Department of Computer Science & Engineering
UNIVERSITY OF LIBERAL ARTS
BANGLADESH

Lab Report 7

Submitted to:
Shakib Mahmud Dipto
Faculty

Submitted by:
Sumaiya Akter
ID: 201014071

Department of CSE
Summer'24

Course code: CSE 2104
Course Title: Object Oriented Programming Lab
Section: 01

University of Liberal Arts Bangladesh

September 20, 2024

Problem 01

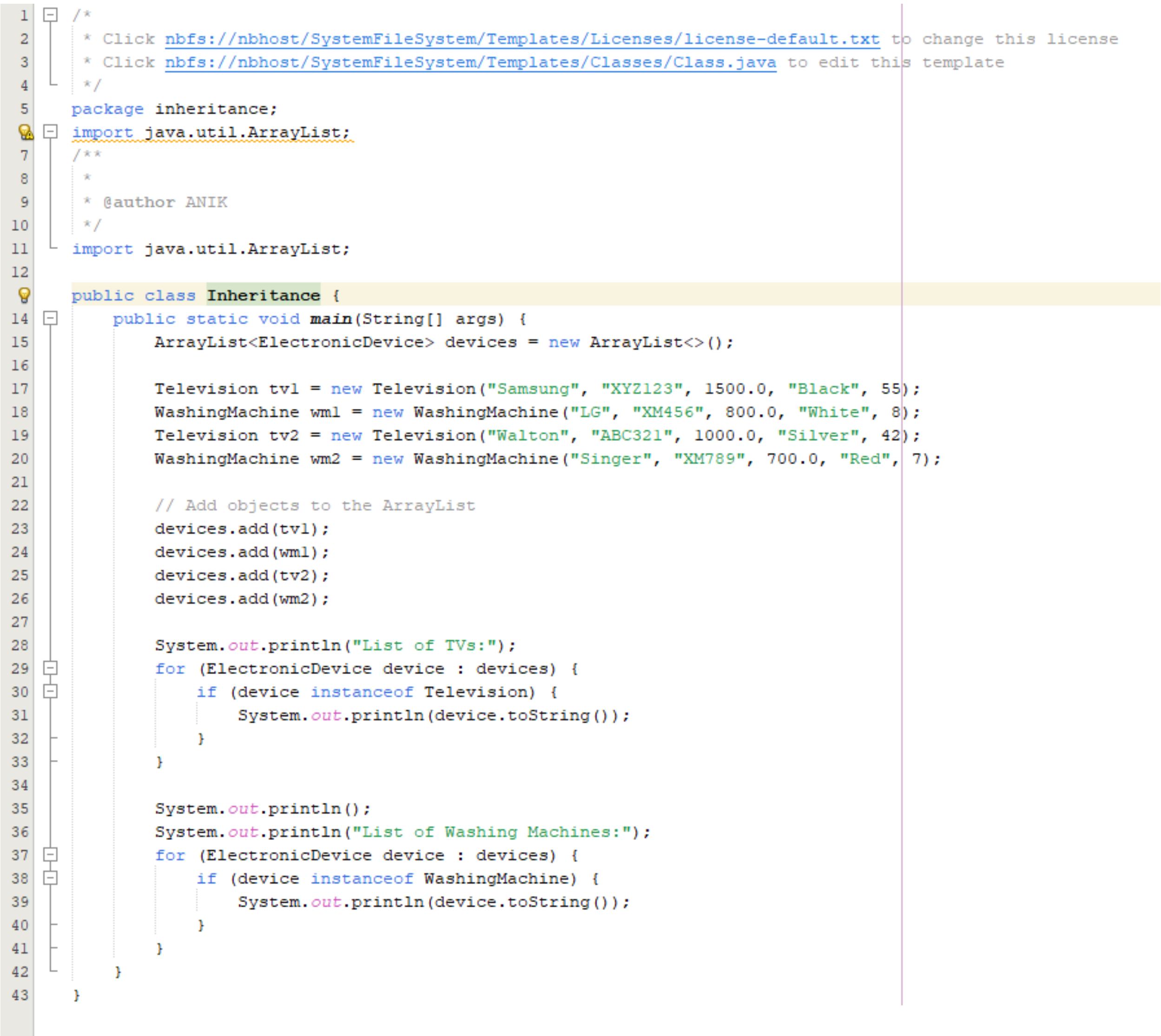
Code Explanation :

This Java code defines a simple program to manage electronic devices, specifically televisions and washing machines. It uses inheritance to create a base class, ‘ElectronicDevice’, which holds common attributes like company, model, price, and color. The ‘Television’ and ‘WashingMachine’ classes extend this base class, adding specific features. The ‘Inheritance’ class contains the ‘main’ method, where instances of these devices are created, stored in a list, and displayed based on their type.

Code Screenshots:

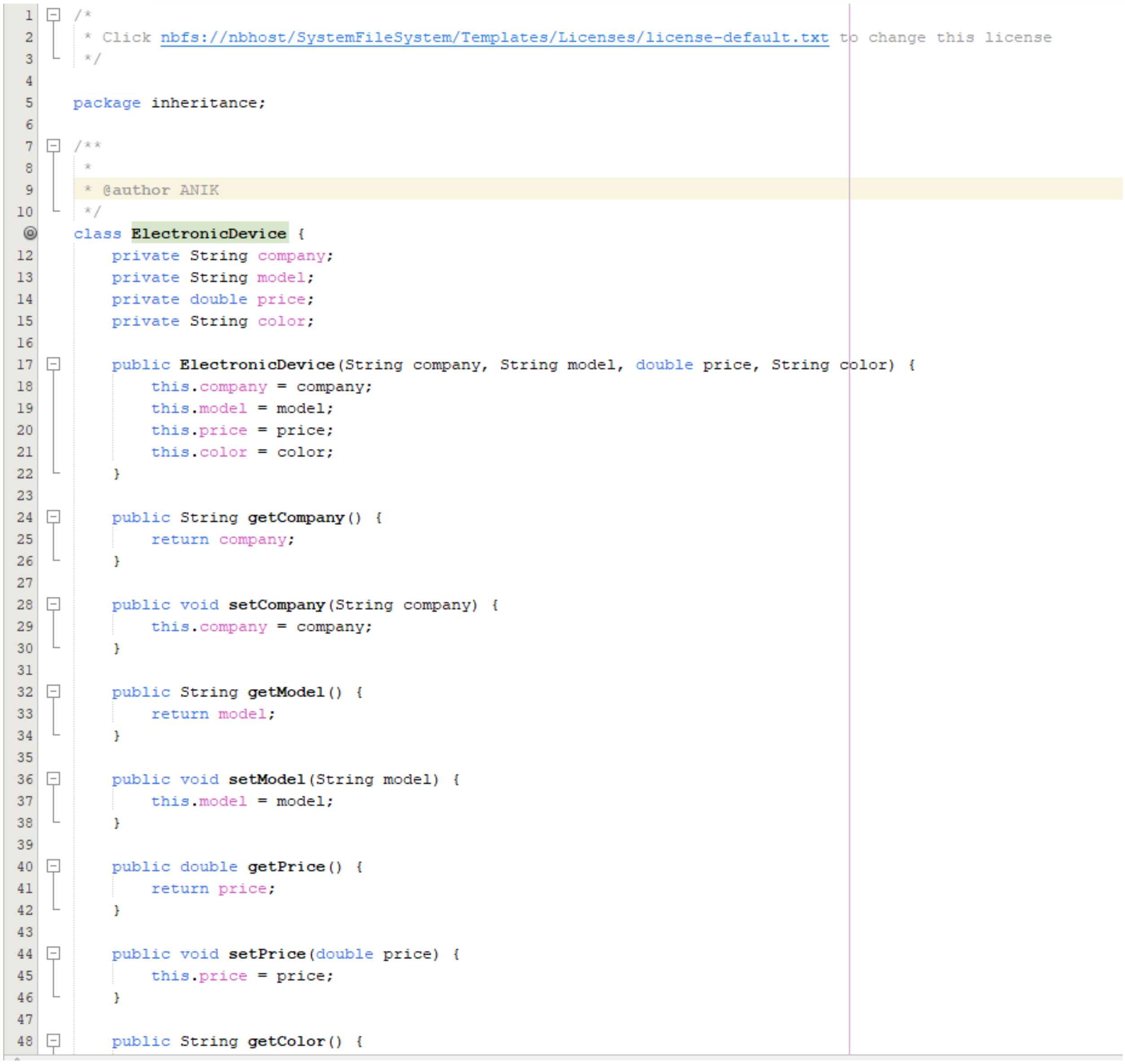
Input:

