = false;
    for (int i = 0; i < count; i++) {
        if (pets[i] instanceof Cat) {
            System.out.println(pets[i].getName());
            foundCat = true;
        }
    }
    if (!foundCat) {
        System.out.println("No cats entered.");
    }

    // Print all dogs next
    System.out.println("\nDogs Names:");
    boolean foundDog = false;
    for (int i = 0; i < count; i++) {
        if (pets[i] instanceof Dog) {
            System.out.println(pets[i].getName());
            foundDog = true;
        }
    }
    if (!foundDog) {
        System.out.println("No dogs entered.");
    }

    scanner.close(); // Close the scanner
}
}

```

Output:



```
Run Q_03 x
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program F
Enter pet name (or 'STOP' to finish): jacky
Enter pet type ('c' for cat, 'd' for dog): d
Enter pet name (or 'STOP' to finish): kavi
Enter pet type ('c' for cat, 'd' for dog): c
Enter pet name (or 'STOP' to finish): stella
Enter pet type ('c' for cat, 'd' for dog): c
Enter pet name (or 'STOP' to finish): astan
Enter pet type ('c' for cat, 'd' for dog): d
Enter pet name (or 'STOP' to finish): stop

List of pets:

Cats Names:
kavi
stella

Dogs Names:
jacky
astan

Process finished with exit code 0
|
```



Q4.

Code:

**Updated dog class**

```
package Q_01;

public class Dog extends Pet {
    private double weight;

    public double getWeight() {
        return weight;
    }

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

    @Override
    public String speak() {
        return "";
    }
}
```

**Updated cat class**

```
package Q_01;

public class Cat extends Pet {
    private String coatColor;

    public String getCoatColor() {
        return coatColor;
    }

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

    @Override
    public String speak() {
        return "";
    }
}
```

**Main class(main class name Q\_04)**

```
package Q_04;

import Q_01.Cat;
import Q_01.Dog;
import Q_01.Pet;

import java.util.Scanner;

public class Q_04 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Pet[] pets = new Pet[100];
        int count = 0;

        while (true) {
            System.out.print("Enter pet name (or 'STOP' to
finish): ");
            String name = scanner.nextLine();
            if (name.equalsIgnoreCase("STOP")) {
                break;
            }

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

            Pet pet;

            if (type == 'c') {
                pet = new Cat();
                System.out.print("Enter coat color: ");
                String color = scanner.nextLine();
                ((Cat) pet).setCoatColor(color);
            } else if (type == 'd') {
                pet = new Dog();
                double weight = -1;
                while (weight <= 0) {
                    System.out.print("Enter weight (positive
number): ");
                    String input = scanner.nextLine();
```

```

        try {
            weight = Double.parseDouble(input);
            if (weight <= 0) {
                System.out.println("Weight must
be greater than 0.");
            }
        } catch (NumberFormatException e) {
            System.out.println("Please enter a
valid number.");
        }
    }
    ((Dog) pet).setWeight(weight);
} else {
    System.out.println("Please enter 'c' or 'd'
only.");
    continue;
}

pet.setName(name);
pets[count] = pet;
count++;

if (count >= pets.length) {
    System.out.println("Cannot add more pets.");
    break;
}
}

System.out.println("\nPet List:");

System.out.println("\nCats:");
boolean foundCat = false;
for (int i = 0; i < count; i++) {
    if (pets[i] instanceof Cat) {
        Cat cat = (Cat) pets[i];
        System.out.println("Name: " + cat.getName() +
" , Type:Cat "+ " , Coat Color: " + cat.getCoatColor());
        foundCat = true;
    }
}
if (!foundCat) {
    System.out.println("No cats.");
}

```

```
        System.out.println("\nDogs:");
        boolean foundDog = false;
        for (int i = 0; i < count; i++) {
            if (pets[i] instanceof Dog) {
                Dog dog = (Dog) pets[i];
                System.out.println("Name: " + dog.getName() +
" , Type:Dog " + ", Weight: " + dog.getWeight());
                foundDog = true;
            }
        }
        if (!foundDog) {
            System.out.println("No dogs.");
        }

        scanner.close();
    }
}
```

## Output:

```
Run Q_04 x
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\P
Enter pet name (or 'STOP' to finish): jacky
Enter pet type ('c' for cat, 'd' for dog): d
Enter weight (positive number): 20
Enter pet name (or 'STOP' to finish): stella
Enter pet type ('c' for cat, 'd' for dog): c
Enter coat color: black
Enter pet name (or 'STOP' to finish): astan
Enter pet type ('c' for cat, 'd' for dog): d
Enter weight (positive number): 15
Enter pet name (or 'STOP' to finish): kavi
Enter pet type ('c' for cat, 'd' for dog): c
Enter coat color: white
Enter pet name (or 'STOP' to finish): stop

