**Worksheet 06**

**CTEC 22043**

**Object Oriented Programming**

Q1.

pet.java

|  |
| --- |
| ***package Q\_01;  public class cat extends pet {  @Override  public String speak() {  return "";   } }*** |

Cat.java

|  |
| --- |
| ***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.";  }  }*** |

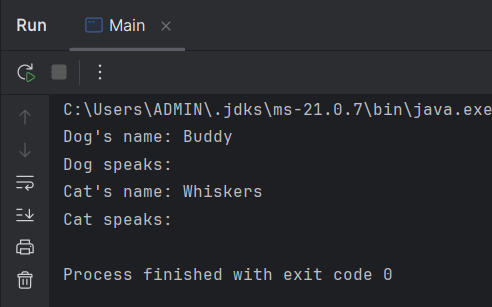
Dog.java

|  |
| --- |
| ***package Q\_01;  public class Dog extends pet {  @Override  public String speak() {  return ""; // Returns an empty string  } }*** |

Main.java

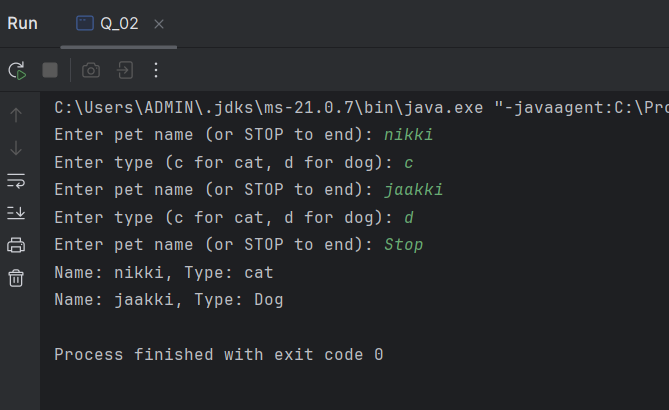
|  |
| --- |
| ***package Q\_01;  public class Main {  public static void main(String[] args) {  // Create a Dog object  Dog myDog = new Dog();  myDog.setName("Buddy");  System.out.println("Dog's name: " + myDog.getName());  System.out.println("Dog speaks: " + myDog.speak());   // Create a Cat object  cat myCat = new cat();  myCat.setName("Whiskers");  System.out.println("Cat's name: " + myCat.getName());  System.out.println("Cat speaks: " + myCat.speak());  }  }*** |

Output



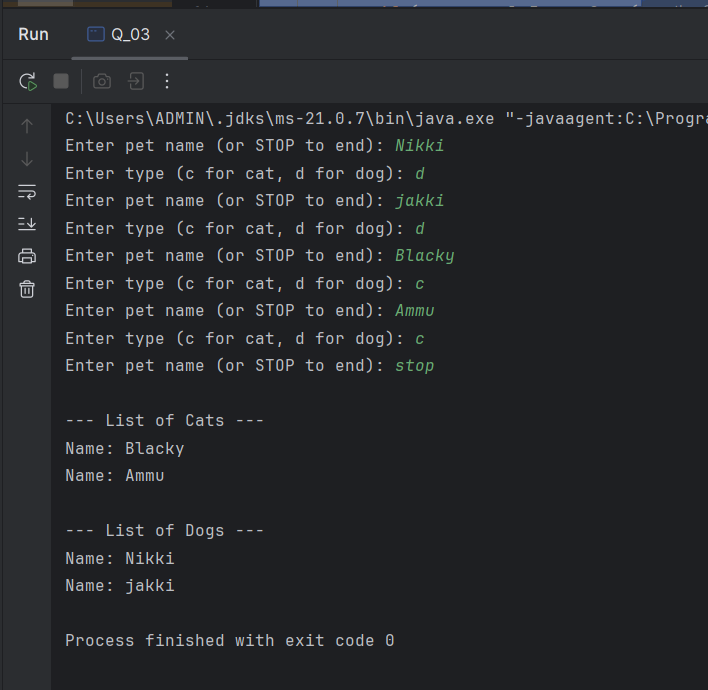
Q2

|  |
| --- |
| ***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 input = new Scanner(System.in);  pet[] pets = new pet[100]; // maximum 100 pets  int count = 0;   while (true) {  System.out.print("Enter pet name (or STOP to end): ");  String name = input.nextLine();  if (name.equalsIgnoreCase("STOP")) break;   System.out.print("Enter type (c for cat, d for dog): ");  String type = input.nextLine();   pet pet;  if (type.equalsIgnoreCase("c")) {  pet = new cat();  } else {  pet = new Dog();  }   pet.setName(name);  pets[count++] = pet;  }   for (int i = 0; i < count; i++) {  System.out.println("Name: " + pets[i].getName() + ", Type: " + pets[i].getClass().getSimpleName());  }  }  }*** |



