Introduction to Java Programming

Lecture 3 of 8
April 2, 2019
Emerson Family School

Maxwell Young

Objects, Attributes, and Methods

Allows you to break code into manageable pieces
- referred to as "***modularization***"

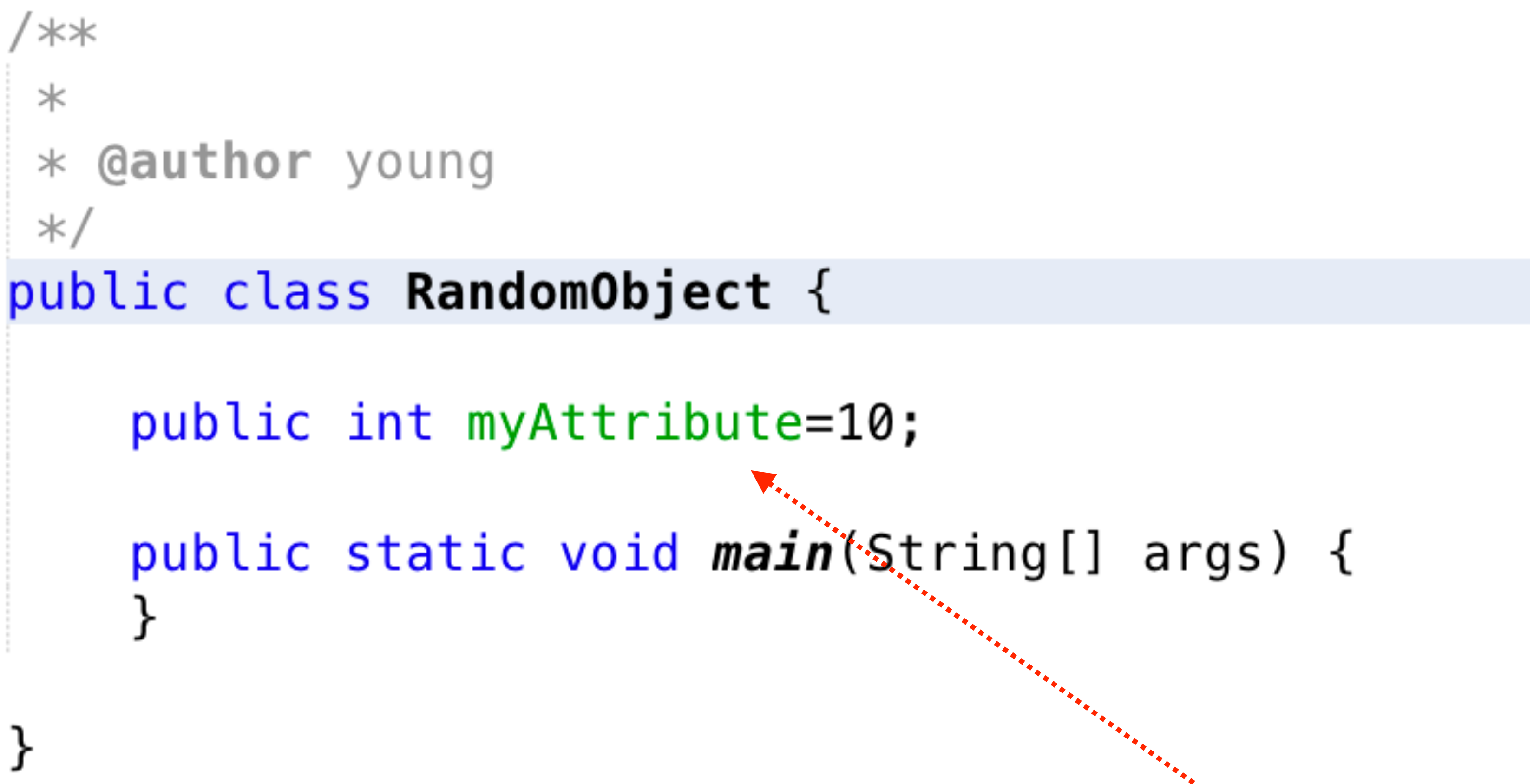
A class defines an ***object***

Objects can have attributes and methods associated with them

Let's see examples of attributes and methods (or functions)

