**Question 01**

1. List down the access modifiers and data types that you can use to define the visibility and type of variables and methods in a class.

Access modifiers

* Public
* Private
* Protected

Data types

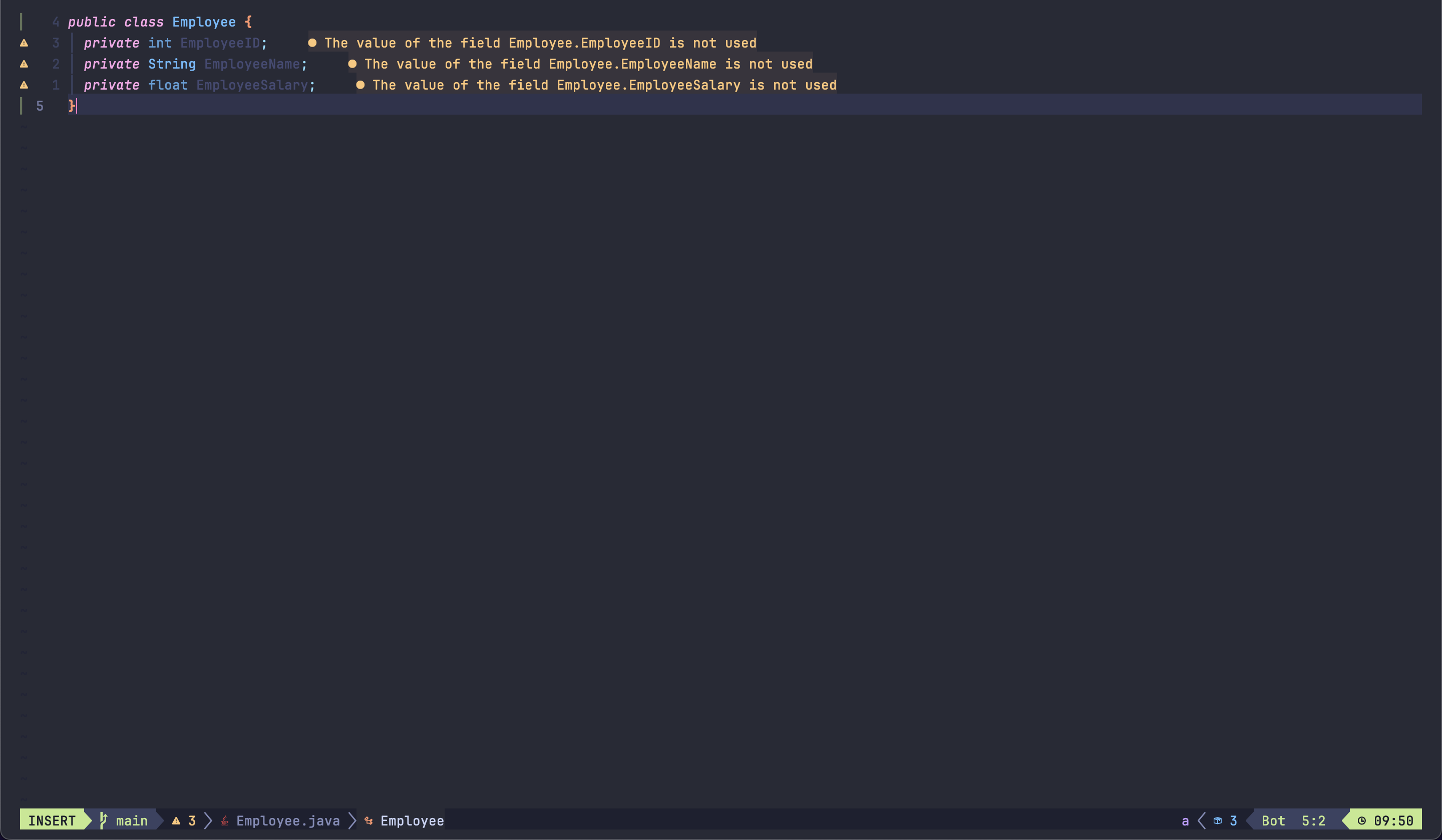
* Int
* String
* Char
* Long
* Float
* Double
* Byte
* Boolean
* Arrays

