

Pet.java

```
1 /*
2  * Written by JJ Shepherd
3  */
4 //Step 1. Define the class
5 public class Pet {
6     //Step 2. Create the properties
7     //Instance Variables
8     private String name;
9     private int age;
10    private double weight;
11    private PetType type;
12    //Step 3. Create Constructors
13    public Pet()//Default
14    {
15        this.name = "none";
16        this.age = 0;
17        this.weight = 1.0;
18        this.type = PetType.UNKNOWN;
19    }
20    public Pet(String aN, int anA, double aW, PetType aT)
21    {
22        this.setName(aN);
23        this.setAge(anA);
24        this.setWeight(aW);
25        this.setType(aT);
26    }
27    //Step 4. Create Accessors
28    public String getName()
29    {
30        return this.name;
31    }
32    public int getAge()
33    {
34        return this.age;
35    }
36    public double getWeight()
37    {
38        return this.weight;
39    }
40    public PetType getType()
41    {
42        return this.type;
43    }
44    //Step 5. Create Mutators
45    public void setName(String aN)
46    {
47        if(aN != null)
48            this.name = aN;
49        else
50            this.name = "none";
51    }
52    public void setAge(int anA)
53    {
54        if(anA >= 0)
55            this.age = anA;
56        else
57            this.age = 0;
```

```

58     }
59     public void setWeight(double aW)
60     {
61         if(aW > 0.0)
62             this.weight = aW;
63         else
64             this.weight = 1.0;
65     }
66     public void setType(PetType aT)
67     {
68         if(aT != null)
69             this.type = aT;
70         else
71             this.type = PetType.UNKNOWN;
72     }
73     //Step 6. Other Methods
74     public String toString()
75     {
76         return this.name+" "+this.age+" "+this.weight+" "+this.type;
77     }
78     public boolean equals(Pet aP)
79     {
80         return aP != null &&
81             this.name.equals(aP.getName()) &&
82             this.age == aP.getAge() &&
83             this.weight == aP.getWeight() &&
84             this.type == aP.getType();
85     }
86     public void giveComplement()
87     {
88         System.out.println(this.name+" reacted with joy!");
89     }
90     public void giveComplement(int c)
91     {
92         for(int i=0;i<c;i++)
93             this.giveComplement();
94     }
95     //Static methods
96     public static Pet createNewPet(PetType aT)
97     {
98         Pet newPet = new Pet();
99         newPet.setType(aT);
100         return newPet;
101     }
102     public static Pet clonePet(Pet aP)
103     {
104         if(aP == null)
105             return null;
106         return new Pet(aP.getName(),aP.getAge(),aP.getWeight(),aP.getType());
107     }
108 }
109

```