Inheritance.java

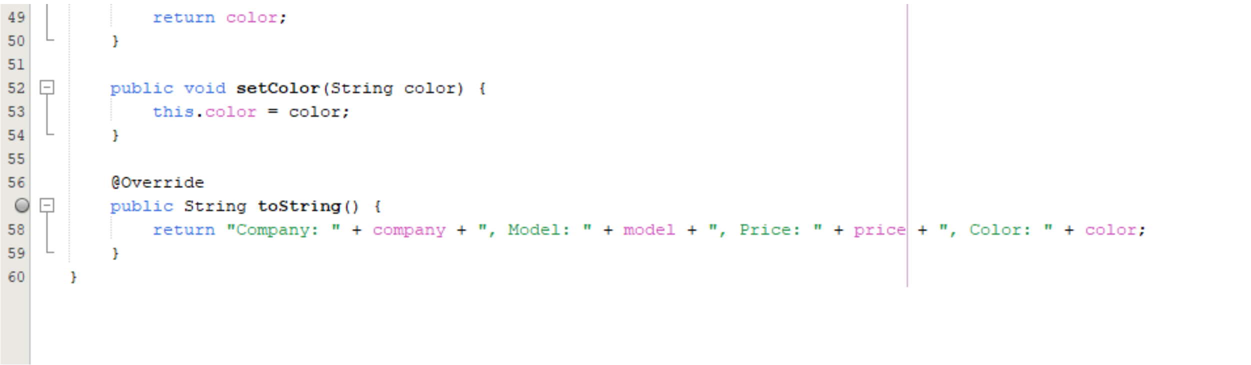


```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package inheritance;
import java.util.ArrayList;
/**
 *
 * @author ANIK
 */
import java.util.ArrayList;

public class Inheritance {
    public static void main(String[] args) {
        ArrayList<ElectronicDevice> devices = new ArrayList<>();

        Television tv1 = new Television("Samsung", "XYZ123", 1500.0, "Black", 55);
        WashingMachine wm1 = new WashingMachine("LG", "XM456", 800.0, "White", 8);
        Television tv2 = new Television("Walton", "ABC321", 1000.0, "Silver", 42);
        WashingMachine wm2 = new WashingMachine("Singer", "XM789", 700.0, "Red", 7);

        // Add objects to the ArrayList
        devices.add(tv1);
        devices.add(wm1);
        devices.add(tv2);
        devices.add(wm2);

        System.out.println("List of TVs:");
        for (ElectronicDevice device : devices) {
            if (device instanceof Television) {
                System.out.println(device.toString());
            }
        }

        System.out.println();
        System.out.println("List of Washing Machines:");
        for (ElectronicDevice device : devices) {
            if (device instanceof WashingMachine) {
                System.out.println(device.toString());
            }
        }
    }
}
```

ElectronicDevice.java

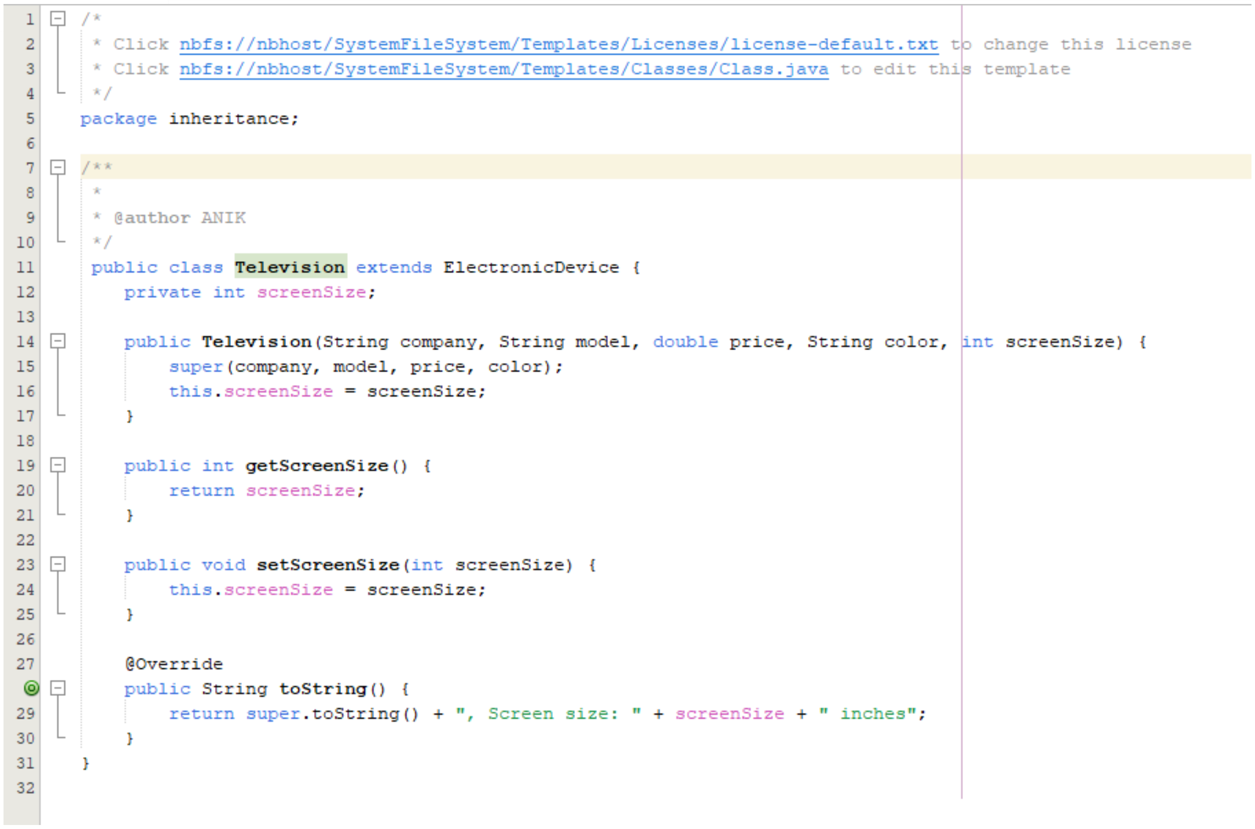


```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */

package inheritance;

/**
 *
 * @author ANIK
 */
class ElectronicDevice {
    private String company;
    private String model;
    private double price;
    private String color;

    public ElectronicDevice(String company, String model, double price, String color) {
        this.company = company;
        this.model = model;
        this.price = price;
        this.color = color;
    }

    public String getCompany() {
        return company;
    }

    public void setCompany(String company) {
        this.company = company;
    }

    public String getModel() {
        return model;
    }

    public void setModel(String model) {
        this.model = model;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public String getColor() {
```



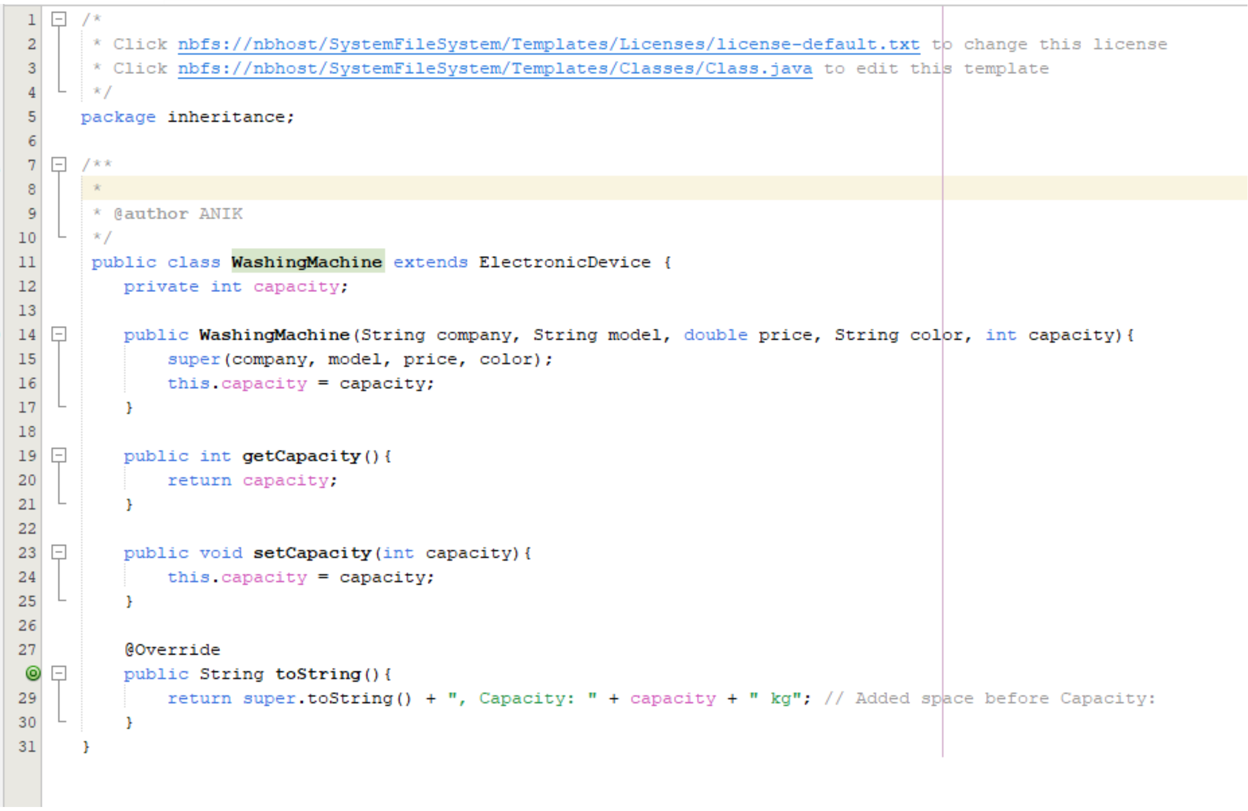
```
        return color;
    }

    public void setColor(String color) {
        this.color = color;
    }

    @Override
    public String toString() {
        return "Company: " + company + ", Model: " + model + ", Price: " + price + ", Color: " + color;
    }
}
```

