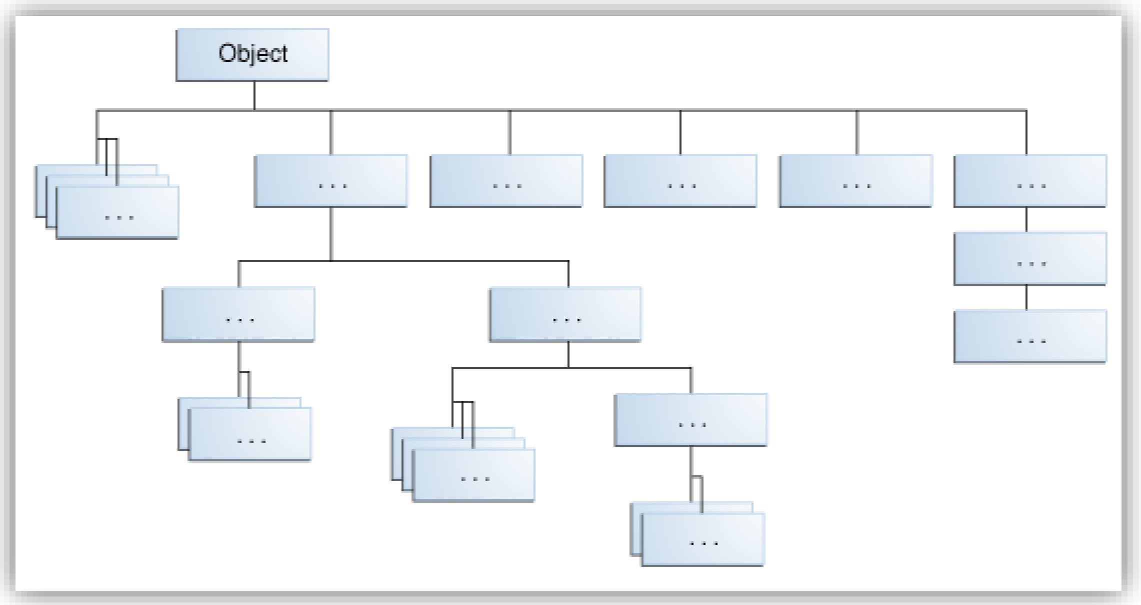
Inheritance Using Java

Inheritance is the OOP ability that allows Java classes to be derived from other classes. The parent class is called a superclass and the derivatives are called subclasses. Subclasses inherit fields and methods from their superclasses.

The “mother of all classes” in Java is the Object class. Each and every class in Java inherits from Object. At the top of the hierarchy, Object is the most general of all classes. Classes near the bottom of the hierarchy provide more specialized behavior.