Objects, Attributes, and Methods

Declare attributes at top of the class section



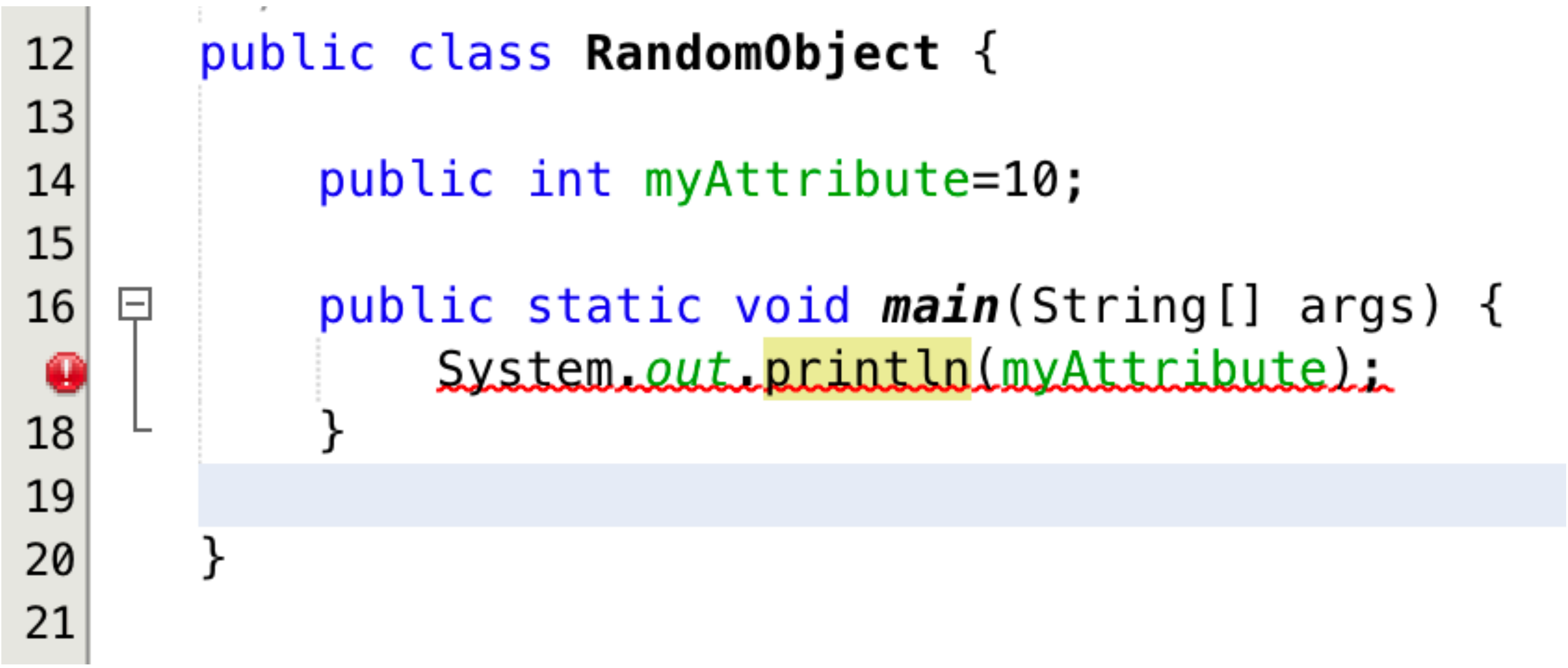
```
/**
 *
 * @author young
 */
public class RandomObject {

    public int myAttribute=10;

    public static void main(String[] args) {
    }

}
```

attribute of type int

`public` keyword means can be accessed by other classes

Can you print this attribute to the screen?