Television.java



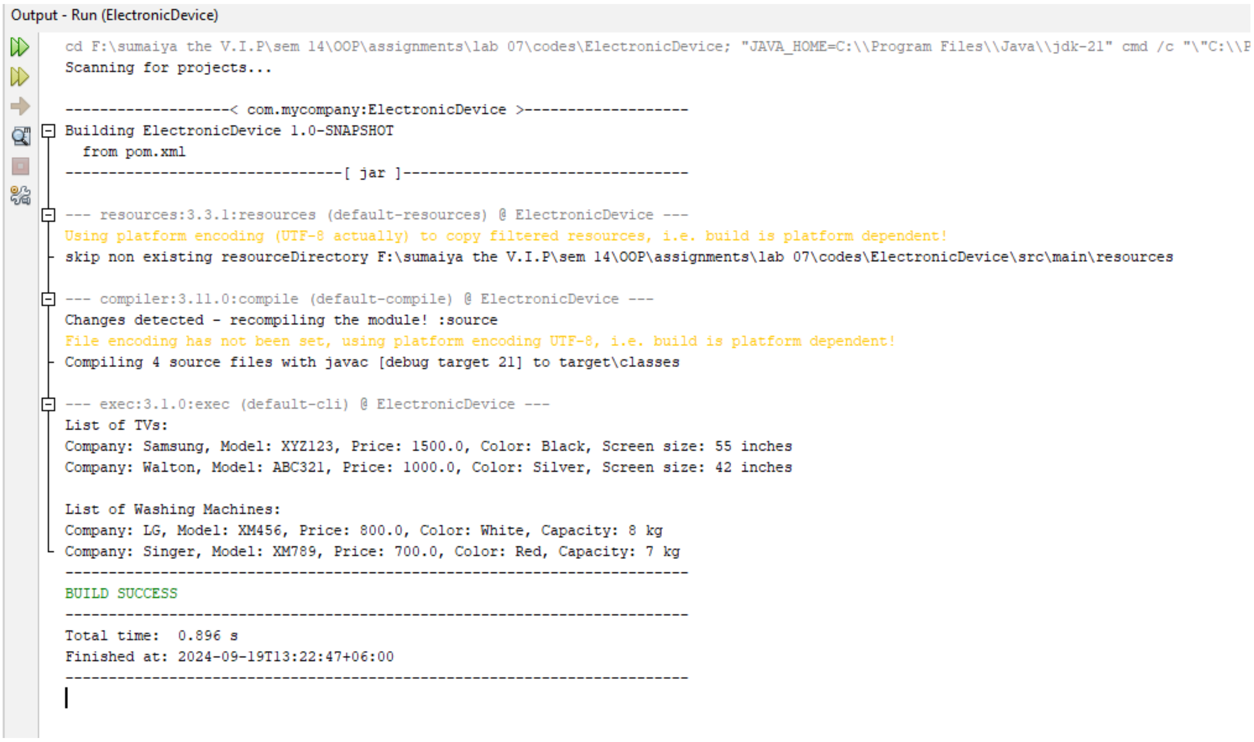
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package inheritance;

/**
 *
 * @author ANIK
 */
 public class Television extends ElectronicDevice {
    private int screenSize;

    public Television(String company, String model, double price, String color, int screenSize) {
        super(company, model, price, color);
        this.screenSize = screenSize;
    }

    public int getScreenSize() {
        return screenSize;
    }

    public void setScreenSize(int screenSize) {
        this.screenSize = screenSize;
    }

    @Override
    public String toString() {
        return super.toString() + ", Screen size: " + screenSize + " inches";
    }
}
```

WashingMachine.java



```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package inheritance;

/**
 *
 * @author ANIK
 */
public class WashingMachine extends ElectronicDevice {
    private int capacity;

    public WashingMachine(String company, String model, double price, String color, int capacity){
        super(company, model, price, color);
        this.capacity = capacity;
    }

    public int getCapacity(){
        return capacity;
    }

    public void setCapacity(int capacity){
        this.capacity = capacity;
    }

    @Override
    public String toString(){
        return super.toString() + ", Capacity: " + capacity + " kg"; // Added space before Capacity:
    }
}
```

Output:



Output - Run (ElectronicDevice)

```
cd F:\sumaiya the V.I.P\sem 14\OOP\assignments\lab 07\codes\ElectronicDevice; "JAVA_HOME=C:\\Program Files\\Java\\jdk-21" cmd /c "\"C:\\P
Scanning for projects...

-------------------< com.mycompany:ElectronicDevice >-------------------
Building ElectronicDevice 1.0-SNAPSHOT
  from pom.xml
--------------------------------[ jar ]---------------------------------

--- resources:3.3.1:resources (default-resources) @ ElectronicDevice ---
Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
skip non existing resourceDirectory F:\sumaiya the V.I.P\sem 14\OOP\assignments\lab 07\codes\ElectronicDevice\src\main\resources

--- compiler:3.11.0:compile (default-compile) @ ElectronicDevice ---
Changes detected - recompiling the module! :source
File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
Compiling 4 source files with javac [debug target 21] to target\classes

--- exec:3.1.0:exec (default-cli) @ ElectronicDevice ---
List of TVs:
Company: Samsung, Model: XYZ123, Price: 1500.0, Color: Black, Screen size: 55 inches
Company: Walton, Model: ABC321, Price: 1000.0, Color: Silver, Screen size: 42 inches