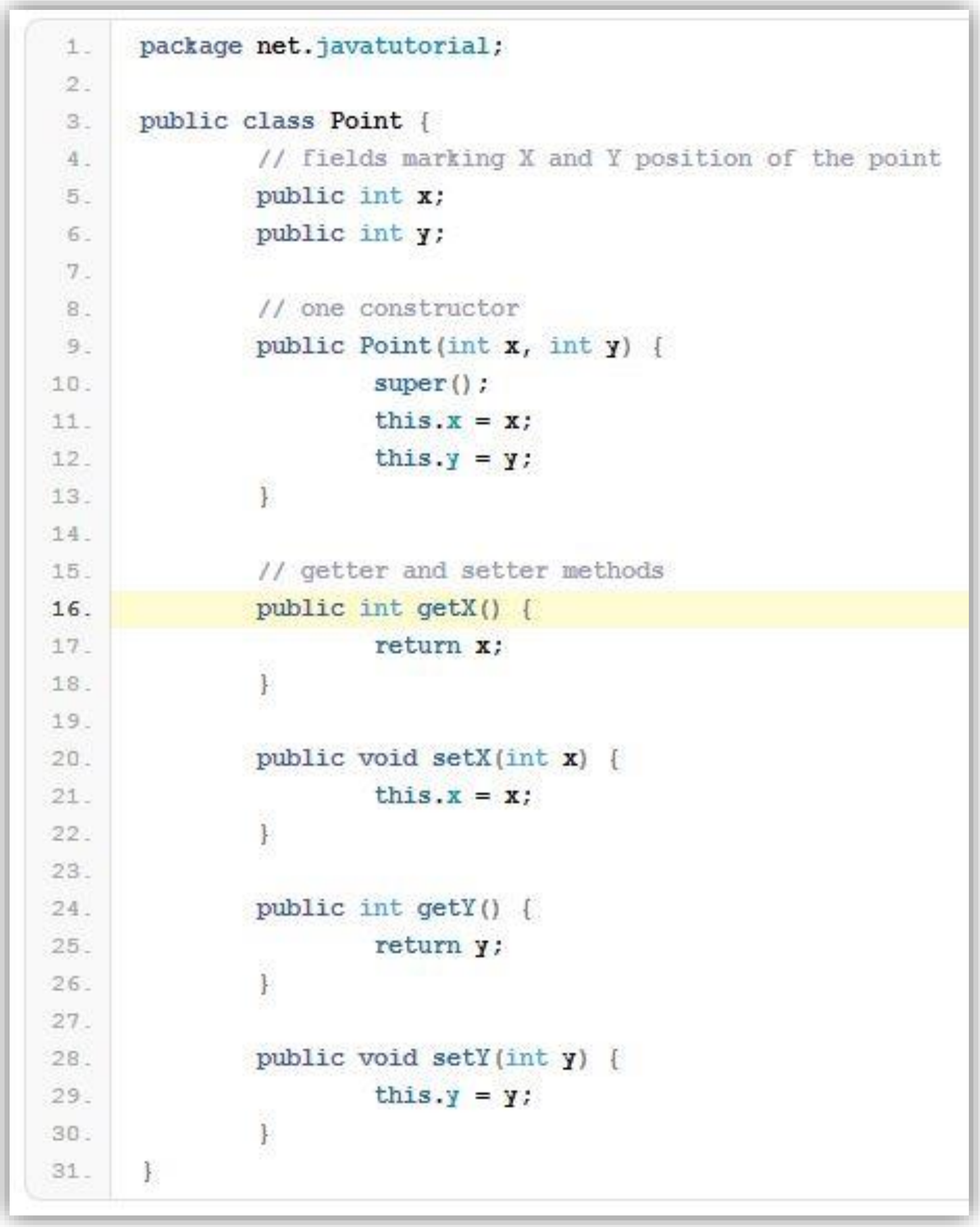


Java has a single inheritance model, which means every class has one and only one direct superclass.

A subclass inherits all of the *public* and *protected* members of its parent, no matter what package the subclass is in. If the subclass is in the same package as its parent, it also inherits the *package-private* members of the parent.