Objects, Attributes, and Methods



```
public class RandomObject {

    public int myAttribute=10;

    public static void main(String[] args) {
        System.out.println(myAttribute);
    }

}
```

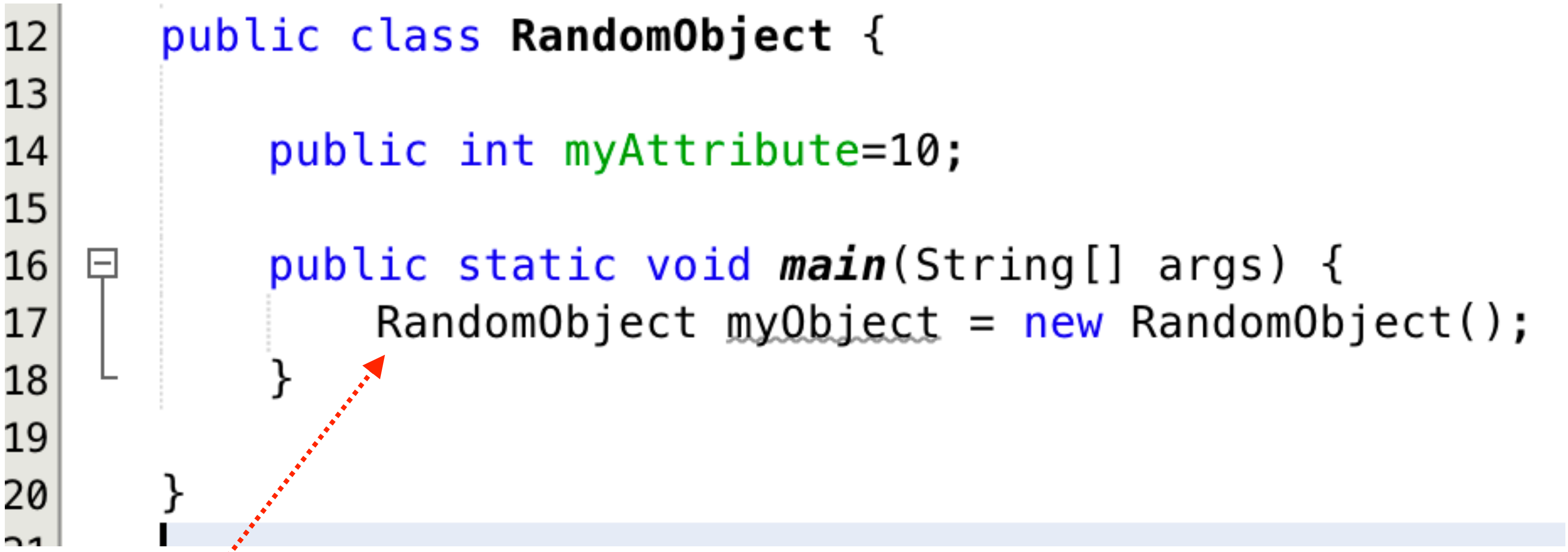
An ***attribute*** belongs to an ***object***

What type is our object here? Type RandomObject

We must ***construct*** an object of type RandomObject and then access this attribute

Objects, Attributes, and Methods

Each class has a special method called a ***constructor***
- you do not have to define this, Java does it for you