List of Washing Machines:
Company: LG, Model: XM456, Price: 800.0, Color: White, Capacity: 8 kg
Company: Singer, Model: XM789, Price: 700.0, Color: Red, Capacity: 7 kg
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.896 s
Finished at: 2024-09-19T13:22:47+06:00
------------------------------------------------------------------------
```

Practice Problem 01

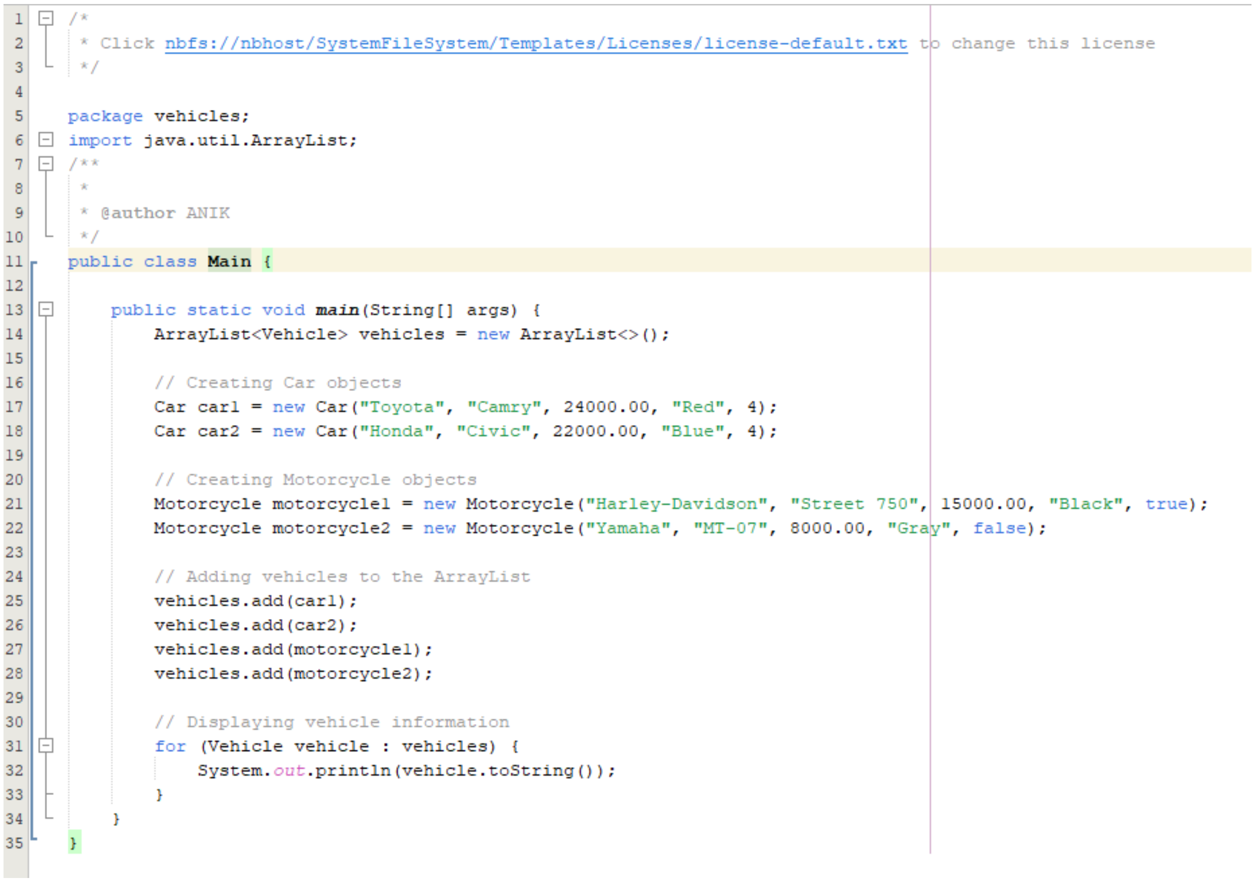
Code Explanation:

This Java program models vehicles using a base class called 'Vehicle', which includes common attributes like brand, model, price, and color. Two subclasses, 'Car' and 'Motorcycle', extend 'Vehicle', adding specific features such as the number of doors for cars and the presence of a sidecar for motorcycles. The main class initializes instances of these vehicles, stores them in an 'ArrayList', and displays their details, showcasing the power of inheritance and method overriding.

Code Screenshot:

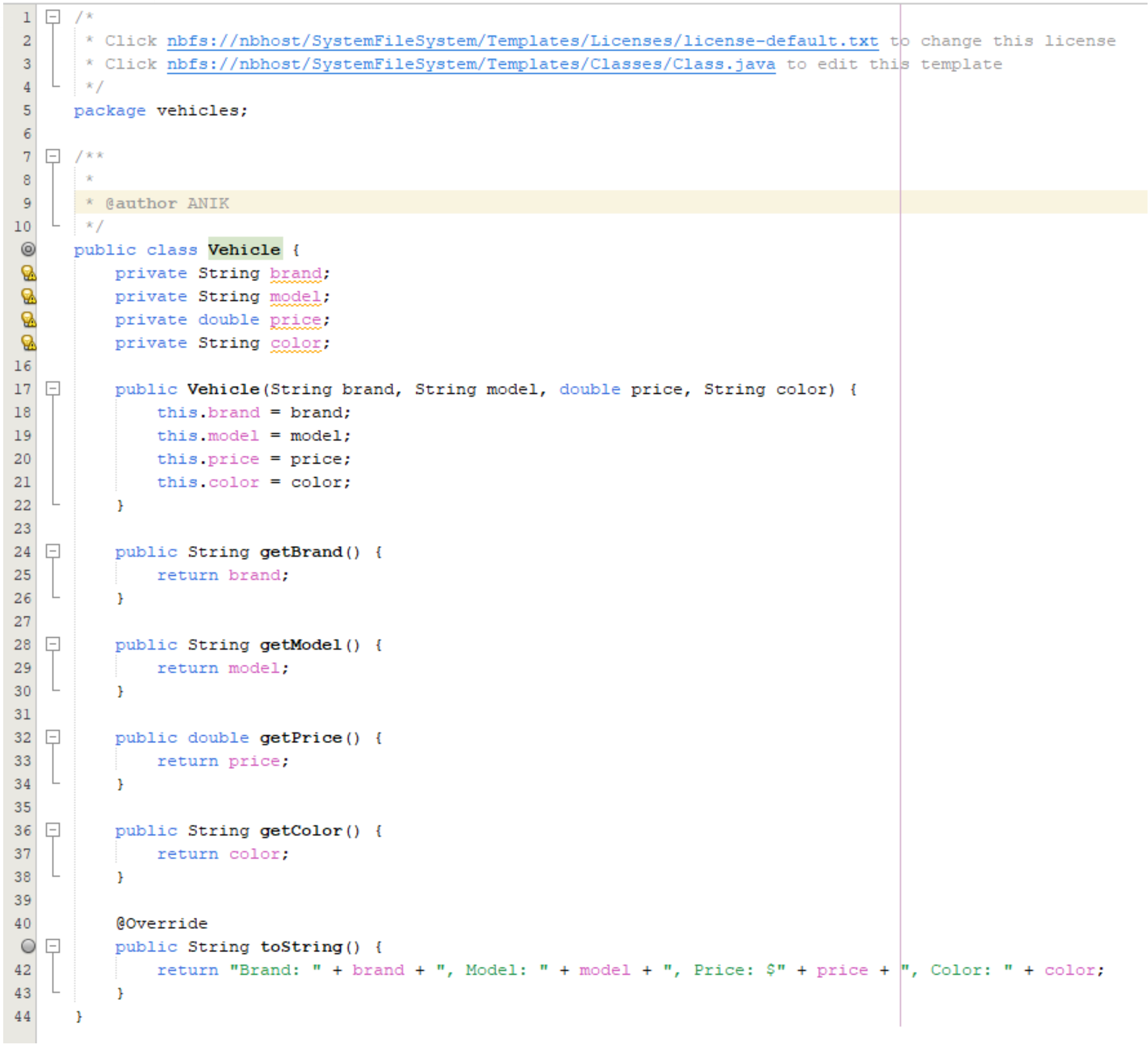
Input:

Main.java



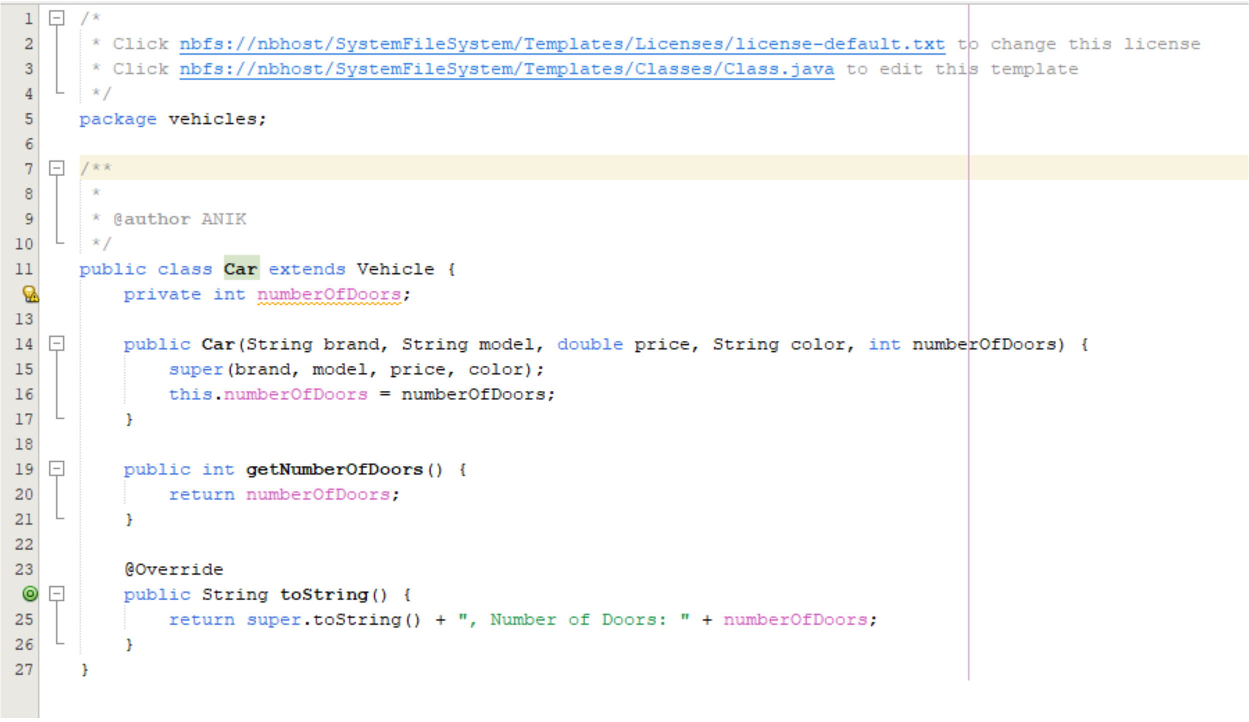
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */

package vehicles;
import java.util.ArrayList;
/**
 *
 * @author ANIK
 */
public class Main {

    public static void main(String[] args) {
        ArrayList<Vehicle> vehicles = new ArrayList<>();

        // Creating Car objects
        Car car1 = new Car("Toyota", "Camry", 24000.00, "Red", 4);
        Car car2 = new Car("Honda", "Civic", 22000.00, "Blue", 4);

        // Creating Motorcycle objects
        Motorcycle motorcycle1 = new Motorcycle("Harley-Davidson", "Street 750", 15000.00, "Black", true);
        Motorcycle motorcycle2 = new Motorcycle("Yamaha", "MT-07", 8000.00, "Gray", false);

        // Adding vehicles to the ArrayList
        vehicles.add(car1);
        vehicles.add(car2);
        vehicles.add(motorcycle1);
        vehicles.add(motorcycle2);

        // Displaying vehicle information
        for (Vehicle vehicle : vehicles) {
            System.out.println(vehicle.toString());
        }
    }
}
```