1. What is the purpose of getter and setter methods in Java?

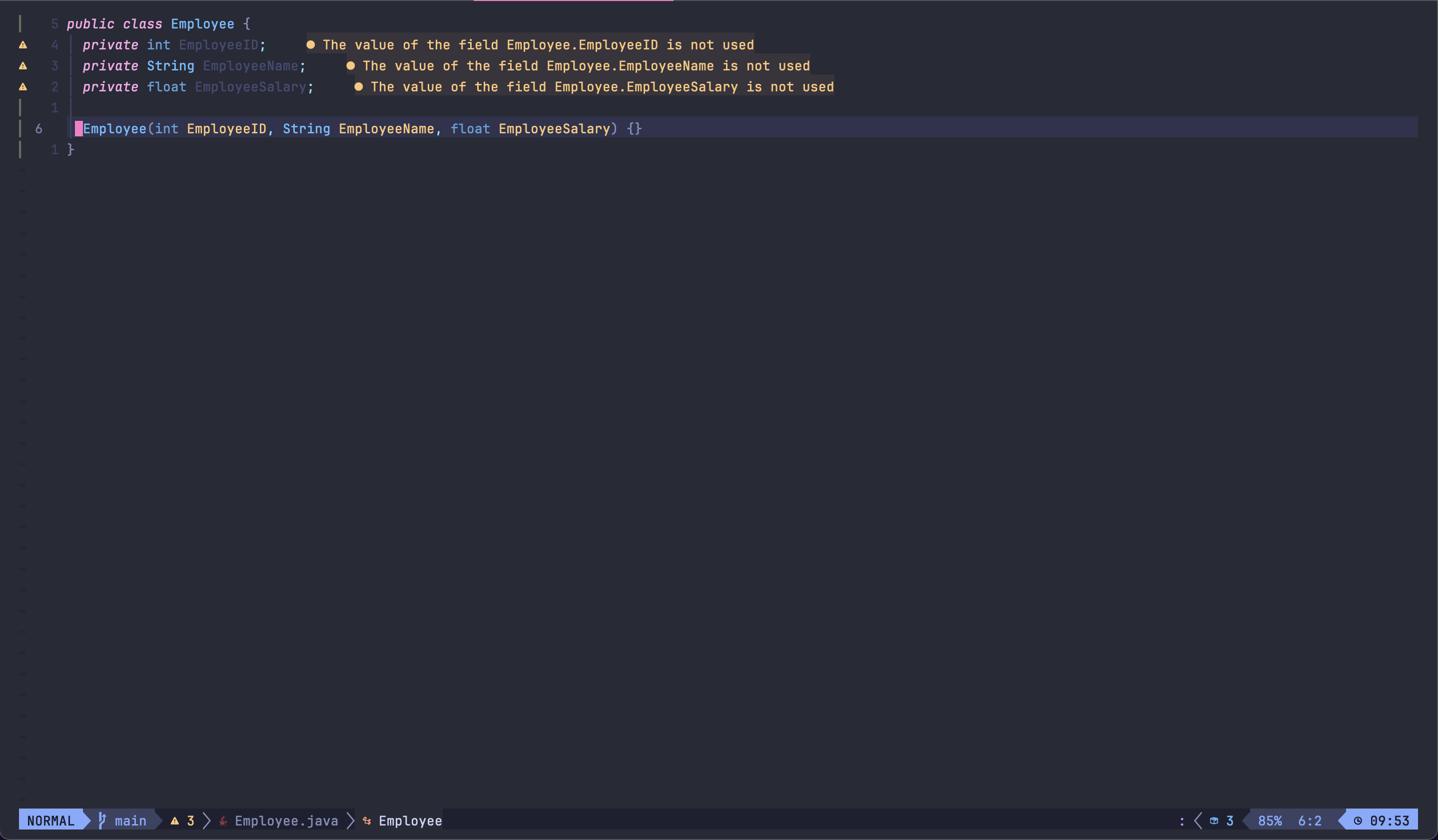
Let’s consider a situation where you have a class and have a variable that should not allowed be read then you would generally make it private but making the variable private removed the ability to change the value of the variable this is where the setter method comes in since even though the variable is private we can modify its value from a method within the class we use this ability and create a method called setter that accepts a parameter of the same data type as the private variable and use that method to change the value of the private variable.

Similarly, we use the getter method to read the value of a variable without changing its value in a similar fashion by defining a public method on that class called getter.