Pet List:

Cats:
Name: stella , Type:Cat , Coat Color: black
Name: kavi , Type:Cat , Coat Color: white

Dogs:
Name: jacky , Type:Dog , Weight: 20.0
Name: astan , Type:Dog , Weight: 15.0

Process finished with exit code 0
|
```

Q5.

Code:

```
package Q_05;

import Q_01.Cat;
import Q_01.Dog;
import Q_01.Pet;

import java.util.Scanner;

public class Q_05 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Pet[] pets = new Pet[100];
        int count = 0;

        while (true) {
            System.out.print("Enter pet name (or 'STOP' to finish): ");
            String name = scanner.nextLine();
            if (name.equalsIgnoreCase("STOP")) {
                break;
            }

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

            Pet pet;

            if (type == 'c') {
                pet = new Cat();
                System.out.print("Enter coat color: ");
                String color = scanner.nextLine();
                ((Cat) pet).setCoatColor(color);
            } else if (type == 'd') {
                pet = new Dog();
                double weight = -1;
                while (weight <= 0) {
                    System.out.print("Enter weight (positive number): ");
```

```

        String input = scanner.nextLine();
        try {
            weight = Double.parseDouble(input);
            if (weight <= 0) {
                System.out.println("Weight must be
greater than 0.");
            }
        } catch (NumberFormatException e) {
            System.out.println("Please enter a valid
number.");
        }
    }
    ((Dog) pet).setWeight(weight);
} else {
    System.out.println("Please enter 'c' or 'd'
only.");
    continue;
}

pet.setName(name);
pets[count] = pet;
count++;

if (count >= pets.length) {
    System.out.println("Cannot add more pets.");
    break;
}
}

System.out.println("\nPet List:");

System.out.println("\nCats:");
boolean foundCat = false;
for (int i = 0; i < count; i++) {
    if (pets[i] instanceof Cat) {
        Cat cat = (Cat) pets[i];
        System.out.println("Name: " + cat.getName() + " ,
Type:Cat " + ", Coat Color: " + cat.getCoatColor());
        foundCat = true;
    }
}
if (!foundCat) {
    System.out.println("No cats.");
}

```

```
}

System.out.println("\nDogs:");
boolean foundDog = false;
for (int i = 0; i < count; i++) {
    if (pets[i] instanceof Dog) {
        Dog dog = (Dog) pets[i];
        System.out.println("Name: " + dog.getName() + " ,
Type:Dog " + ", Weight: " + dog.getWeight());
        foundDog = true;
    }
}

if (!foundDog) {
    System.out.println("No dogs.");
}

// Create Dog array and calculate statistics
Dog[] dogArray = new Dog[count];
int dogCount = 0;

for (int i = 0; i < count; i++) {
    if (pets[i] instanceof Dog) {
        dogArray[dogCount] = (Dog) pets[i];
        dogCount++;
    }
}

if (dogCount > 0) {
    double sum = 0;
    double min = dogArray[0].getWeight();
    double max = dogArray[0].getWeight();