Vehicle.java



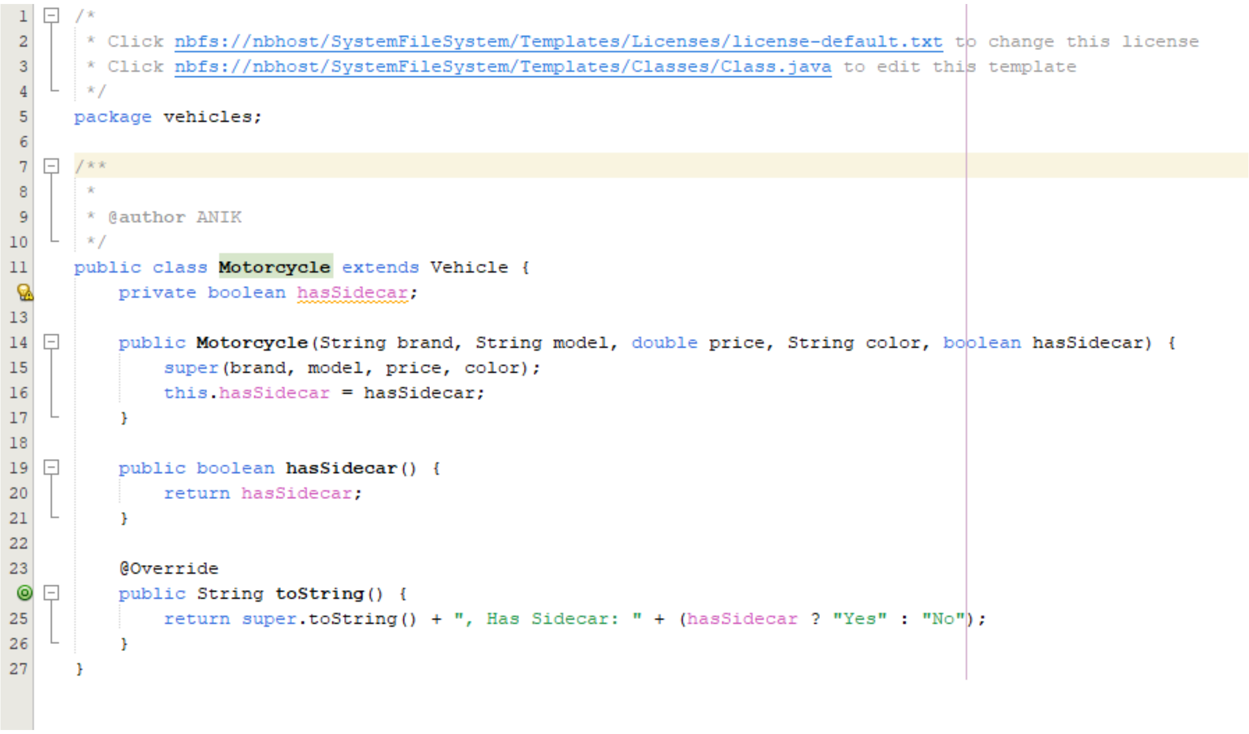
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package vehicles;

/**
 *
 * @author ANIK
 */
public class Vehicle {
    private String brand;
    private String model;
    private double price;
    private String color;

    public Vehicle(String brand, String model, double price, String color) {
        this.brand = brand;
        this.model = model;
        this.price = price;
        this.color = color;
    }

    public String getBrand() {
        return brand;
    }

    public String getModel() {
        return model;
    }

    public double getPrice() {
        return price;
    }

    public String getColor() {
        return color;
    }

    @Override
    public String toString() {
        return "Brand: " + brand + ", Model: " + model + ", Price: $" + price + ", Color: " + color;
    }
}
```

Car.java



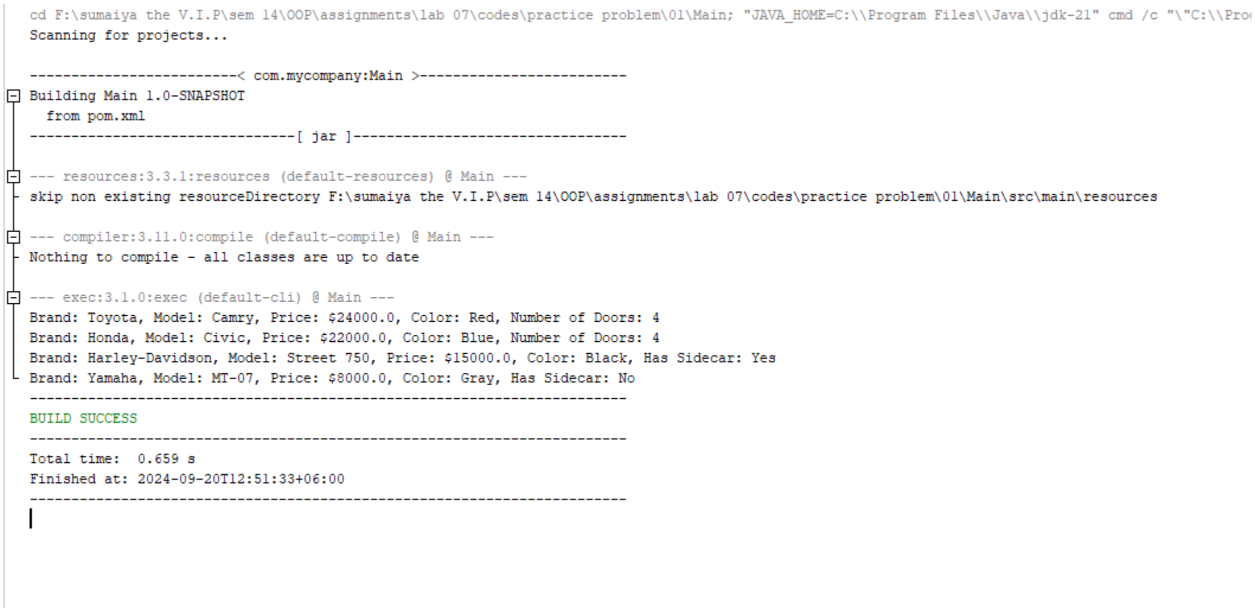
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package vehicles;

/**
 *
 * @author ANIK
 */
public class Car extends Vehicle {
    private int numberOfDoors;

    public Car(String brand, String model, double price, String color, int numberOfDoors) {
        super(brand, model, price, color);
        this.numberOfDoors = numberOfDoors;
    }

    public int getNumberOfDoors() {
        return numberOfDoors;
    }

    @Override
    public String toString() {
        return super.toString() + ", Number of Doors: " + numberOfDoors;
    }
}
```

Motorcycle.java



```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package vehicles;

/**
 *
 * @author ANIK
 */
public class Motorcycle extends Vehicle {
    private boolean hasSidecar;

    public Motorcycle(String brand, String model, double price, String color, boolean hasSidecar) {
        super(brand, model, price, color);
        this.hasSidecar = hasSidecar;
    }

    public boolean hasSidecar() {
        return hasSidecar;
    }

    @Override
    public String toString() {
        return super.toString() + ", Has Sidecar: " + (hasSidecar ? "Yes" : "No");
    }
}
```

Output:



```
cd F:\sumaiya the V.I.P\sem 14\OOP\assignments\lab 07\codes\practice problem\01\Main; "JAVA_HOME=C:\\Program Files\\Java\\jdk-21" cmd /c "\"C:\\Pro
Scanning for projects...

------------------------< com.mycompany:Main >------------------------
Building Main 1.0-SNAPSHOT
  from pom.xml
--------------------------------[ jar ]---------------------------------

--- resources:3.3.1:resources (default-resources) @ Main ---
skip non existing resourceDirectory F:\sumaiya the V.I.P\sem 14\OOP\assignments\lab 07\codes\practice problem\01\Main\src\main\resources

--- compiler:3.11.0:compile (default-compile) @ Main ---
Nothing to compile - all classes are up to date

--- exec:3.1.0:exec (default-cli) @ Main ---
Brand: Toyota, Model: Camry, Price: $24000.0, Color: Red, Number of Doors: 4
Brand: Honda, Model: Civic, Price: $22000.0, Color: Blue, Number of Doors: 4
Brand: Harley-Davidson, Model: Street 750, Price: $15000.0, Color: Black, Has Sidecar: Yes
Brand: Yamaha, Model: MT-07, Price: $8000.0, Color: Gray, Has Sidecar: No
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.659 s
Finished at: 2024-09-20T12:51:33+06:00
------------------------------------------------------------------------
```

