

Object Interaction in Java – Class Hierarchies

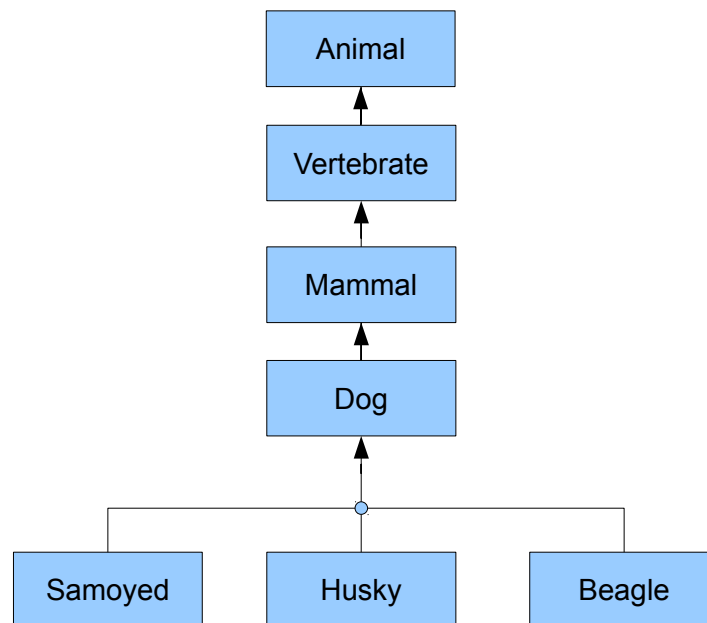
Consider the following: a dictionary might describe the *Samoyed* as a dog of medium size with a thick white coat, erect ears, and a tail which curls over the back. Even if you had never heard of such a dog breed before, these few descriptive phrases would give you a fairly clear picture of the Samoyed.

The key to such a clear picture is all of the additional information that is carried along with the word *dog*. Without knowing that the Samoyed is a type of dog, you might have a very difficult time in trying to picture it. The additional information provided helps to distinguish the Samoyed from other dogs, but all dogs share certain characteristics that we can use as a starting point for this particular breed. We say that the Samoyed **inherits** characteristics from dogs, in general.

In computer programming, and Java in particular, we would consider the `Dog` and `Samoyed` as classes. From our analogy, the `Dog` class is the **superclass** of `Samoyed`. Considering the relationship in the opposite direction, we say that the class `Samoyed` is a **subclass** of `Dog`, or that `Samoyed` **extends** `Dog`.

A class such as `Dog` may have many subclasses, such as `Husky` or `Beagle`. Conversely, any class may have only a single superclass. It is possible, however, to form a chain of superclasses, so a single class may actually inherit from many classes.

Example 1 – Consider the following class-hierarchy diagram which includes the dog breeds we have been discussing, and adds several layers of superclasses.



Each class in the chain inherits from all of the classes above it. In this case, the `Samoyed` inherits from `Dog` (as previously discussed), but also from `Mammal`, `Vertebrate`, and `Animal`. In this example, the top class is `Animal`, which doesn't have a superclass. All animals (obviously) inherit from this class.

In Java, the class with this property is the `Object` class. All other classes inherit from `Object`, although most do so without actually referring to `Object`.

Object Interaction in Java – Class Hierarchies

To indicate in Java that a class extends another class, we use the keyword `extends` in the class header. If `Samoyed` and `Dog` were actual Java classes, the header of `Samoyed` would be

```
class Samoyed extends Dog
```

If no class is specified using the `extends` clause, `Object` is the default superclass. Thus the following headers are equivalent:

```
class Fraction
class Fraction extends Object
```

The classes from which a class inherits are sometimes called the *ancestors*. The classes which inherit from a class are sometimes called the *descendants* (like a family tree).

The concept of **inheritance** is fundamental to object-oriented programming. When a class inherits from another class, it incorporates all of the fields and methods of the superclass into its own class definition.

Example 2 – Building a `Student` class.