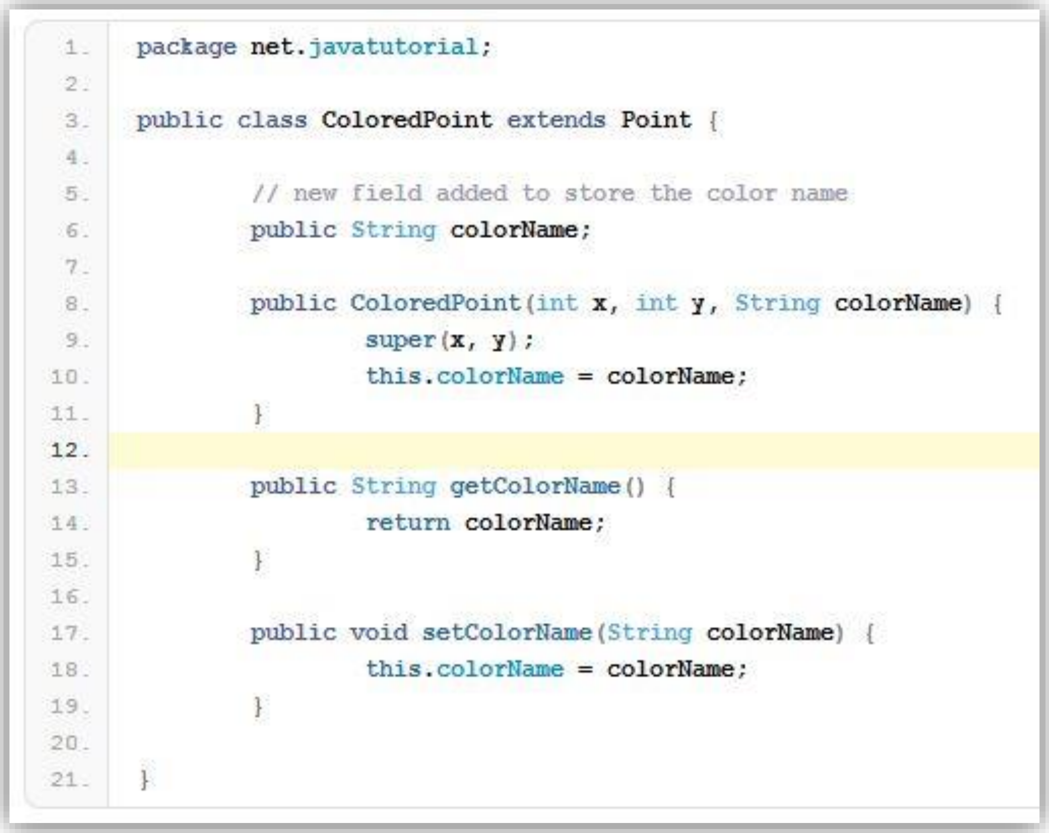
Example:

In the example below we create 3 classes. The superclass Point represents a point with x and y coordinates in the 2-dimensional space.



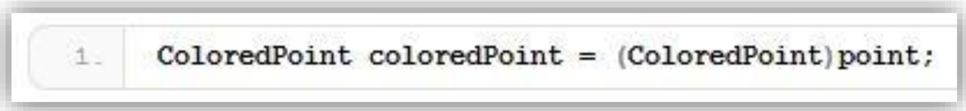
```java
package net.javatutorial;

public class Point {
        // fields marking X and Y position of the point
        public int x;
        public int y;

        // one constructor
        public Point(int x, int y) {
                super();
                this.x = x;
                this.y = y;
        }

        // getter and setter methods
        public int getX() {
                return x;
        }

        public void setX(int x) {
                this.x = x;
        }

        public int getY() {
                return y;
        }

        public void setY(int y) {
                this.y = y;
        }
}
```

ColoredPoint is a subclass which extends all the properties and methods of Point and adds one additional field – colorName. Note how this is done – we use the keyword **extends** to tell which class we want to derive from