Practice Problem 02

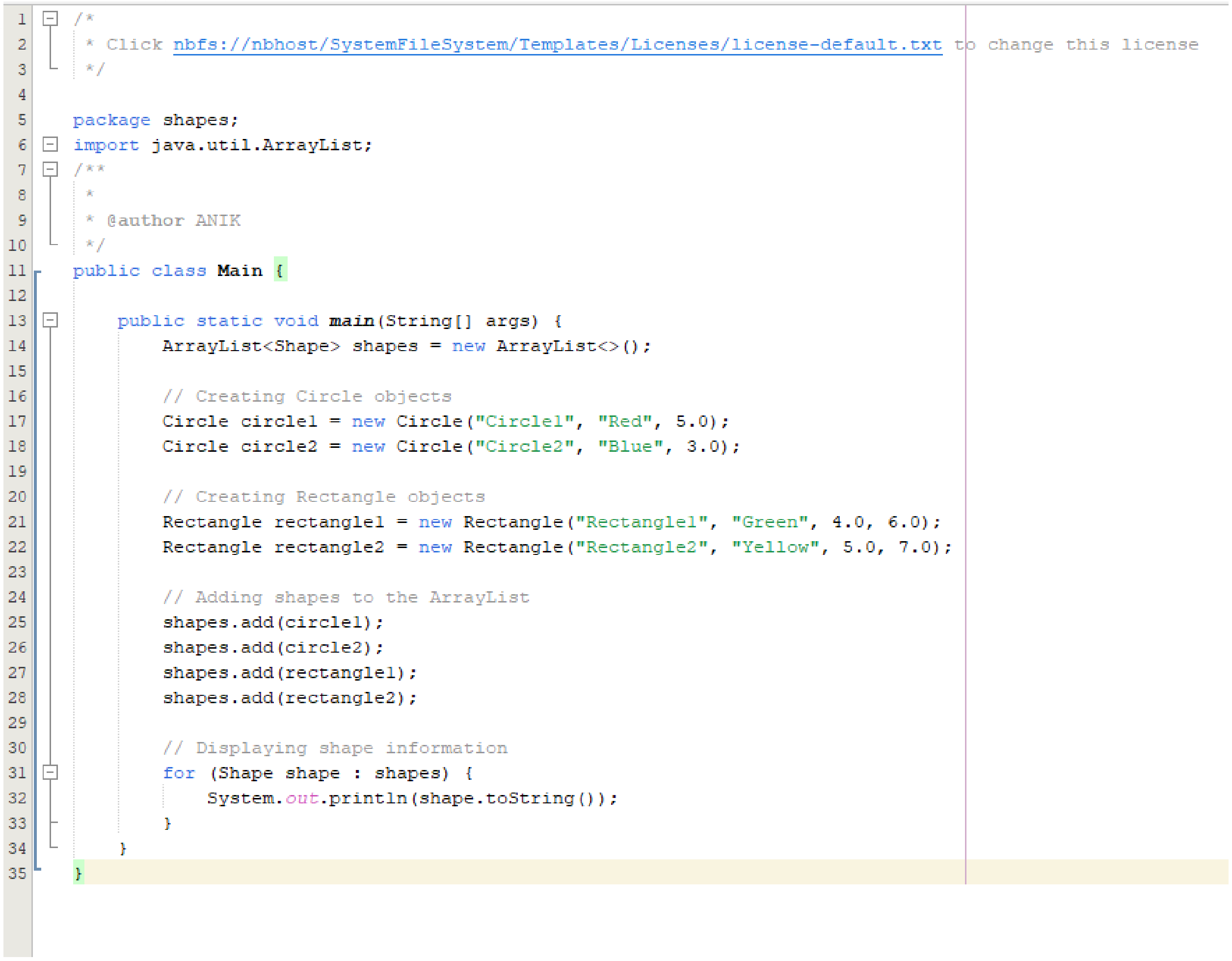
Code Explanation:

This Java program models geometric shapes using an abstract class called ‘Shape’, which includes properties like name and color. Two subclasses, ‘Circle’ and ‘Rectangle’, extend this base class, implementing methods to calculate their area and perimeter. The main class initializes instances of these shapes, storing them in an ‘ArrayList’. It then displays each shape's details, showcasing the power of inheritance and polymorphism in organizing and managing different shape types efficiently.

Code Screenshot:

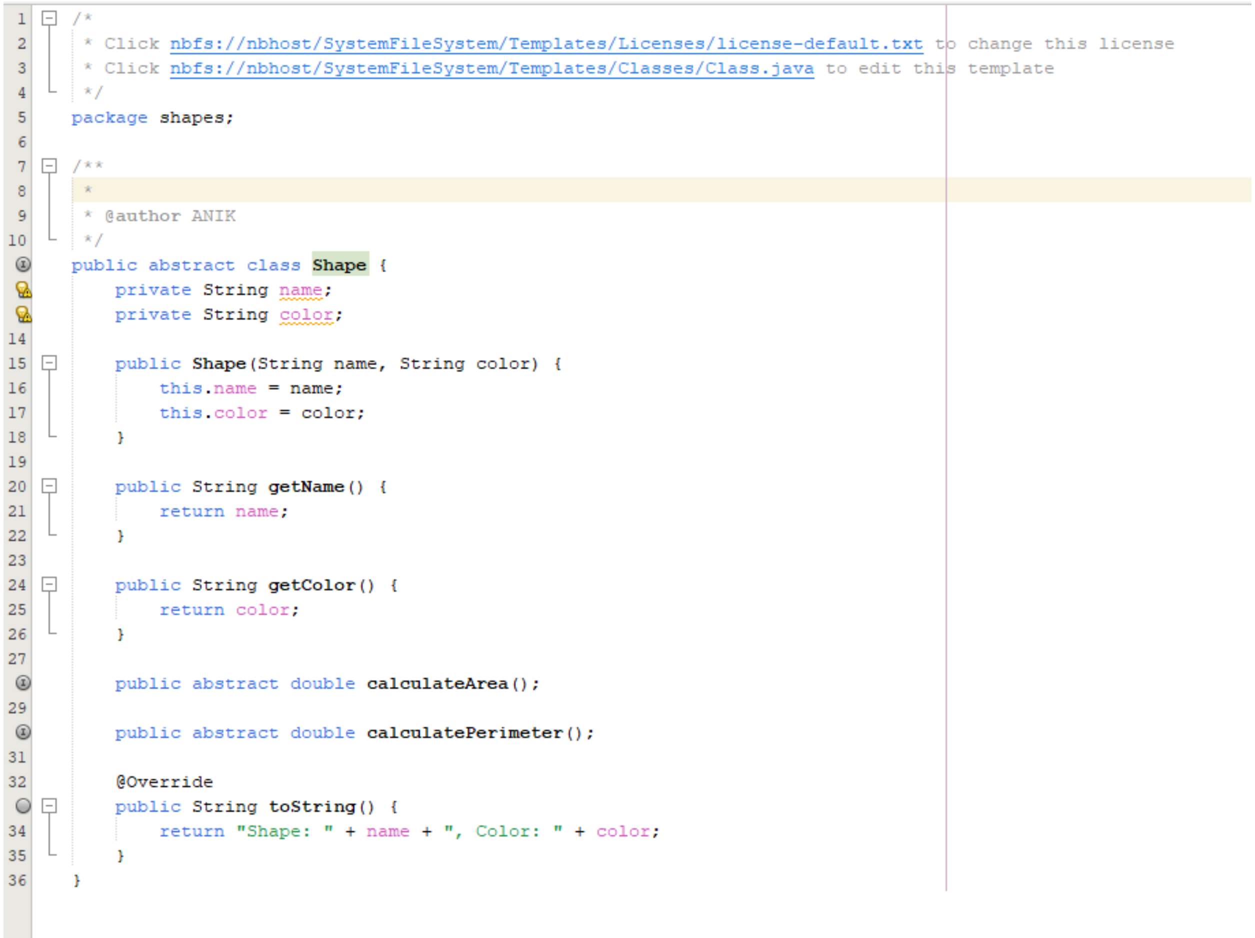
Input:

Main.java



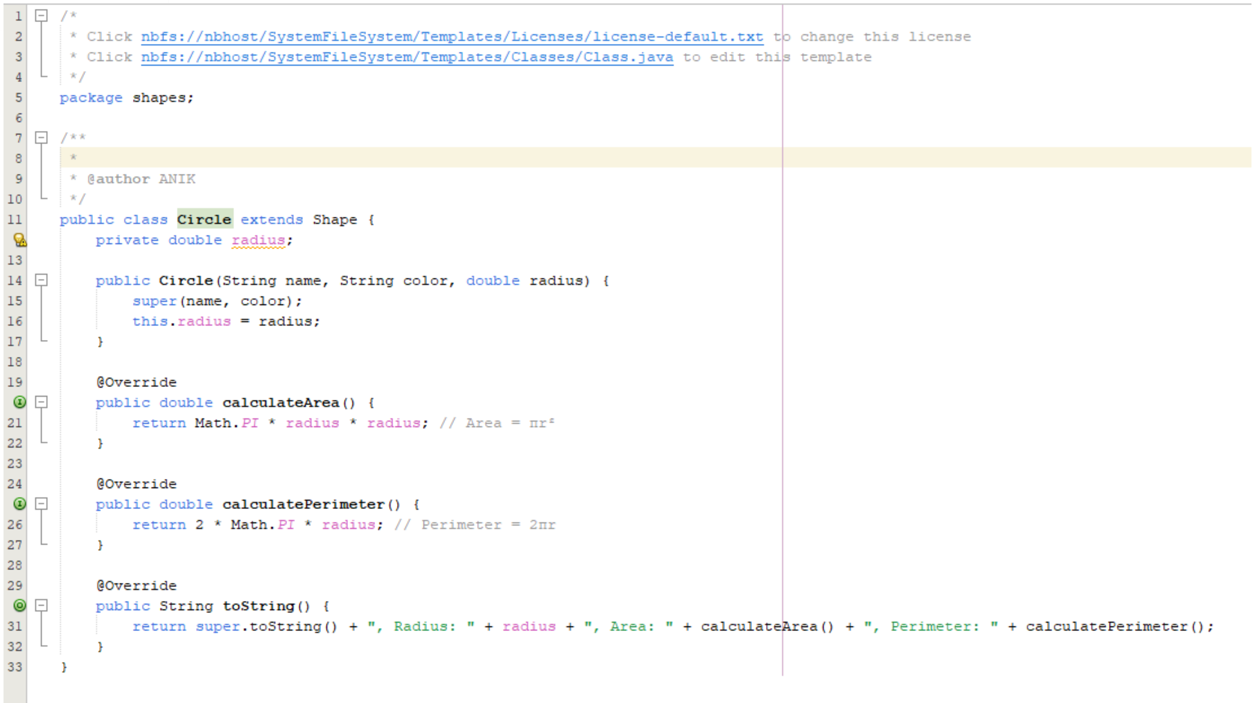
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */

package shapes;
import java.util.ArrayList;
/**
 *
 * @author ANIK
 */
public class Main {

    public static void main(String[] args) {
        ArrayList<Shape> shapes = new ArrayList<>();