Suppose we begin by defining a `Person` class

```
class Person
```

or

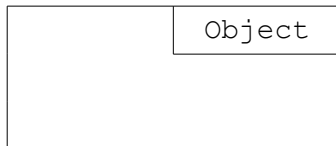
```
class Person extends Object
```

From here, we could define a `Student` class as an extension of the `Person` class.

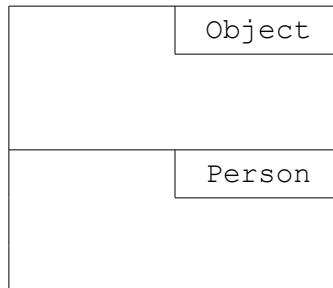
```
class Student extends Person
```

A more convenient form of diagram to represent the relationship between classes is as follows.

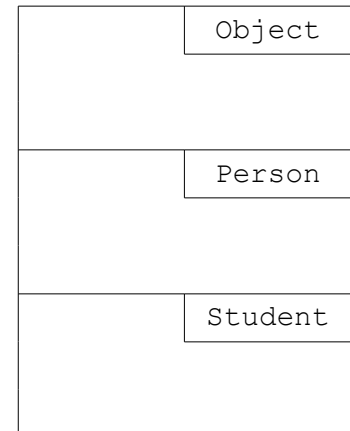
An Object



A Person



A Student



Notice how the diagrams reflect the inheritance relations. Any `Person` object has both an `Object` part and a `Person` part. Similarly, any `Student` object has an `Object` part, a `Person` part, and a `Student` part.

Object Interaction in Java – Class Hierarchies

Exercises

1.
 - a) In the class hierarchy diagram shown in Example 1, what is the superclass of `Mammal`?
 - b) What class or classes does `Dog` extend?
 - c) From what class or classes does `Husky` inherit?
 - d) Extend the diagram to include `Invertebrate`, `Human`, and `Insect`.
 - e) Suggest names of two other subclasses of `Vertebrate`.

2. If you were to write a definition of each of the following, what class would you extend? (Do *not* actually write the definitions)
 - a) Carrot
 - b) Apartment
 - c) Flute
 - d) Neutrino

3. Does every Java class have a superclass? Explain.

4. Explain the difference between the statements "`ClassA extends ClassB`" and "`ClassA inherits from ClassB`".

5. Draw a diagram, like the one in Example 1, showing the hierarchical relationship between the following classes: `Rectangle`, `Polygon`, `Quadrilateral`, `Square`, `Object`, `Triangle`, and `Shape`.

Draw a diagram, like the one in Example 2, to show the structure of the `Quadrilateral` object.

Object Interaction in Java – Class Hierarchies

Solutions

1. a) In the class hierarchy diagram shown in Example 1, what is the superclass of `Mammal`?
Vertebrate. Although a class can have an unlimited number of subclasses, it may only have a single superclass, which is the one directly above it in the hierarchy.
 - b) What class or classes does `Dog` extend?
Dog extends Mammal.
 - c) From what class or classes does `Husky` inherit?
Husky inherits properties from Dog, Mammal, Vertebrate, and Animal, even though it only extends Dog.
 - d) Extend the diagram to include `Invertebrate`, `Human`, and `Insect`.
 - e) Suggest names of two other subclasses of `Vertebrate`.
-
2. If you were to write a definition of each of the following, what class would you extend? (Do *not* actually write the definitions)
 - a) Carrot
 - b) Apartment
 - c) Flute
 - d) Neutrino
-
3. Does every Java class have a superclass? Explain.
Yes. At the very least, every class has the object superclass by default.
-
4. Explain the difference between the statements "`ClassA extends ClassB`" and "`ClassA inherits from ClassB`".
-
5. Draw a diagram, like the one in Example 1, showing the hierarchical relationship between the following classes: `Rectangle`, `Polygon`, `Quadrilateral`, `Square`, `Object`, `Triangle`, and `Shape`.

Draw a diagram, like the one in Example 2, to show the structure of the `Quadrilateral` object.