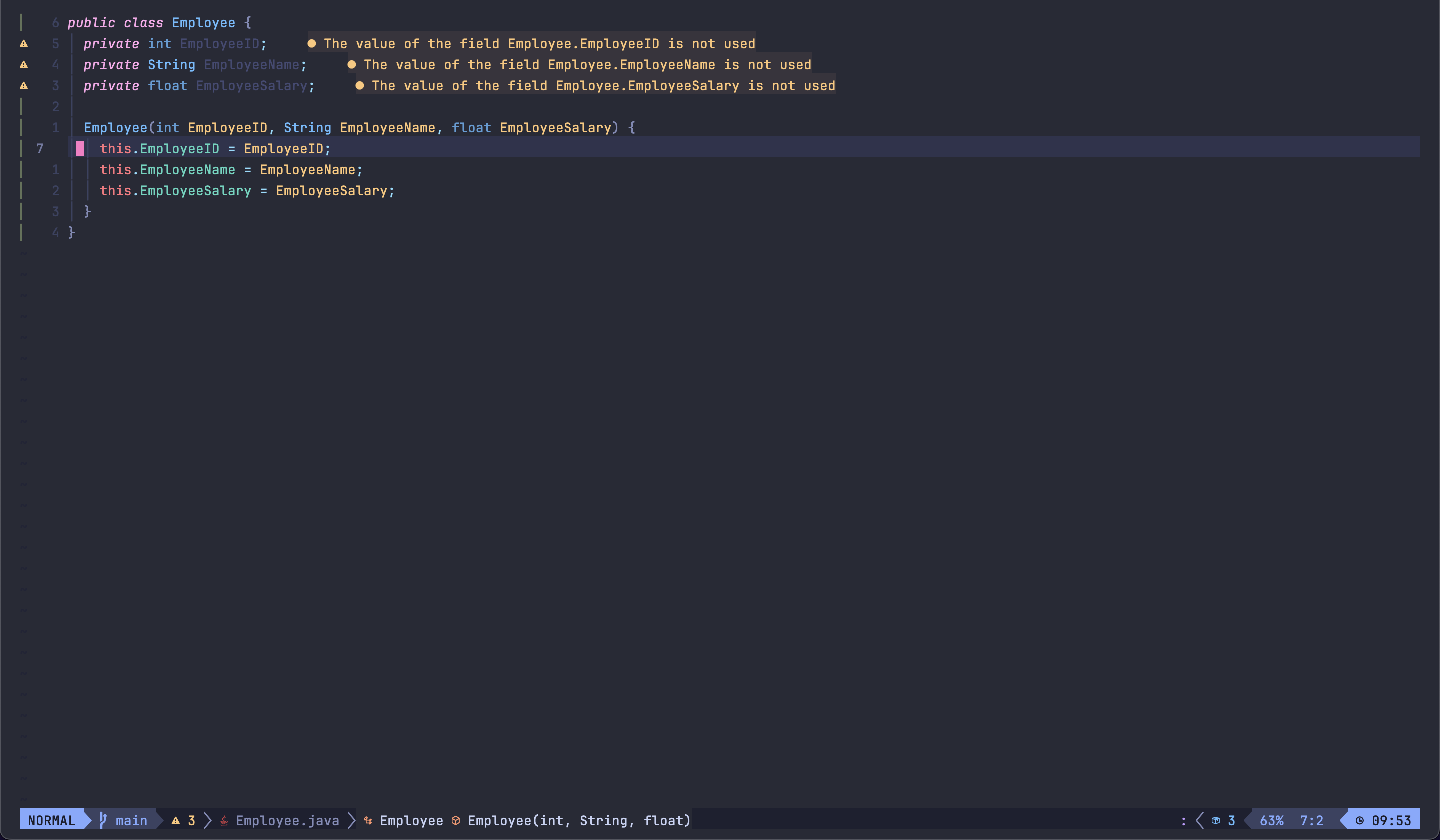
1. Create a new class called 'Employee' with three private instance variables, an integer variable called 'EmployeeID’, a String variable called 'EmployeeName', and a float variable called "Employee Salary.