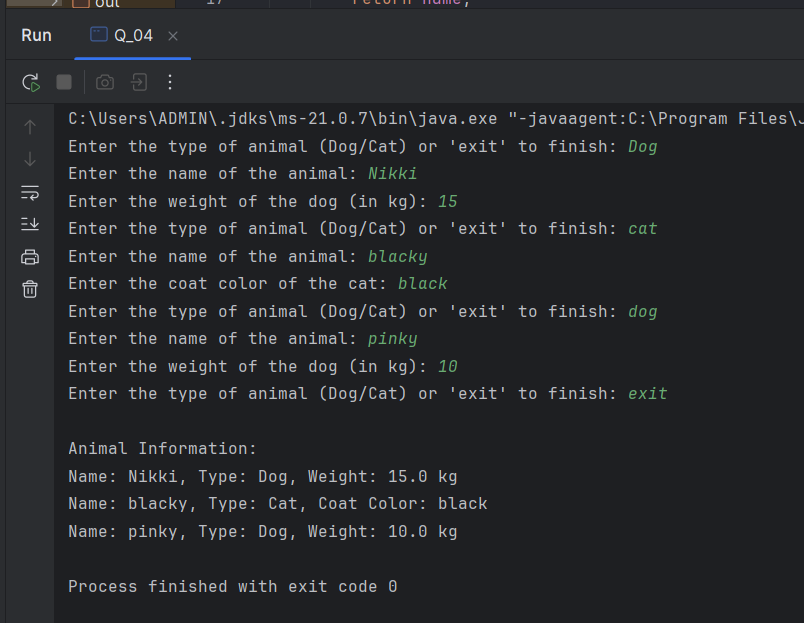
Q3.

|  |
| --- |
| ***package Q\_03; import Q\_01.cat; import Q\_01.Dog; import Q\_01.pet; import java.util.ArrayList; import java.util.Scanner;  public class Q\_03 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  ArrayList<pet> pets = new ArrayList<>();   while (true) {  System.out.print("Enter pet name (or STOP to end): ");  String name = input.nextLine();  if (name.equalsIgnoreCase("STOP")) break;   System.out.print("Enter type (c for cat, d for dog): ");  String type = input.nextLine();   pet pet = type.equalsIgnoreCase("c") ? new cat() : new Dog();  pet.setName(name);  pets.add(pet);  }  System.out.println("\n--- List of Cats ---");  for (pet p : pets) {  if (p instanceof cat) {  System.out.println("Name: " + p.getName());  }  }  System.out.println("\n--- List of Dogs ---");  for (pet p : pets) {  if (p instanceof Dog) {  System.out.println("Name: " + p.getName());  }  }  } }*** |



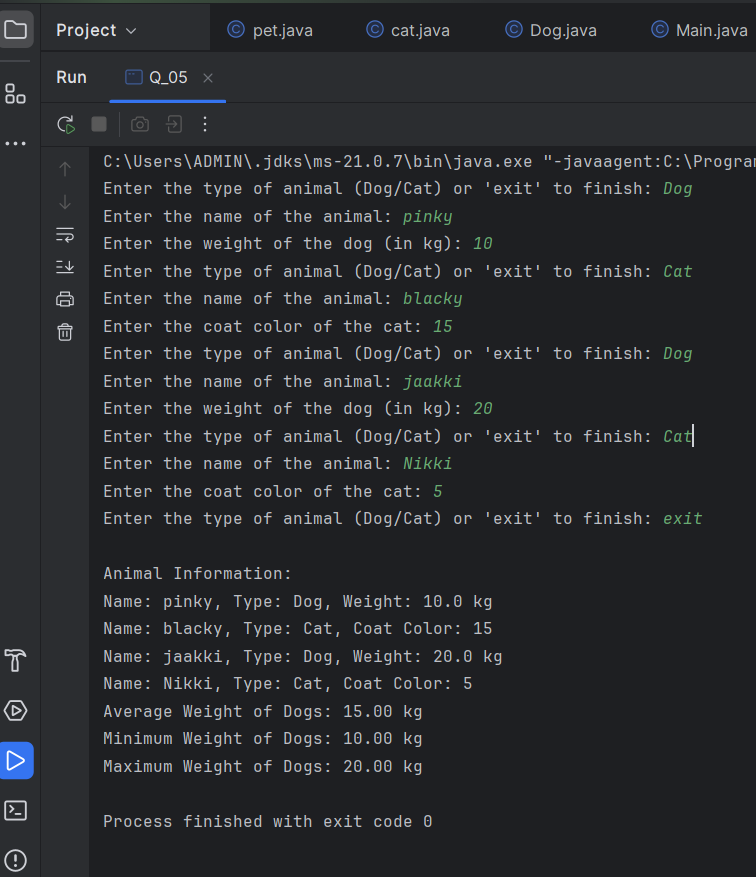
Q4.