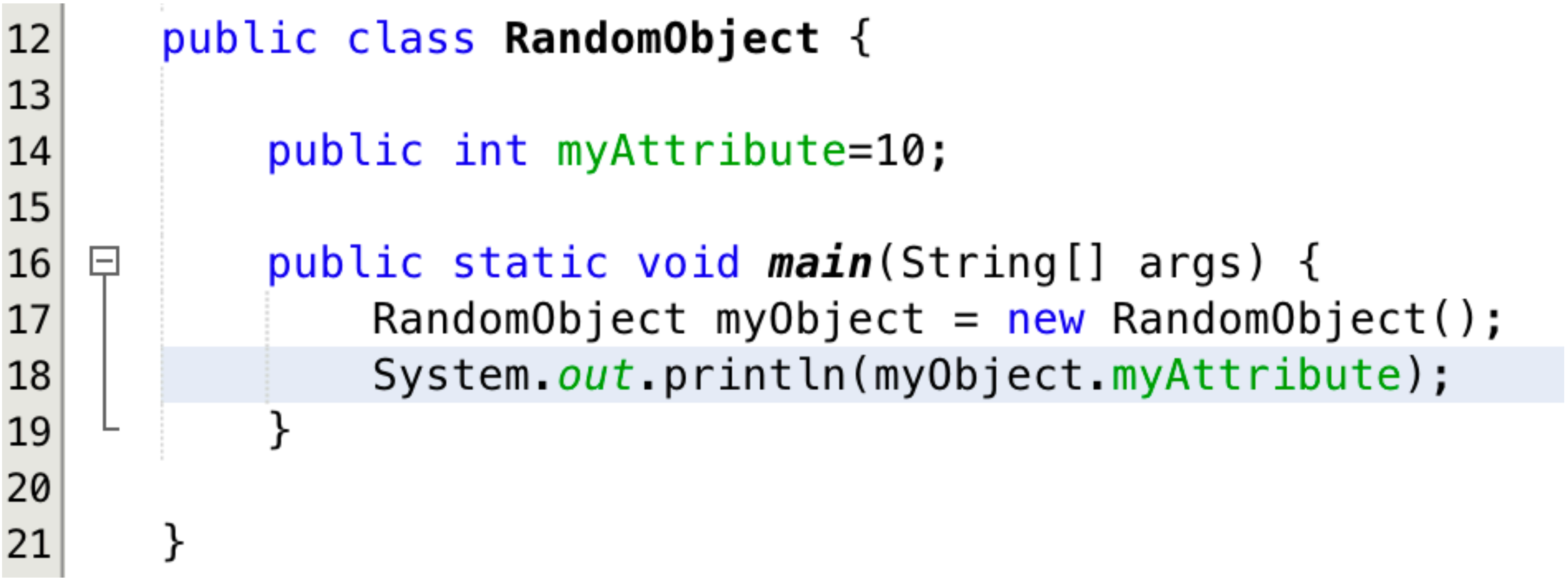


```
public class RandomObject {

    public int myAttribute=10;

    public static void main(String[] args) {
        RandomObject myObject = new RandomObject();
    }

}
```

This is how you call the ***default constructor***

Objects, Attributes, and Methods

Now we can access the attributes of myObject:

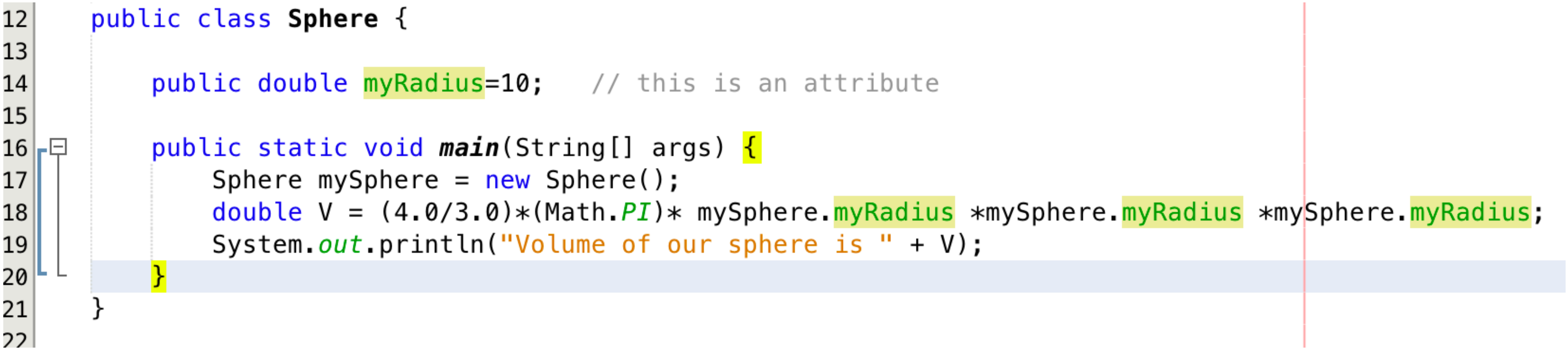


```java
public class RandomObject {

    public int myAttribute=10;

    public static void main(String[] args) {
        RandomObject myObject = new RandomObject();
        System.out.println(myObject.myAttribute);
    }

}
```

Practice by creating a Sphere class (you can reuse your old one) and define an attribute for radius called `myRadius`

Can you print the volume of the sphere to the screen?