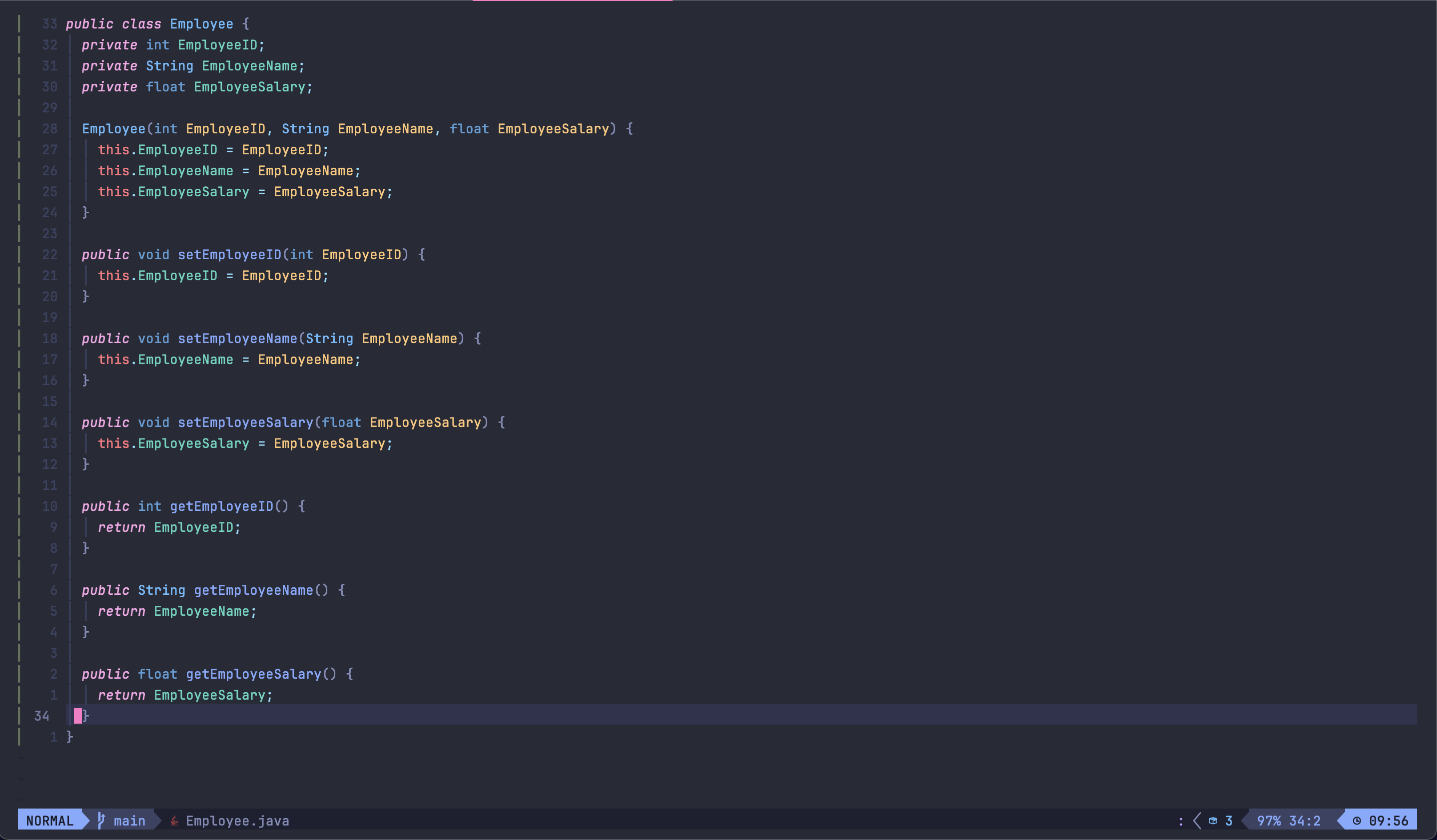
1. Add a constructor method for the Employee class that takes the above defined data types as arguments.



1. The constructor should assign the value of these parameters to the corresponding instance variables.



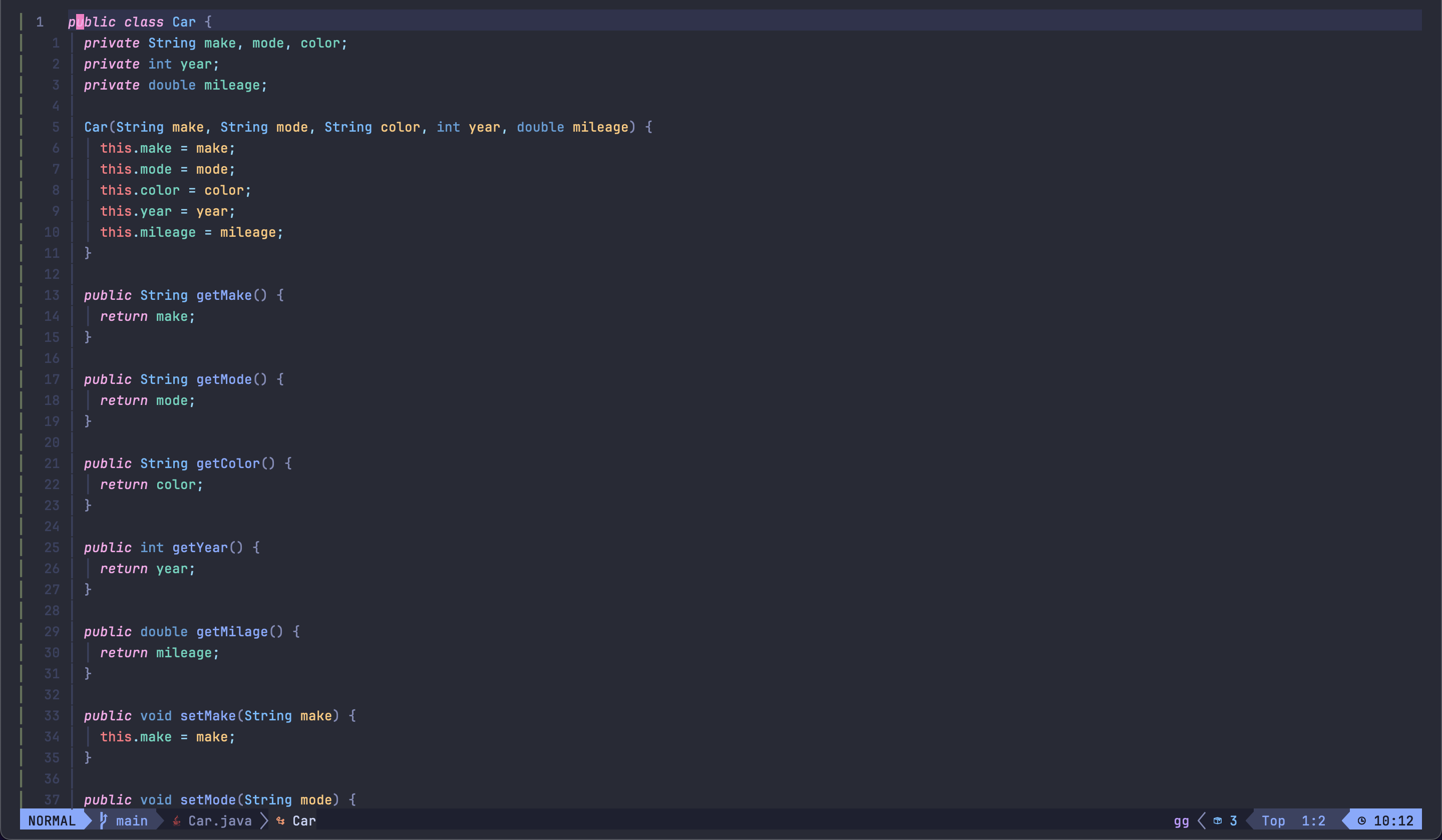
1. Add getter & setter methods for the above three variables.