|  |
| --- |
| ***Package Q\_04;***  ***import java.util.ArrayList; import java.util.List; import java.util.Scanner;  class Dog {  private String name;  private String type;  private double weight;   public Dog(String name, double weight) {  this.name = name;  this.type = "Dog";  this.weight = weight;  }   public String getName() {  return name;  }   public String getType() {  return type;  }   public double getWeight() {  return weight;  }   public void setWeight(double weight) {  this.weight = weight;  }   @Override  public String toString() {  return "Name: " + name + ", Type: " + type + ", Weight: " + weight + " kg";  } }  class Cat {  private String name;  private String type;  private String coatColor;   public Cat(String name, String coatColor) {  this.name = name;  this.type = "Cat";  this.coatColor = coatColor;  }   public String getName() {  return name;  }   public String getType() {  return type;  }   public String getCoatColor() {  return coatColor;  }   public void setCoatColor(String coatColor) {  this.coatColor = coatColor;  }   @Override  public String toString() {  return "Name: " + name + ", Type: " + type + ", Coat Color: " + coatColor;  } }  public class Q\_04 {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  List<Object> animals = new ArrayList<>();   while (true) {  System.out.print("Enter the type of animal (Dog/Cat) or 'exit' to finish: ");  String animalType = scanner.nextLine().trim();   if (animalType.equalsIgnoreCase("exit")) {  break;  }   System.out.print("Enter the name of the animal: ");  String name = scanner.nextLine().trim();   if (animalType.equalsIgnoreCase("dog")) {  System.out.print("Enter the weight of the dog (in kg): ");  double weight = 0;  while (true) {  try {  weight = Double.parseDouble(scanner.nextLine().trim());  break;  } catch (NumberFormatException e) {  System.out.print("Invalid input. Please enter a valid number for weight: ");  }  }  animals.add(new Dog(name, weight));   } else if (animalType.equalsIgnoreCase("cat")) {  System.out.print("Enter the coat color of the cat: ");  String coatColor = scanner.nextLine().trim();  animals.add(new Cat(name, coatColor));  } else {  System.out.println("Invalid animal type. Please enter 'Dog' or 'Cat'.");  }  }   System.out.println("\nAnimal Information:");  for (Object animal : animals) {  System.out.println(animal.toString());  }   scanner.close();  } }*** |



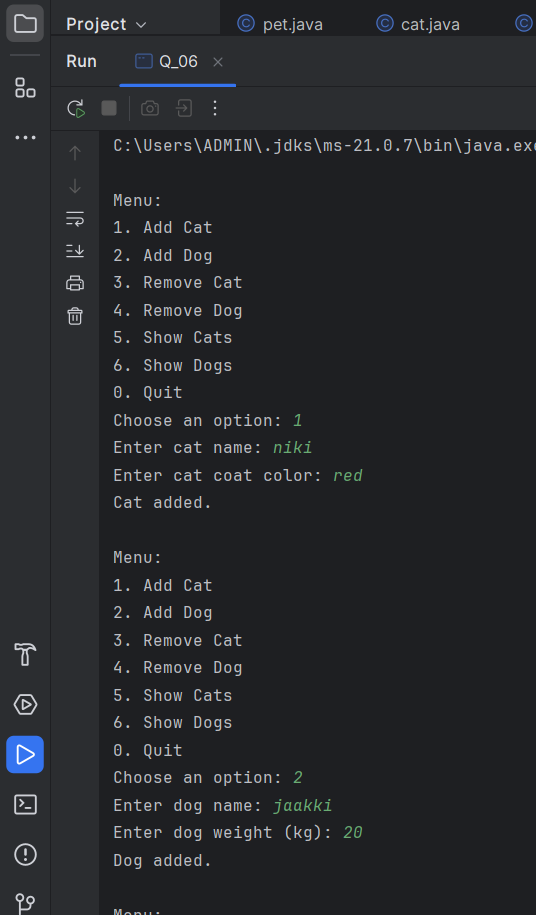
Q5.