        // Creating Circle objects
        Circle circle1 = new Circle("Circle1", "Red", 5.0);
        Circle circle2 = new Circle("Circle2", "Blue", 3.0);

        // Creating Rectangle objects
        Rectangle rectangle1 = new Rectangle("Rectangle1", "Green", 4.0, 6.0);
        Rectangle rectangle2 = new Rectangle("Rectangle2", "Yellow", 5.0, 7.0);

        // Adding shapes to the ArrayList
        shapes.add(circle1);
        shapes.add(circle2);
        shapes.add(rectangle1);
        shapes.add(rectangle2);

        // Displaying shape information
        for (Shape shape : shapes) {
            System.out.println(shape.toString());
        }
    }
}
```

Shape.java



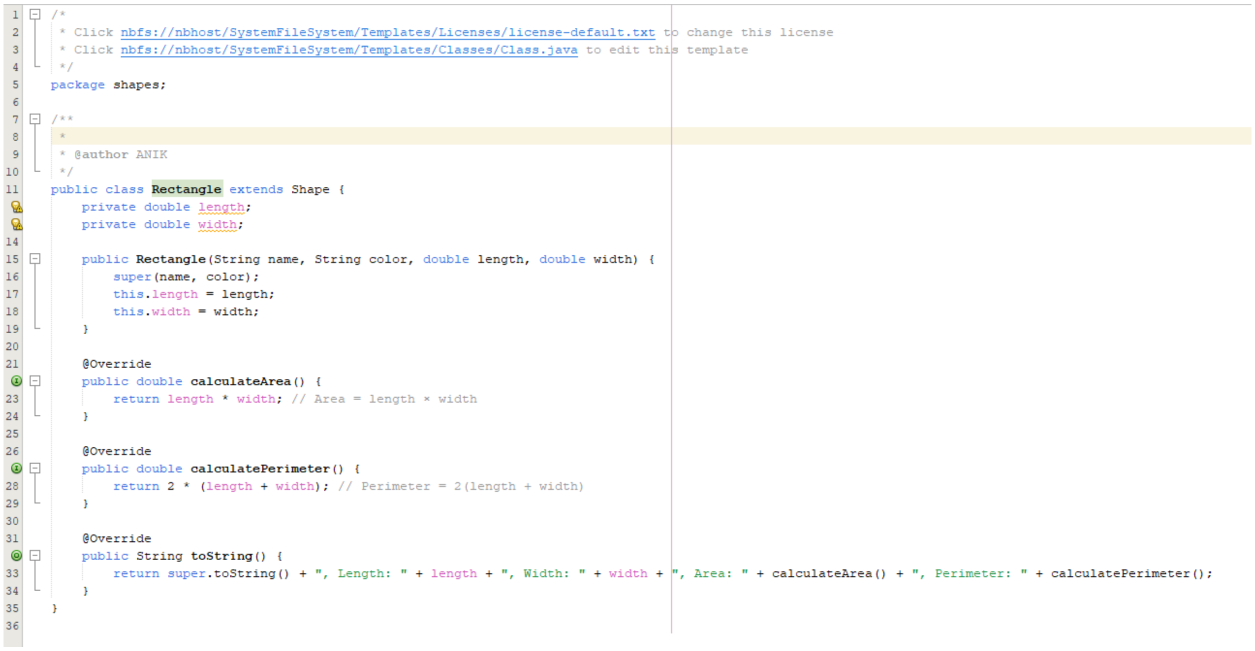
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package shapes;

/**
 *
 * @author ANIK
 */
public abstract class Shape {
    private String name;
    private String color;

    public Shape(String name, String color) {
        this.name = name;
        this.color = color;
    }

    public String getName() {
        return name;
    }

    public String getColor() {
        return color;
    }

    public abstract double calculateArea();

    public abstract double calculatePerimeter();

    @Override
    public String toString() {
        return "Shape: " + name + ", Color: " + color;
    }
}
```

Circle.java



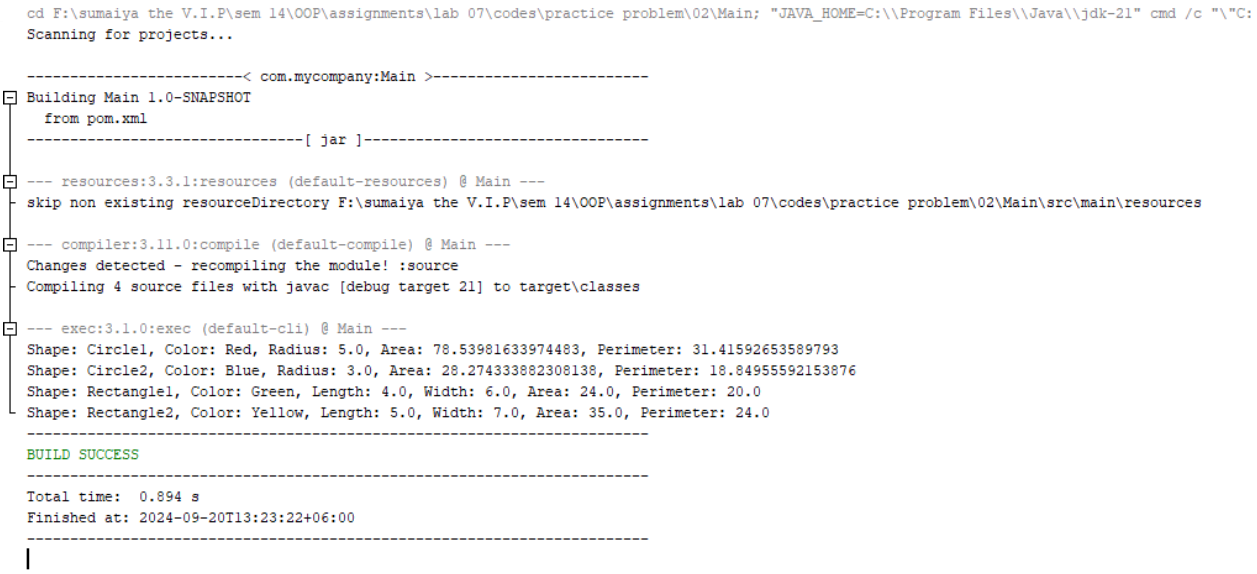
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package shapes;

/**
 *
 * @author ANIK
 */
public class Circle extends Shape {
    private double radius;

    public Circle(String name, String color, double radius) {
        super(name, color);
        this.radius = radius;
    }

    @Override
    public double calculateArea() {
        return Math.PI * radius * radius; // Area = πr²
    }

    @Override
    public double calculatePerimeter() {
        return 2 * Math.PI * radius; // Perimeter = 2πr
    }

    @Override
    public String toString() {
        return super.toString() + ", Radius: " + radius + ", Area: " + calculateArea() + ", Perimeter: " + calculatePerimeter();
    }
}
```

Rectangle.java



```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package shapes;

/**
 *
 * @author ANIK
 */
public class Rectangle extends Shape {
    private double length;
    private double width;

    public Rectangle(String name, String color, double length, double width) {
        super(name, color);
        this.length = length;
        this.width = width;
    }

    @Override
    public double calculateArea() {
        return length * width; // Area = length × width
    }

    @Override
    public double calculatePerimeter() {
        return 2 * (length + width); // Perimeter = 2(length + width)
    }

    @Override
    public String toString() {
        return super.toString() + ", Length: " + length + ", Width: " + width + ", Area: " + calculateArea() + ", Perimeter: " + calculatePerimeter();
    }
}
```

Output:



```
cd F:\sumaiya the V.I.P\sem 14\OOP\assignments\lab 07\codes\practice problem\02\Main; "JAVA_HOME=C:\\Program Files\\Java\\jdk-21" cmd /c "\"C:
Scanning for projects...