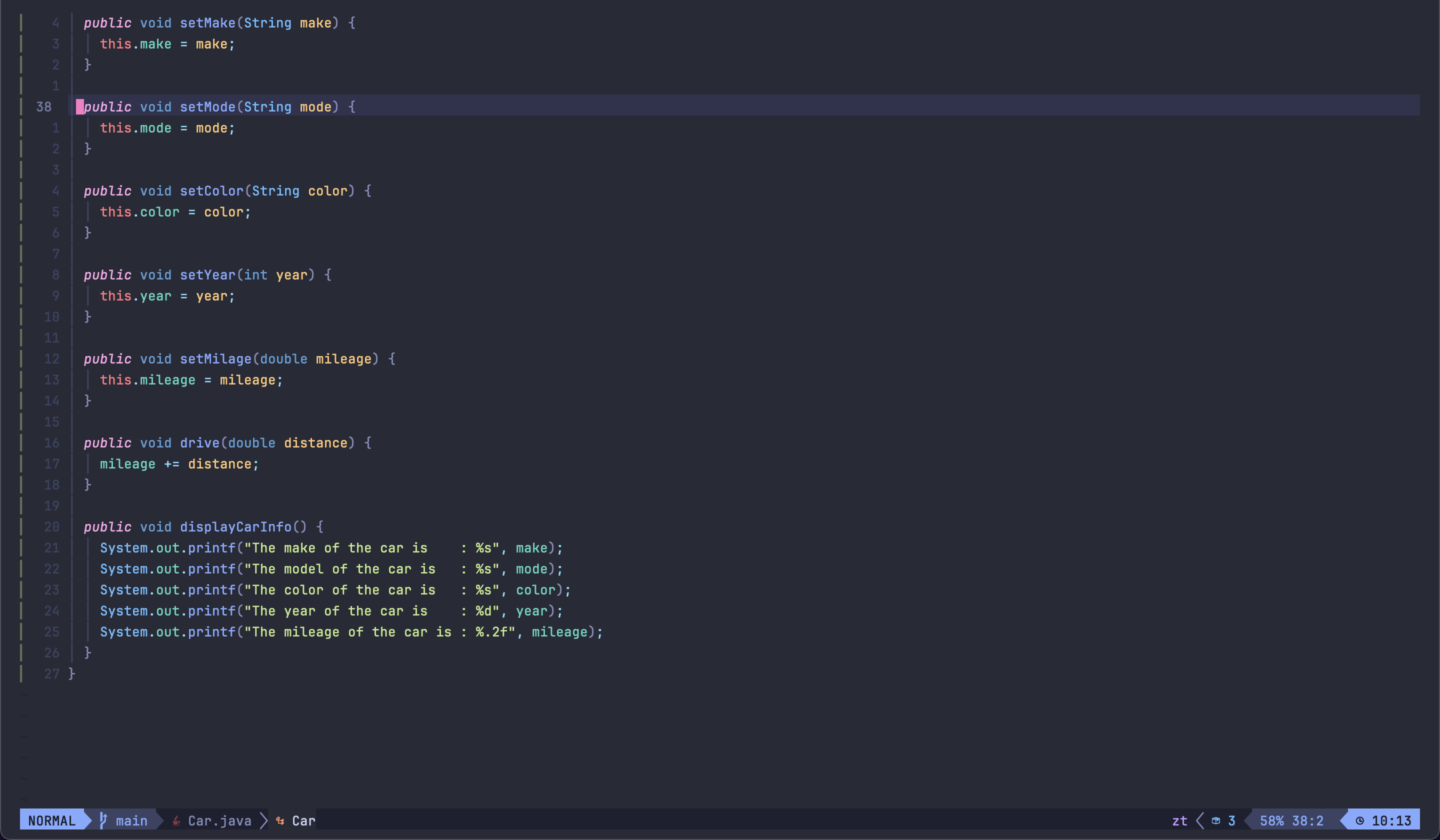


**Question 02**

You are tasked with implementing a Car class in java, the Car class should contain the following features:

1. Five private instance variables, where three String variables are "make", "mode", and "color", where another integer variable called "year" and the final variable as double variable called "mileage"
2. A parameterized constructor that takes make,model,color,year and mileage as arguments and initializes the corresponding instance variables.
3. A default constructor that initializes the instance variables with default values.
4. Getter and setter values for all the instance variables
5. A method called drive(distance) that takes a parameter distance (double) and increments the mileage instance variable by given distance.
6. A method called displayCarinfo() that displays the details of the car.

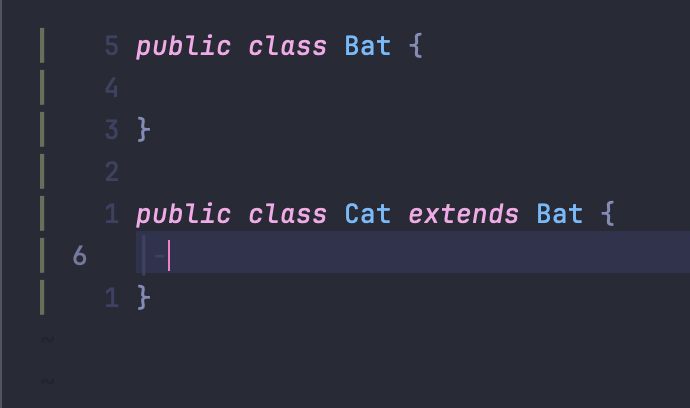




**Question 03**

1. How is Inheritance represented in Java syntax?

Inheritance is the scenario where one class is extends another class for the sake of code simplicity.