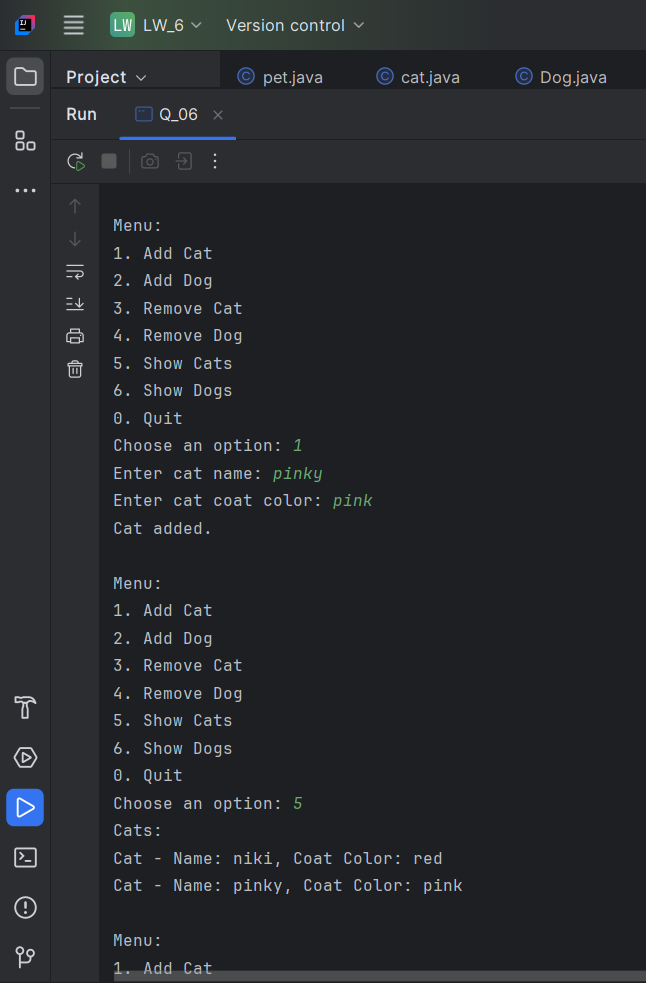
|  |
| --- |
| ***package Q\_05; import java.util.ArrayList; import java.util.List; import java.util.Scanner;  class Dog {  private String name;  private String type;  private double weight;   public Dog(String name, double weight) {  this.name = name;  this.type = "Dog";  this.weight = weight;  }   public String getName() {  return name;  }   public String getType() {  return type;  }   public double getWeight() {  return weight;  }   public void setWeight(double weight) {  this.weight = weight;  }   @Override  public String toString() {  return "Name: " + name + ", Type: " + type + ", Weight: " + weight + " kg";  } }  class Cat {  private String name;  private String type;  private String coatColor;   public Cat(String name, String coatColor) {  this.name = name;  this.type = "Cat";  this.coatColor = coatColor;  }   public String getName() {  return name;  }   public String getType() {  return type;  }   public String getCoatColor() {  return coatColor;  }   public void setCoatColor(String coatColor) {  this.coatColor = coatColor;  }   @Override  public String toString() {  return "Name: " + name + ", Type: " + type + ", Coat Color: " + coatColor;  } }  public class Q\_05 {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  List<Object> animals = new ArrayList<>();   while (true) {  System.out.print("Enter the type of animal (Dog/Cat) or 'exit' to finish: ");  String animalType = scanner.nextLine().trim();   if (animalType.equalsIgnoreCase("exit")) {  break;  }   System.out.print("Enter the name of the animal: ");  String name = scanner.nextLine().trim();   if (animalType.equalsIgnoreCase("dog")) {  System.out.print("Enter the weight of the dog (in kg): ");  double weight = 0;  while (true) {  try {  weight = Double.parseDouble(scanner.nextLine().trim());  break;  } catch (NumberFormatException e) {  System.out.print("Invalid input. Please enter a valid number for weight: ");  }  }  animals.add(new Dog(name, weight));   } else if (animalType.equalsIgnoreCase("cat")) {  System.out.print("Enter the coat color of the cat: ");  String coatColor = scanner.nextLine().trim();  animals.add(new Cat(name, coatColor));  } else {  System.out.println("Invalid animal type. Please enter 'Dog' or 'Cat'.");  }  }   System.out.println("\nAnimal Information:");  for (Object animal : animals) {  System.out.println(animal.toString());  }   // Create an array to hold only Dog objects  Dog[] dogArray = new Dog[animals.size()];  int dogCount = 0;   // Populate the dogArray with Dog objects  for (Object animal : animals) {  if (animal instanceof Dog) {  dogArray[dogCount++] = (Dog) animal;  }  }   // Calculate average, minimum, and maximum weights  if (dogCount > 0) {  double totalWeight = 0;  double minWeight = Double.MAX\_VALUE;  double maxWeight = Double.MIN\_VALUE;   for (int i = 0; i < dogCount; i++) {  double weight = dogArray[i].getWeight();  totalWeight += weight;  if (weight < minWeight) {  minWeight = weight;  }  if (weight > maxWeight) {  maxWeight = weight;  }  }   double averageWeight = totalWeight / dogCount;   System.out.printf("Average Weight of Dogs: %.2f kg%n", averageWeight);  System.out.printf("Minimum Weight of Dogs: %.2f kg%n", minWeight);  System.out.printf("Maximum Weight of Dogs: %.2f kg%n", maxWeight);  } else {  System.out.println("No dogs were entered.");  }   scanner.close();  } }*** |