    for (int i = 0; i < dogCount; i++) {
        double weight = dogArray[i].getWeight();
        sum += weight;
        if (weight < min) {
            min = weight;
        }
        if (weight > max) {
            max = weight;
        }
    }
}
```

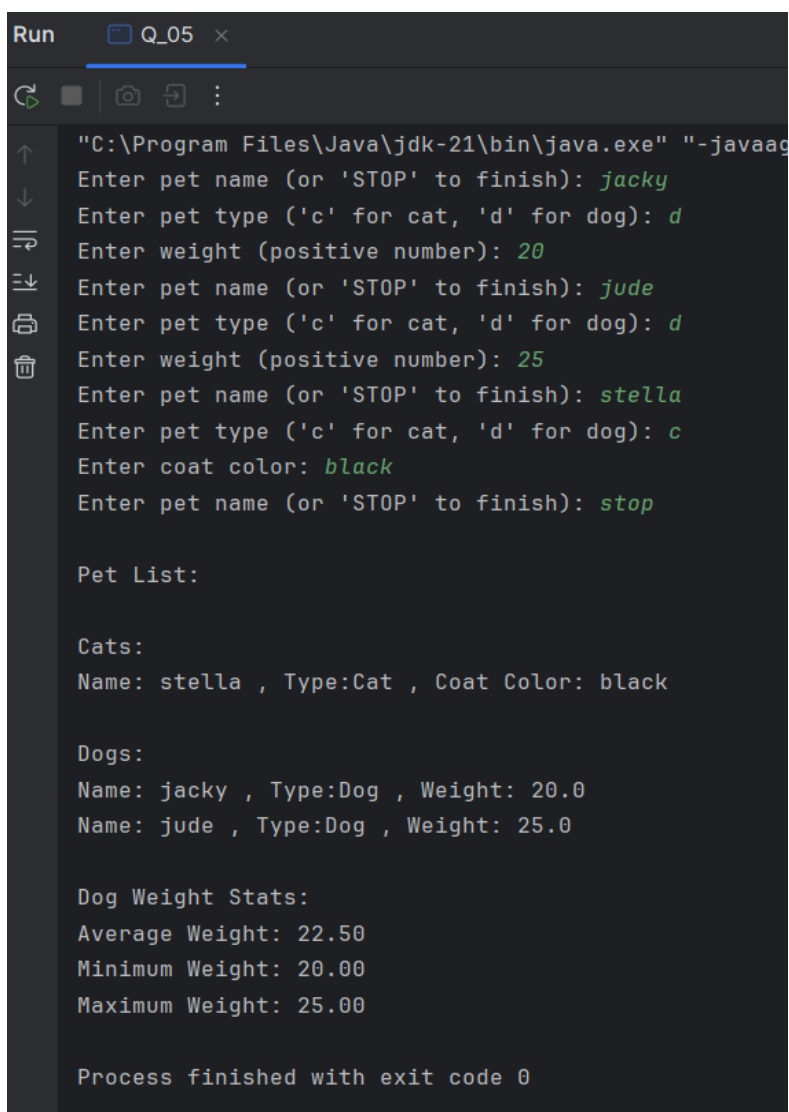


```

        double average = sum / dogCount;
        System.out.printf("\nDog Weight Stats:\n");
        System.out.printf("Average Weight: %.2f\n", average);
        System.out.printf("Minimum Weight: %.2f\n", min);
        System.out.printf("Maximum Weight: %.2f\n", max);
    } else {
        System.out.println("\nNo dogs available to calculate
weight statistics.");
    }
    scanner.close();
}
}

```

## Output:



```

Run    Q_05 x
Enter pet name (or 'STOP' to finish): jacky
Enter pet type ('c' for cat, 'd' for dog): d
Enter weight (positive number): 20
Enter pet name (or 'STOP' to finish): jude
Enter pet type ('c' for cat, 'd' for dog): d
Enter weight (positive number): 25
Enter pet name (or 'STOP' to finish): stella
Enter pet type ('c' for cat, 'd' for dog): c
Enter coat color: black
Enter pet name (or 'STOP' to finish): stop

Pet List:

Cats:
Name: stella , Type:Cat , Coat Color: black

Dogs:
Name: jacky , Type:Dog , Weight: 20.0
Name: jude , Type:Dog , Weight: 25.0

Dog Weight Stats:
Average Weight: 22.50
Minimum Weight: 20.00
Maximum Weight: 25.00

Process finished with exit code 0

```

Q6.

Code:

```
package Q_06;
import Q_01.Cat;
import Q_01.Dog;
import Q_01.Pet;

import java.util.Scanner;

public class Q_06 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        Pet[] pets = new Pet[100];
        Cat[] cats = new Cat[100];
        Dog[] dogs = new Dog[100];

        int petCount = 0;
        int catCount = 0;
        int dogCount = 0;

        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("Enter your choice: ");
            String choice = scanner.nextLine();

            switch (choice) {
                case "1": // Add Cat
                    System.out.print("Enter cat name: ");
                    String catName = scanner.nextLine();
                    System.out.print("Enter coat color: ");
                    String coatColor = scanner.nextLine();

                    Cat newCat = new Cat();
                    newCat.setName(catName);
                    newCat.setCoatColor(coatColor);
```

```

        if (petCount < 100 && catCount < 100) {
            pets[petCount++] = newCat;
            cats[catCount++] = newCat;
            System.out.println("Cat added.");
        } else {
            System.out.println("Cannot add more
cats.");
        }
        break;

    case "2": // Add Dog
        System.out.print("Enter dog name: ");
        String dogName = scanner.nextLine();

        double weight = -1;
        while (weight <= 0) {
            System.out.print("Enter weight
(positive number): ");
            try {
                weight =
Double.parseDouble(scanner.nextLine());
                if (weight <= 0) {
                    System.out.println("Weight
must be greater than 0.");
                }
            } catch (NumberFormatException e) {
                System.out.println("Invalid
number.");
            }
        }

