

Create a Pet class that keeps track of the name, age, weight, type of animal, and breed for records at an animal clinic with a constructor, a toString method, and getters and setters for each instance variable.

Save & Run7/17/2024, 1:58:37 PM - 2 of 2DownloadShow CodeLensReformatPair?

```
1/**
2 * Pet class (complete comments)
3 *
4 * Akshat Gang
5 * 7/17/24
6 */
7class Pet
8{
9 // keep track of the name, age, weight, type of animal, and breed of the pet
10 private String name;
11 private int age;
12 private double weight;
13 private String type;
14 private String breed;
15 // Write a constructor, accessor (get) methods, and a toString method. Use good
16 // commenting.
17
18 //creates default pet
19 public Pet(){
20     name = "Fido";
21     age = 5;
22     weight = 10.0;
23     type = "dog";
24     breed = "terrier";
25 }
26 //user entered info pet constructor
27 public Pet(String initName, int initAge, double initWeight,
28     String initType, String initBreed){
29     name = initName;
30     age = initAge;
31     weight = initWeight;
32     type = initType;
33     breed = initBreed;
34 }
35 //returns attributes of pet
36 public String getName(){
37     return name;
38 }
39 public int getAge(){
40     return age;
41 }
42 public double getWeight(){
```

Create a Pet class that keeps track of the name, age, weight, type of animal, and breed for records at an animal clinic with a constructor, a toString method, and getters and setters for each instance variable.

Save & Run7/17/2024, 1:59:38 PM - 3 of 3DownloadShow CodeLensReformatPair?

```
42 public double getWeight(){
43     return weight;
44 }
45 public String getType(){
46     return type;
47 }
48 public String getBreed(){
49     return breed;
50 }
51 //sets attributes of pet
52 public String setName(String initName){
53     return name;
54 }
55 public int setAge(int initAge){
56     return age;
57 }
58 public double setWeight(double initWeight){
59     return weight;
60 }
61 public String setType(String initType){
62     return type;
63 }
64 public String setBreed(String initBreed){
65     return breed;
66 }
67 //returns all data in string format
68 public String toString(){
69     return name + ", age " + age + ", weighs " + weight + " pounds and is a "
70         + breed + " " + type;
71 }
72 // Don't forget to complete the main method in the TesterClass below!
73
74
75 public class TesterClass
76 {
77     // main method for testing
78     public static void main(String[] args)
79     {
80         // Create 2 Pet objects and test all your methods
81         Pet doggy = new Pet();
82         System.out.println(doggy.toString());
83         Pet kitty = new Pet("Socks", 3, 15, "cat", "calico");
84         System.out.println(kitty.toString());
85         System.out.println("Your " + kitty.getType() + " weighs "
86             + kitty.getWeight());
87         kitty.setAge(4);
88         kitty.setWeight(18);
89     }
90 }
91
92
93
```

Create a Pet class that keeps track of the name, age, weight, type of animal, and breed for records at an animal clinic with a constructor, a toString method, and getters and setters for each instance variable.

Save & Run7/17/2024, 1:59:38 PM - 3 of 3DownloadShow CodeLensReformatPair?

```
52 //sets attributes of pet
53 public String setName(String initName){
54     return name;
55 }
56 public int setAge(int initAge){
57     return age;
58 }
59 public double setWeight(double initWeight){
60     return weight;
61 }
62 public String setType(String initType){
63     return type;
64 }
65 public String setBreed(String initBreed){
66     return breed;
67 }
68 //returns all data in string format
69 public String toString(){
70     return name + ", age " + age + ", weighs " + weight + " pounds and is a "
71         + breed + " " + type;
72 }
73 // Don't forget to complete the main method in the TesterClass below!
74
75
76 public class TesterClass
77 {
78     // main method for testing
79     public static void main(String[] args)
80     {
81         // Create 2 Pet objects and test all your methods
82         Pet doggy = new Pet();
83         System.out.println(doggy.toString());
84         Pet kitty = new Pet("Socks", 3, 15, "cat", "calico");
85         System.out.println(kitty.toString());
86         System.out.println("Your " + kitty.getType() + " weighs "
87             + kitty.getWeight());
88         kitty.setAge(4);
89         kitty.setWeight(18);
90     }
91 }
92
93
```

Create method(s) with parameters to print out verses of the song The Ants Go Marching.

<https://youtu.be/QPwEZ8Vv2YQ/The+Ants+Go+Marching>

Save & Run

7/17/2024, 2:11:52 PM - 3 of 3

Download

Show CodeLens

Reformat

Pair?

```

1 //Code by Akshat Garg
2 public class Song
3 {
4     // Create at least 1 method called verse that takes 2 parameters
5     // that can be used to print out the verses of the song The Ants Go Marching
6     public void verse(String line1, String num){
7         System.out.println("The ants go marching " + num + " by " + num + ", hurrah, hurrah");
8         System.out.println("The ants go marching " + num + " by " + num + ", hurrah, hurrah");
9         System.out.println("The ants go marching " + num + " by " + num + ", hurrah, hurrah");
10        System.out.println("The little one stops to " + line1);
11        System.out.println("And they all go marching down to the ground");
12        System.out.println("To get out of the rain, BOOM! BOOM! BOOM! BOOM!");
13    }
14    public static void main(String args[])
15    {
16        Song theAntsGoMarching = new Song();
17        theAntsGoMarching.verse("suck a thumb","one");
18        theAntsGoMarching.verse("tie a shoe","two");
19        theAntsGoMarching.verse("climb a tree","three");
20        // Create a Song object and call its method(s) to print out
21        // the verses of The Ants Go Marching
22        // There should be atleast 1 method called verse that takes 2 arguments.
23    }
24 }
25 }
26 }
27 }

```

The ants go marching one by one, hurrah, hurrah
The ants go marching one by one, hurrah, hurrah
The ants go marching one by one, hurrah, hurrah
The little one stops to suck a thumb
And they all go marching down to the ground
To get out of the rain, BOOM! BOOM! BOOM! BOOM!
The ants go marching two by two, hurrah, hurrah
The ants go marching two by two, hurrah, hurrah
The ants go marching two by two, hurrah, hurrah
The little one stops to tie a shoe
And they all go marching down to the ground
To get out of the rain, BOOM! BOOM! BOOM! BOOM!
The ants go marching three by three, hurrah, hurrah
The ants go marching three by three, hurrah, hurrah
The ants go marching three by three, hurrah, hurrah
The little one stops to climb a tree
And they all go marching down to the ground
To get out of the rain, BOOM! BOOM! BOOM! BOOM!

Result	Expected	Actual	Notes
Pass	true	true	Checking that code contains verse(...) method header with two String parameters
Pass	true	true	Checking that code contains a new Song object
Pass	3	3	Checking that code contains three calls to verse(...) method using object.method(...) syntax
Pass	No errors	No errors	Checking output from main

You got 4 out of 4 correct. 100.00%

Activity: 5.6.2.2 ActiveCode (challenge-5-6-song)

Song activity was really fun to learn how to use repetitive methods, using string manipulation to print out a repetitive song is really interesting.