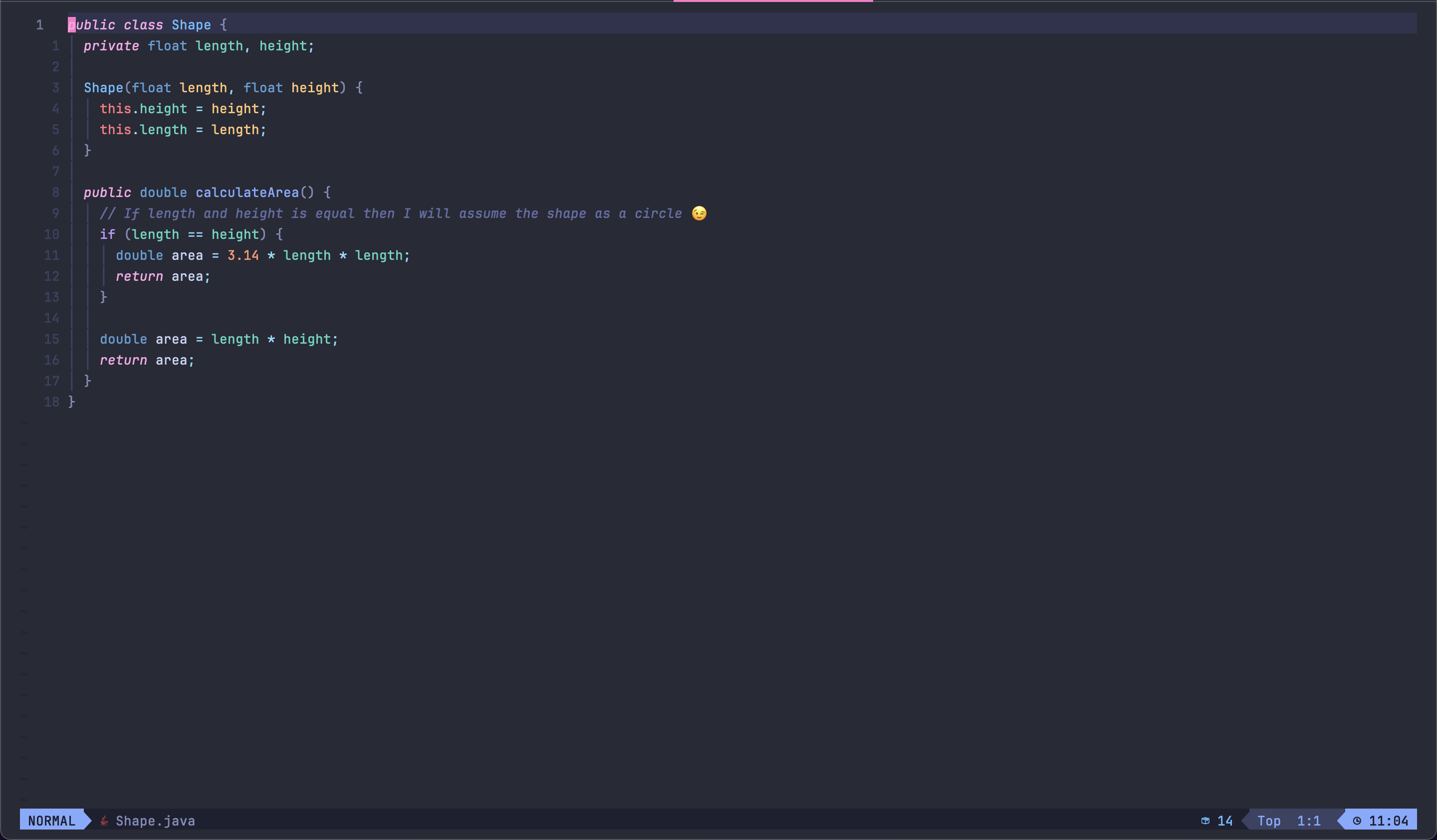


The above diagram represents a basic scenario where the Cat class inherits the Bat class.

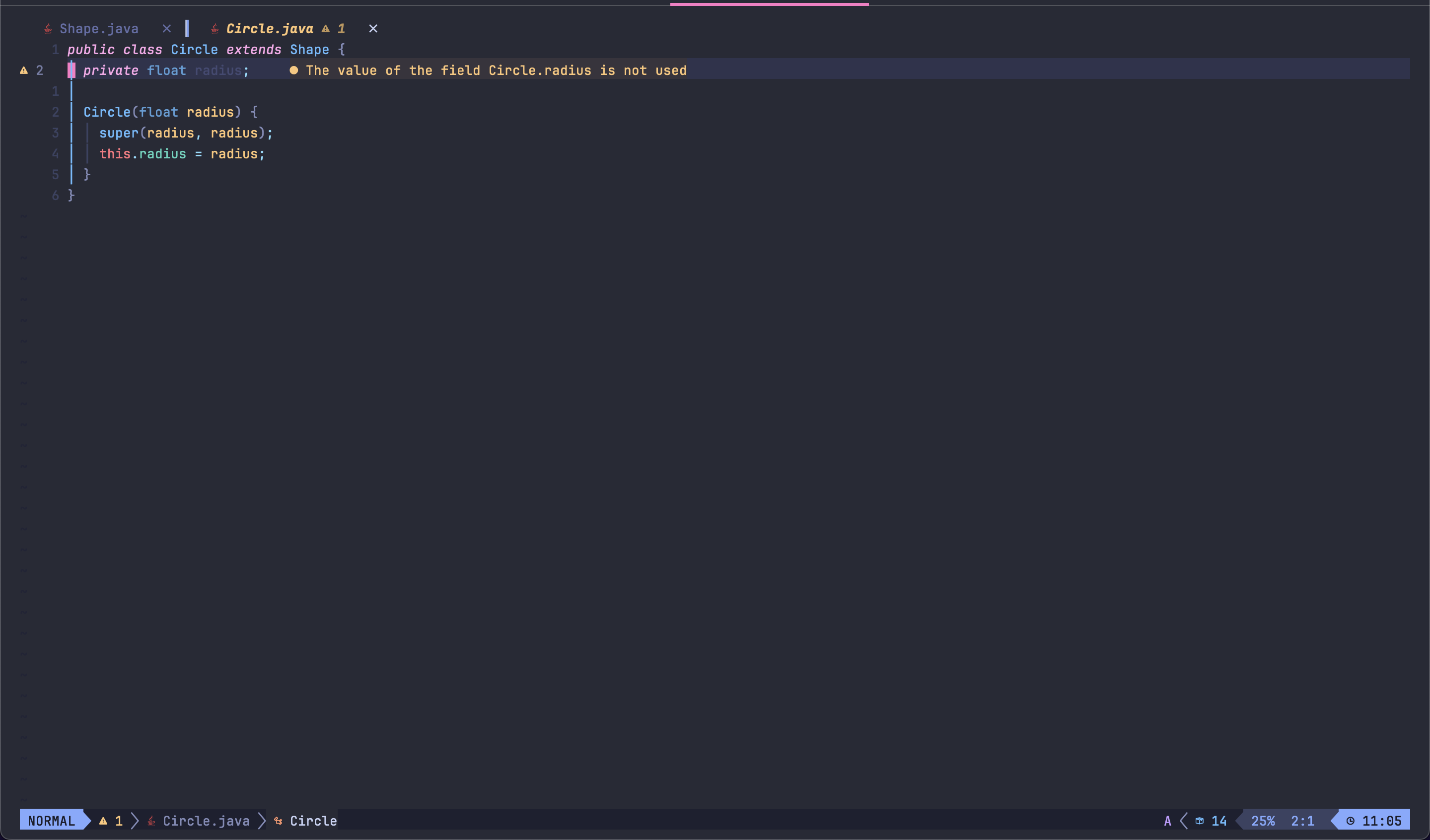
According to the above example the Bat class is the super class and the Cat class is the subclass.