        Dog newDog = new Dog();
        newDog.setName(dogName);
        newDog.setWeight(weight);

        if (petCount < 100 && dogCount < 100) {
            pets[petCount++] = newDog;
            dogs[dogCount++] = newDog;
            System.out.println("Dog added.");
        } else {
            System.out.println("Cannot add more
dogs.");
        }
    }
}

```

```

        }
        break;

    case "3": // Remove Cat
        System.out.print("Enter cat name to
remove: ");

        String removeCatName =
scanner.nextLine();

        boolean catFound = false;
        for (int i = 0; i < catCount; i++) {
            if
(cats[i].getName().equalsIgnoreCase(removeCatName)) {
                for (int j = i; j < catCount - 1;
j++) {
                    cats[j] = cats[j + 1];
                }
                cats[--catCount] = null;
                catFound = true;
                break;
            }
        }

        if (catFound) {
            for (int i = 0; i < petCount; i++) {
                if (pets[i] instanceof Cat &&
pets[i].getName().equalsIgnoreCase(removeCatName)) {
                    for (int j = i; j < petCount
- 1; j++) {
                        pets[j] = pets[j + 1];
                    }
                    pets[--petCount] = null;
                    break;
                }
            }
            System.out.println("Cat removed.");
        } else {
            System.out.println("Cat not found.");
        }
        break;

    case "4": // Remove Dog
        System.out.print("Enter dog name to

```

```

remove: ");

        String removeDogName =
scanner.nextLine();

        boolean dogFound = false;
        for (int i = 0; i < dogCount; i++) {
            if
(dogs[i].getName().equalsIgnoreCase(removeDogName)) {
                for (int j = i; j < dogCount - 1;
j++) {

                    dogs[j] = dogs[j + 1];
                }
                dogs[--dogCount] = null;
                dogFound = true;
                break;
            }
        }

        if (dogFound) {
            for (int i = 0; i < petCount; i++) {
                if (pets[i] instanceof Dog &&
pets[i].getName().equalsIgnoreCase(removeDogName)) {
                    for (int j = i; j < petCount
- 1; j++) {

                        pets[j] = pets[j + 1];
                    }
                    pets[--petCount] = null;
                    break;
                }
            }
            System.out.println("Dog removed.");
        } else {
            System.out.println("Dog not found.");
        }
        break;

    case "0": // Quit and show final list
        System.out.println("\nFinal List of
Cats:");

        if (catCount == 0) {
            System.out.println("No cats.");
        } else {
            for (int i = 0; i < catCount; i++) {

```

```

                System.out.println("Name: " +
cats[i].getName() + " , Type:Cat " + " , Coat Color: " +
cats[i].getCoatColor());
            }
        }

        System.out.println("\nFinal List of
Dogs:");