Objects, Attributes, and Methods

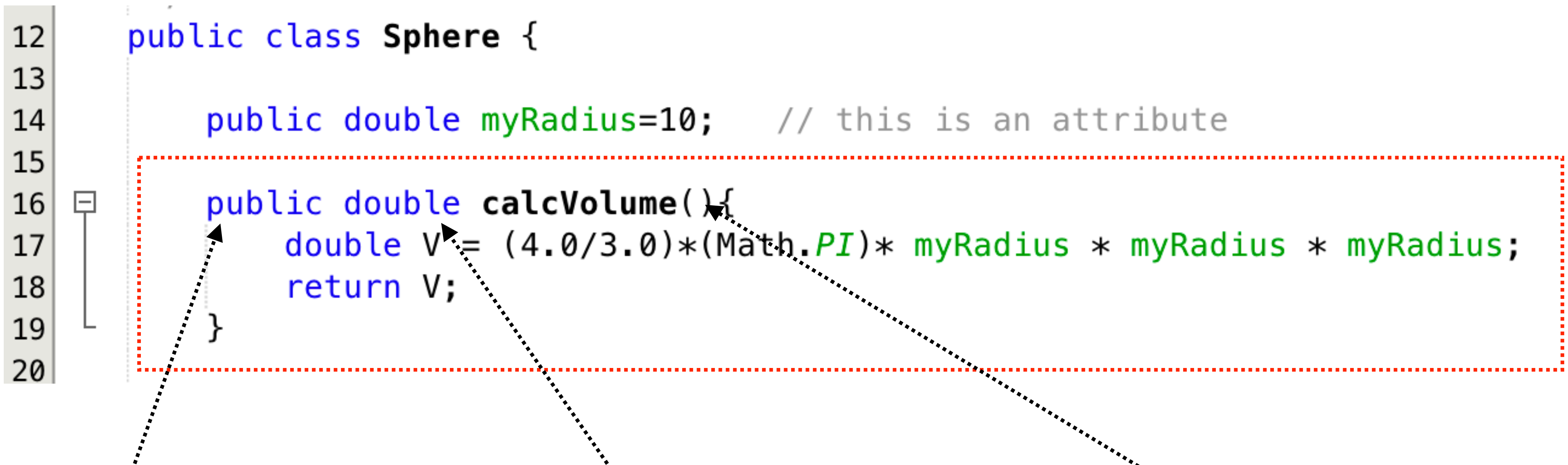


```
public class Sphere {

    public double myRadius=10;    // this is an attribute

    public static void main(String[] args) {
        Sphere mySphere = new Sphere();
        double V = (4.0/3.0)*(Math.PI)* mySphere.myRadius *mySphere.myRadius *mySphere.myRadius;
        System.out.println("Volume of our sphere is " + V);
    }
}
```

All that math looks messy. Can we modularize our code?

Let's build a method that calculates the volume and can be called inside main()

Objects, Attributes, and Methods

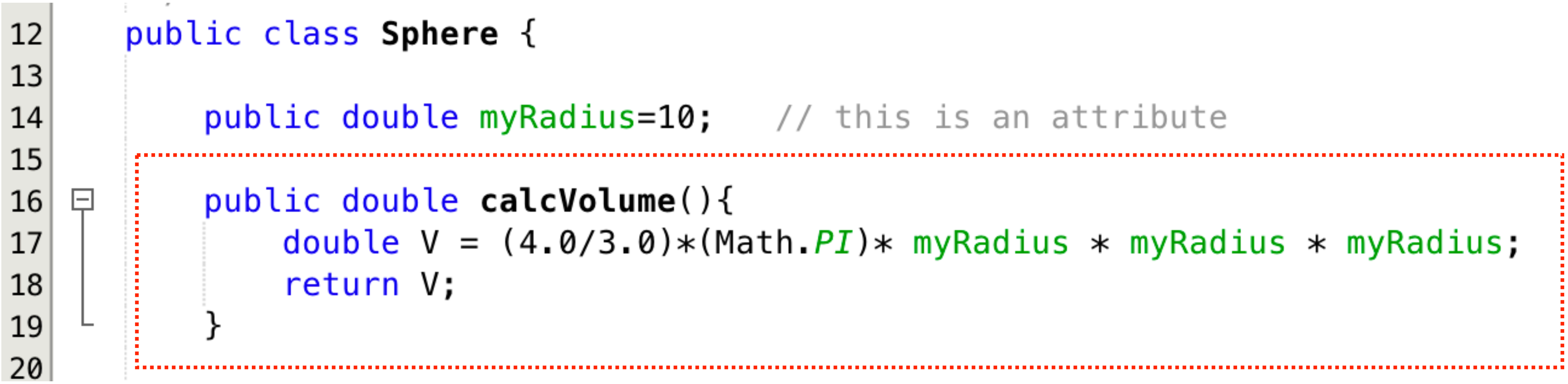


```
public class Sphere {

    public double myRadius=10;    // this is an attribute

    public double calcVolume(){
        double V = (4.0/3.0)*(Math.PI)* myRadius * myRadius * myRadius;
        return V;
    }