Q6

|  |
| --- |
| ***package Q\_06; import java.util.ArrayList; import java.util.Scanner;  class Dog {  private String name;  private double weight;   public Dog(String name, double weight) {  this.name = name;  this.weight = weight;  }   public String getName() {  return name;  }   public double getWeight() {  return weight;  }   public String toString() {  return "Dog - Name: " + name + ", Weight: " + weight + " kg";  } }  class Cat {  private String name;  private String coatColor;   public Cat(String name, String coatColor) {  this.name = name;  this.coatColor = coatColor;  }   public String getName() {  return name;  }   public String getCoatColor() {  return coatColor;  }   public String toString() {  return "Cat - Name: " + name + ", Coat Color: " + coatColor;  } }  public class Q\_06 {  public static void main(String[] args) {  ArrayList<Dog> dogs = new ArrayList<>();  ArrayList<Cat> cats = new ArrayList<>();  Scanner scanner = new Scanner(System.in);  int choice;   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("5. Show Cats");  System.out.println("6. Show Dogs");  System.out.println("0. Quit");  System.out.print("Choose an option: ");  choice = scanner.nextInt();  scanner.nextLine();    if (choice == 0) {  System.out.println("Goodbye!");  break;  }   switch (choice) {  case 1:  System.out.print("Enter cat name: ");  String catName = scanner.nextLine();  System.out.print("Enter cat coat color: ");  String coatColor = scanner.nextLine();  cats.add(new Cat(catName, coatColor));  System.out.println("Cat added.");  break;  case 2:  System.out.print("Enter dog name: ");  String dogName = scanner.nextLine();  System.out.print("Enter dog weight (kg): ");  double weight = scanner.nextDouble();  scanner.nextLine();   dogs.add(new Dog(dogName, weight));  System.out.println("Dog added.");  break;  case 3:  System.out.print("Enter cat name to remove: ");  String removeCatName = scanner.nextLine();  boolean catRemoved = false;  for (int i = 0; i < cats.size(); i++) {  if (cats.get(i).getName().equalsIgnoreCase(removeCatName)) {  cats.remove(i);  catRemoved = true;  System.out.println("Cat removed.");  break;  }  }  if (!catRemoved) {  System.out.println("Cat not found.");  }  break;  case 4:  System.out.print("Enter dog name to remove: ");  String removeDogName = scanner.nextLine();  boolean dogRemoved = false;  for (int i = 0; i < dogs.size(); i++) {  if (dogs.get(i).getName().equalsIgnoreCase(removeDogName)) {  dogs.remove(i);  dogRemoved = true;  System.out.println("Dog removed.");  break;  }  }  if (!dogRemoved) {  System.out.println("Dog not found.");  }  break;  case 5:  if (cats.isEmpty()) {  System.out.println("No cats in the list.");  } else {  System.out.println("Cats:");  for (Cat c : cats) {  System.out.println(c);  }  }  break;  case 6:  if (dogs.isEmpty()) {  System.out.println("No dogs in the list.");  } else {  System.out.println("Dogs:");  for (Dog d : dogs) {  System.out.println(d);  }  }  break;  default:  System.out.println("Invalid option.");  }  }   scanner.close();***  ***} }*** |





A screenshot of a computer program

AI-generated content may be incorrect.