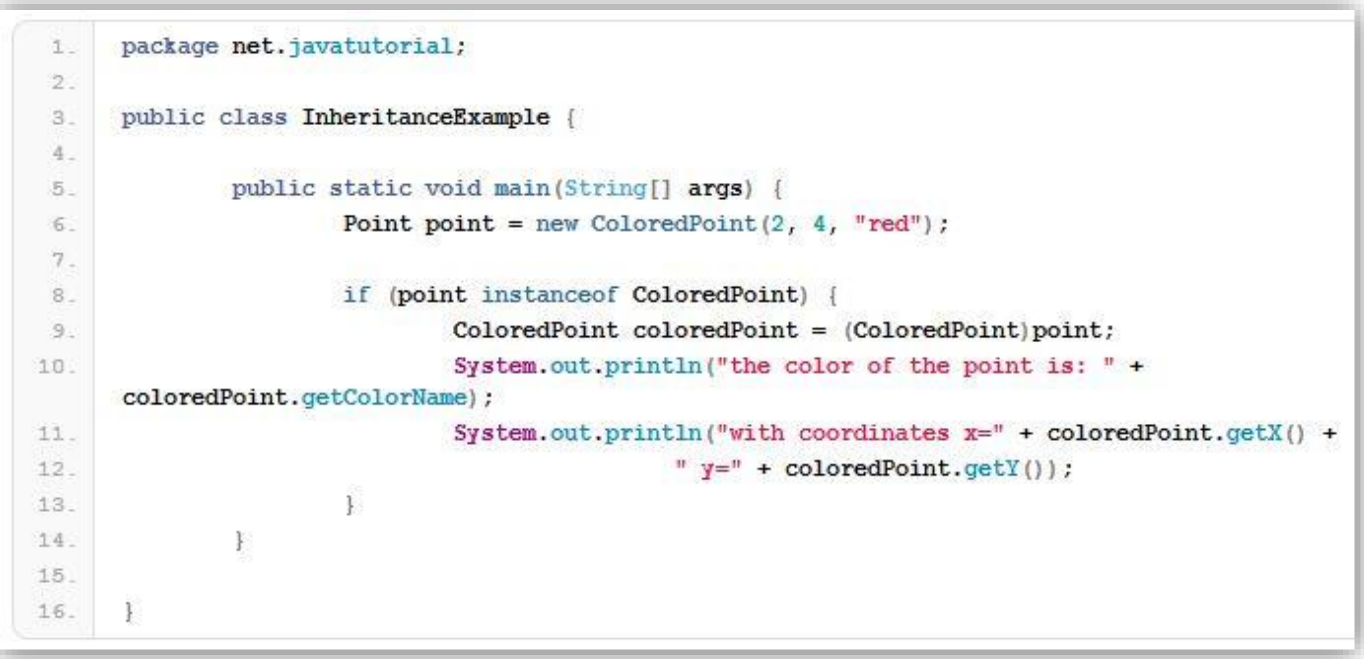
```java
package net.javatutorial;

public class ColoredPoint extends Point {

        // new field added to store the color name
        public String colorName;

        public ColoredPoint(int x, int y, String colorName) {
                super(x, y);
                this.colorName = colorName;
        }

        public String getColorName() {
                return colorName;
        }

        public void setColorName(String colorName) {
                this.colorName = colorName;
        }

}
```

And finally a program to test the inheritance. First we create a new Point of type ColoredPoint. Note the usage of **instanceof** keyword. This way we can check if an object is of certain type. Once we have determined point is of type ColoredPoint we can explicitly type-cast by using:



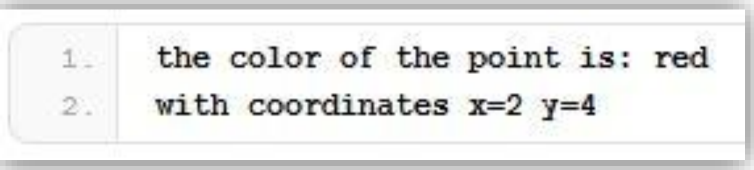
```java
ColoredPoint coloredPoint = (ColoredPoint)point;
```

now we can access the new property *colorName*



```java
package net.javatutorial;

public class InheritanceExample {

        public static void main(String[] args) {
                Point point = new ColoredPoint(2, 4, "red");

                if (point instanceof ColoredPoint) {
                        ColoredPoint coloredPoint = (ColoredPoint)point;
                        System.out.println("the color of the point is: " +
coloredPoint.getColorName);
                        System.out.println("with coordinates x=" + coloredPoint.getX() +
                                        " y=" + coloredPoint.getY());
                }
        }

}
```

Running the example above will produce following output



```
the color of the point is: red
with coordinates x=2 y=4
```

Reference: https://javatutorial.net/java-inheritance-example

……