```

is a public method   return type is double   takes no arguments

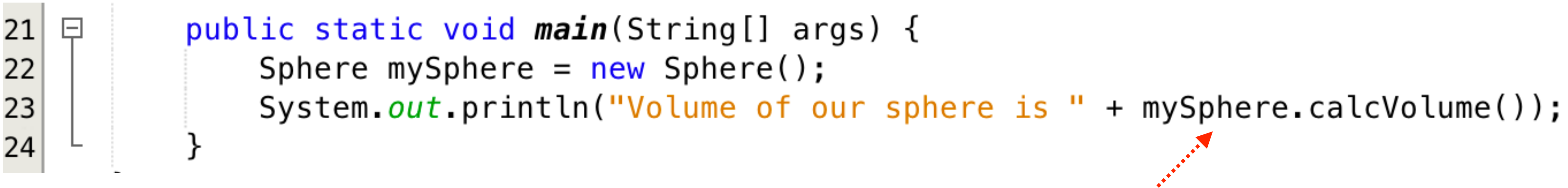
You must specify public (or private), whether it returns a type, and what arguments are fed to your method

Objects, Attributes, and Methods



```java
public class Sphere {

    public double myRadius=10;    // this is an attribute

    public double calcVolume(){
        double V = (4.0/3.0)*(Math.PI)* myRadius * myRadius * myRadius;
        return V;
    }

```

Then, print this in `main()`

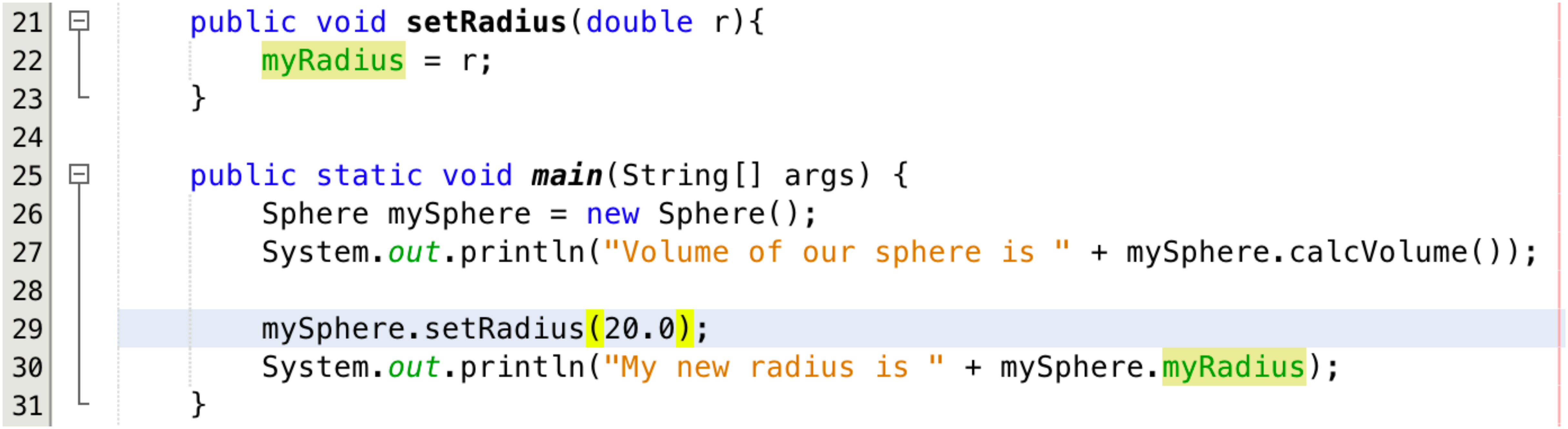


```java
    public static void main(String[] args) {
        Sphere mySphere = new Sphere();
        System.out.println("Volume of our sphere is " + mySphere.calcVolume());
    }
```