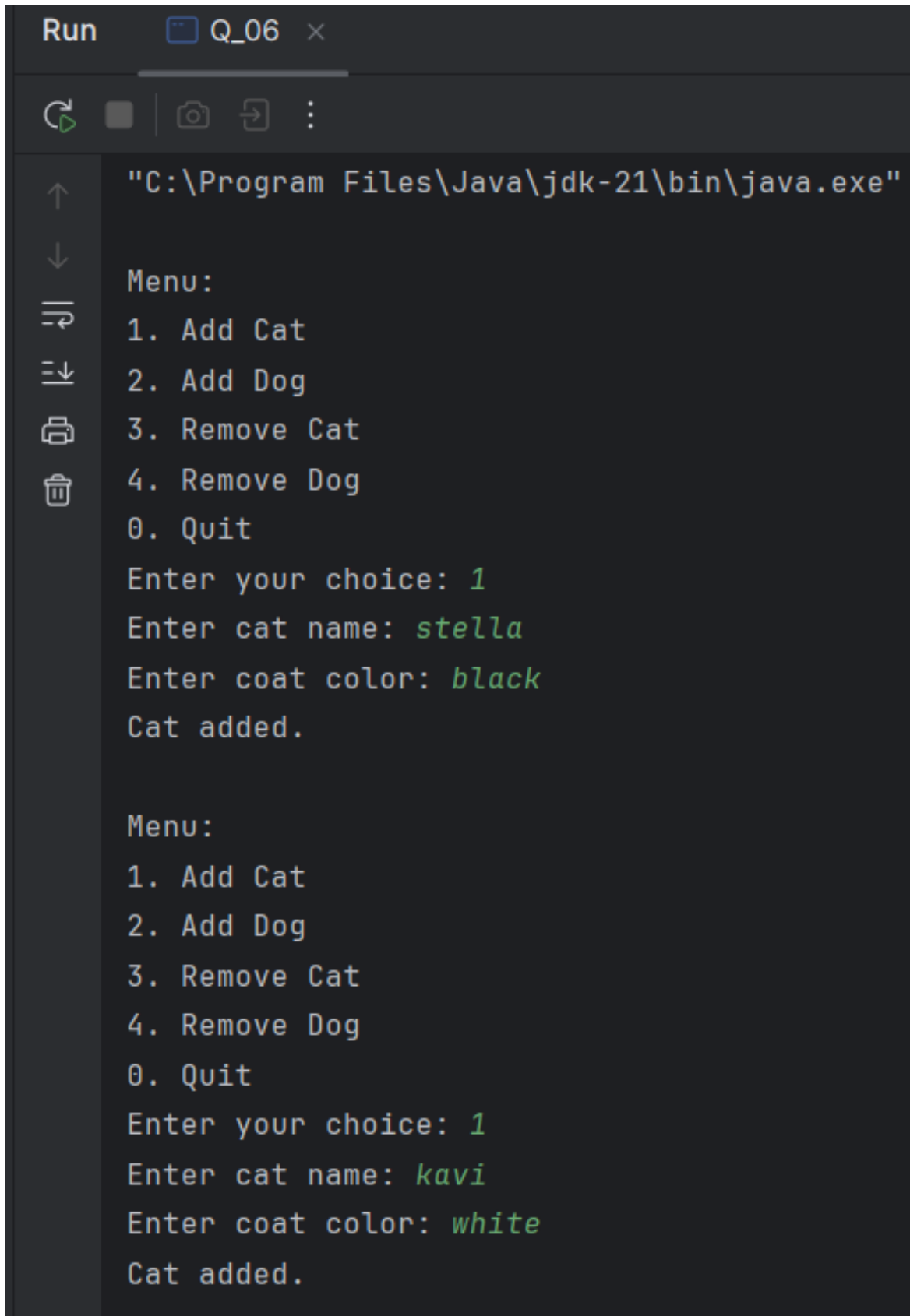
        if (dogCount == 0) {
            System.out.println("No dogs.");
        } else {
            for (int i = 0; i < dogCount; i++) {
                System.out.println("Name: " +
dogs[i].getName() + " , Type: Dog " + " , Weight: " +
dogs[i].getWeight());
            }
        }

        System.out.println("\nExiting program.");
        scanner.close();
        return;

        default:
            System.out.println("Invalid choice.
Please enter a number from 0 to 4.");
        }
    }
}

```

Output:



```
Run Q_06 x
"C:\Program Files\Java\jdk-21\bin\java.exe"

Menu:
1. Add Cat
2. Add Dog
3. Remove Cat
4. Remove Dog
0. Quit
Enter your choice: 1
Enter cat name: stella
Enter coat color: black
Cat added.

Menu:
1. Add Cat
2. Add Dog
3. Remove Cat
4. Remove Dog
0. Quit
Enter your choice: 1
Enter cat name: kavi
Enter coat color: white
Cat added.
```

```
Run Q_06 x
[Icons: Run, Stop, Copy, Paste, More]

↑ Menu:
↓ 1. Add Cat
⇌ 2. Add Dog
⇌ 3. Remove Cat
⇌ 4. Remove Dog
🖨 0. Quit
🗑 Enter your choice: 2
Enter dog name: jacky
Enter weight (positive number): 12
Dog added.

Menu:
1. Add Cat
2. Add Dog
3. Remove Cat
4. Remove Dog
0. Quit
Enter your choice: 2
Enter dog name: astan
Enter weight (positive number): 15
Dog added.
```



```
Run  Q_06 x
[Icons: Run, Stop, Copy, Paste, More]

↑
↓
≡
≡
≡
≡
≡

Menu:
1. Add Cat
2. Add Dog
3. Remove Cat
4. Remove Dog
0. Quit
Enter your choice: 3
Enter cat name to remove: stella
Cat removed.

Menu:
1. Add Cat
2. Add Dog
3. Remove Cat
4. Remove Dog
0. Quit
Enter your choice: 4
Enter dog name to remove: astan
Dog removed.
```

Menu:

1. Add Cat
2. Add Dog
3. Remove Cat
4. Remove Dog
0. Quit

Enter your choice: 0

Final List of Cats:

Name: kavi , Type:Cat , Coat Color: white

Final List of Dogs:

Name: jacky , Type: Dog , Weight: 12.0

Exiting program.

Process finished with exit code 0