1. Create a class called “Shape” with a method called “calculateArea().
2. Create two subclasses , “Circle” and “Rectangle’ , that inherit from the “Shape” class.
3. Implement the calculateArea() method in each subclass to calculate the area of a circle and a rectangle.
4. Test the implementation by creating objects of both subclasses and calling the calculateArea() method on each object.

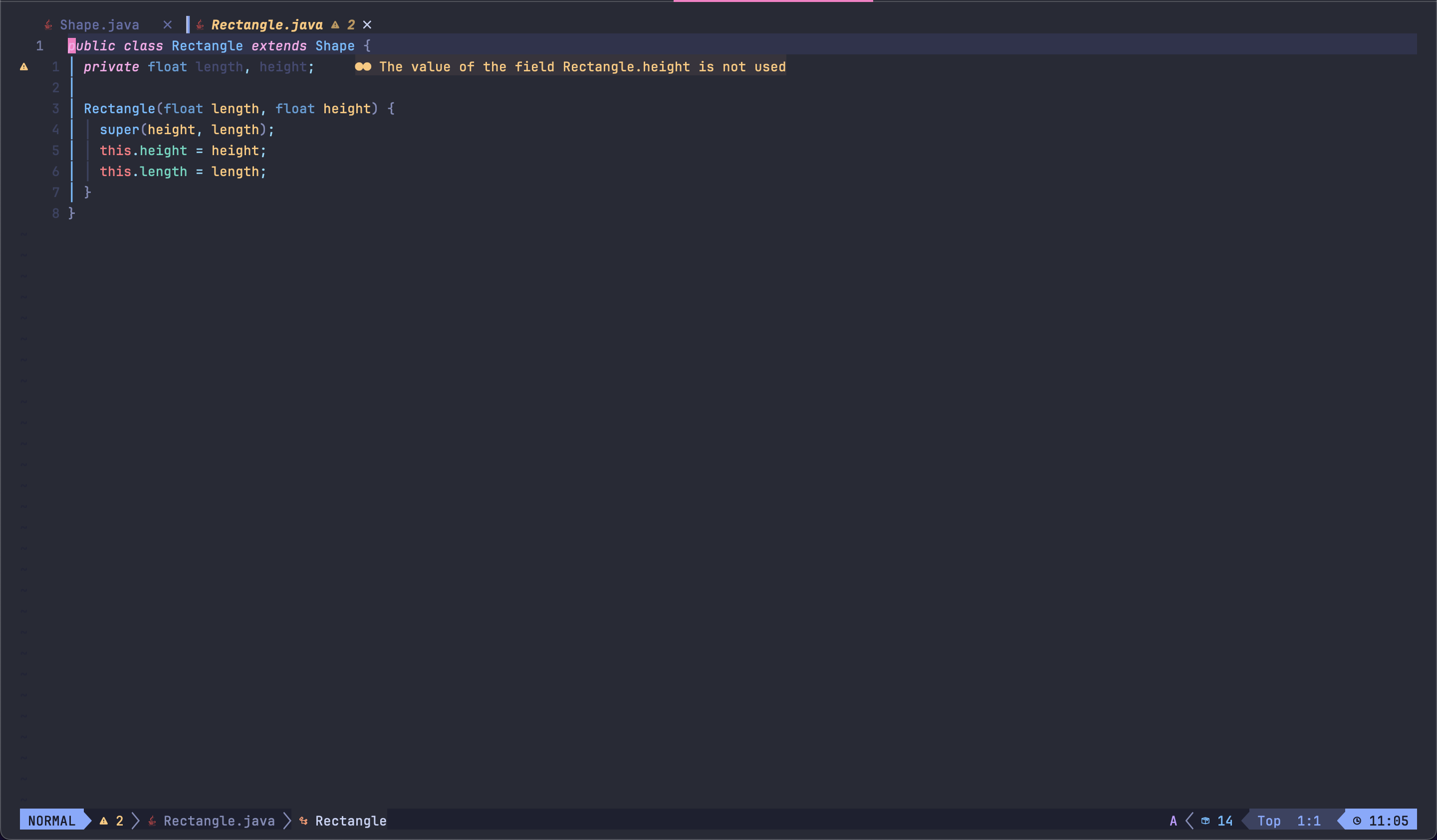
**Shape.java**

****

**Circle.java**

****

**Rectangle.java**

****

**Main.java**

Contains creating objects with the above created classes.