This is how we call the `calcVolume()` function

Objects, Attributes, and Methods

What if I'd like the change value of myRadius?



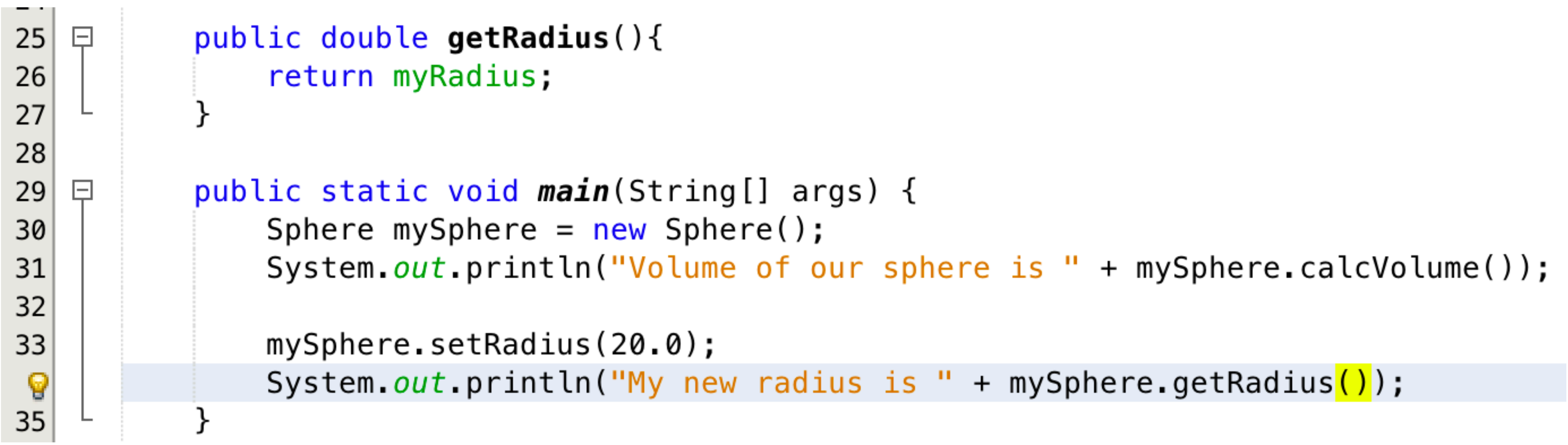
```
    public void setRadius(double r){
        myRadius = r;
    }

    public static void main(String[] args) {
        Sphere mySphere = new Sphere();
        System.out.println("Volume of our sphere is " + mySphere.calcVolume());

        mySphere.setRadius(20.0);
        System.out.println("My new radius is " + mySphere.myRadius);
    }
```

It's better practice to have set and get functions

Write a separate method to return the value of myRadius called `getRadius`

Objects, Attributes, and Methods

Write a separate method to return the value of myRadius called `getRadius`



```
    public double getRadius(){
        return myRadius;
    }

    public static void main(String[] args) {
        Sphere mySphere = new Sphere();
        System.out.println("Volume of our sphere is " + mySphere.calcVolume());

        mySphere.setRadius(20.0);
        System.out.println("My new radius is " + mySphere.getRadius());
    }
```

Now we have seen attributes, methods, and objects

Objects, Attributes, and Methods

Write a rectangle class that has

attributes for length and height

has get and set methods for both

has a method for calculating the area of the rectangle