------------------------< com.mycompany:Main >------------------------
Building Main 1.0-SNAPSHOT
  from pom.xml
--------------------------------[ jar ]---------------------------------

--- resources:3.3.1:resources (default-resources) @ Main ---
skip non existing resourceDirectory F:\sumaiya the V.I.P\sem 14\OOP\assignments\lab 07\codes\practice problem\02\Main\src\main\resources

--- compiler:3.11.0:compile (default-compile) @ Main ---
Changes detected - recompiling the module! :source
Compiling 4 source files with javac [debug target 21] to target\classes

--- exec:3.1.0:exec (default-cli) @ Main ---
Shape: Circle1, Color: Red, Radius: 5.0, Area: 78.53981633974483, Perimeter: 31.41592653589793
Shape: Circle2, Color: Blue, Radius: 3.0, Area: 28.274333882308138, Perimeter: 18.84955592153876
Shape: Rectangle1, Color: Green, Length: 4.0, Width: 6.0, Area: 24.0, Perimeter: 20.0
Shape: Rectangle2, Color: Yellow, Length: 5.0, Width: 7.0, Area: 35.0, Perimeter: 24.0
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.894 s
Finished at: 2024-09-20T13:23:22+06:00
------------------------------------------------------------------------
```

Practice Problem 03

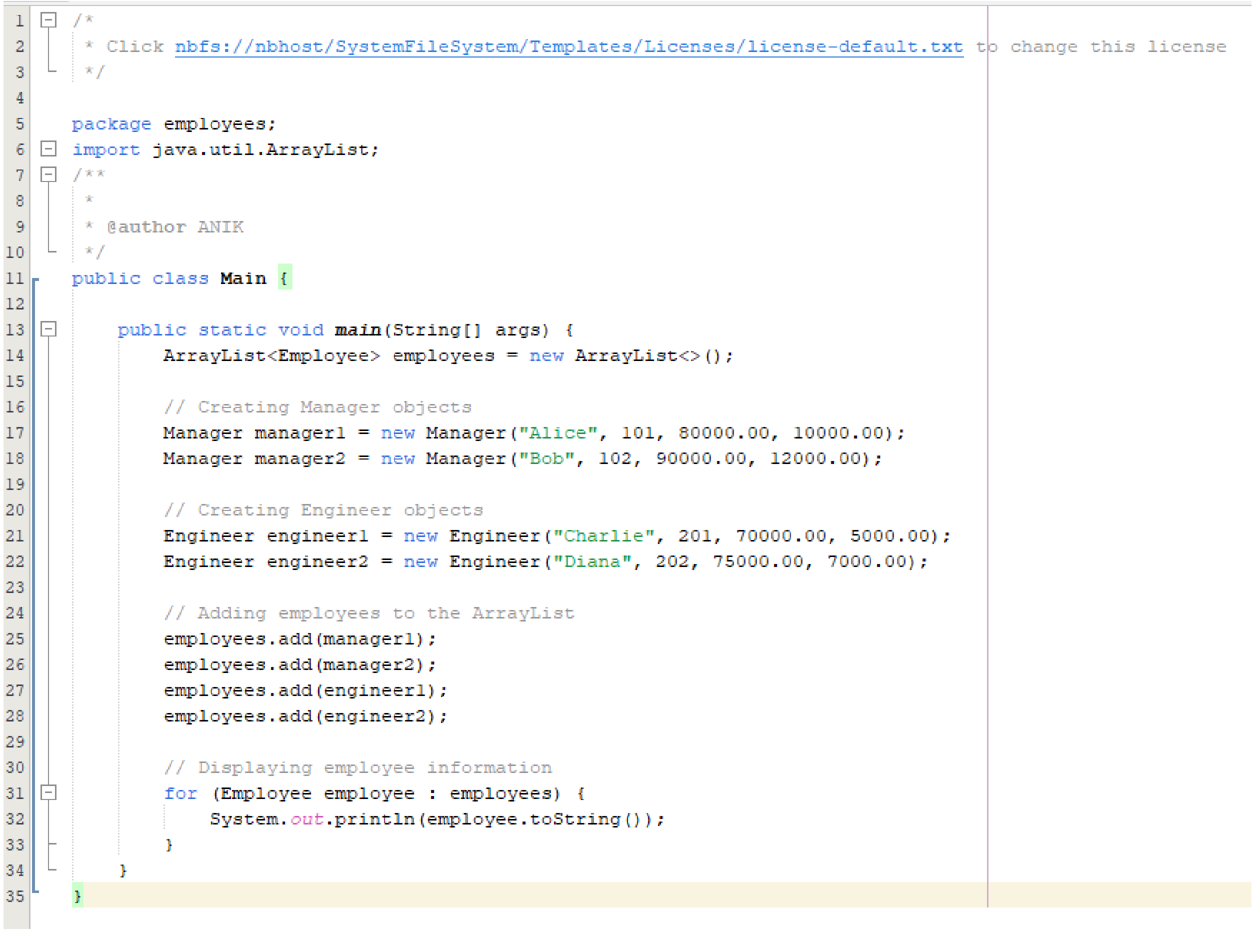
Code Explanation:

This Java program models employees using a base class called ‘Employee’, which includes common attributes like name, ID, and salary. Two subclasses, ‘Manager’ and ‘Engineer’, extend this base class, adding specific properties such as bonuses. Each subclass implements a method to calculate the total salary, including these bonuses. The main class creates instances of both types, stores them in an ‘ArrayList’, and displays their details, showcasing the benefits of inheritance and polymorphism in object-oriented programming.

Code Screenshot:

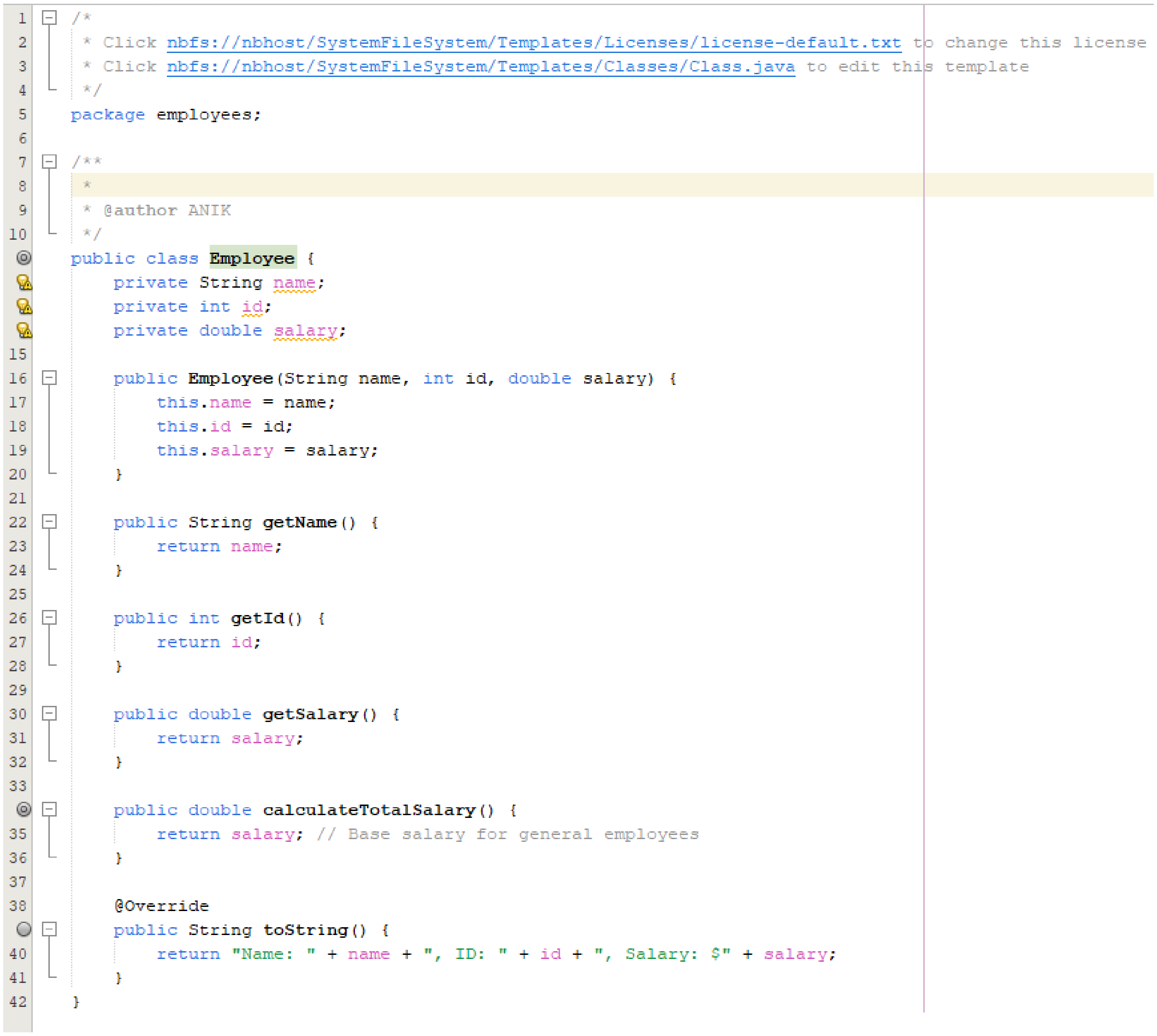
Input:

Main.java



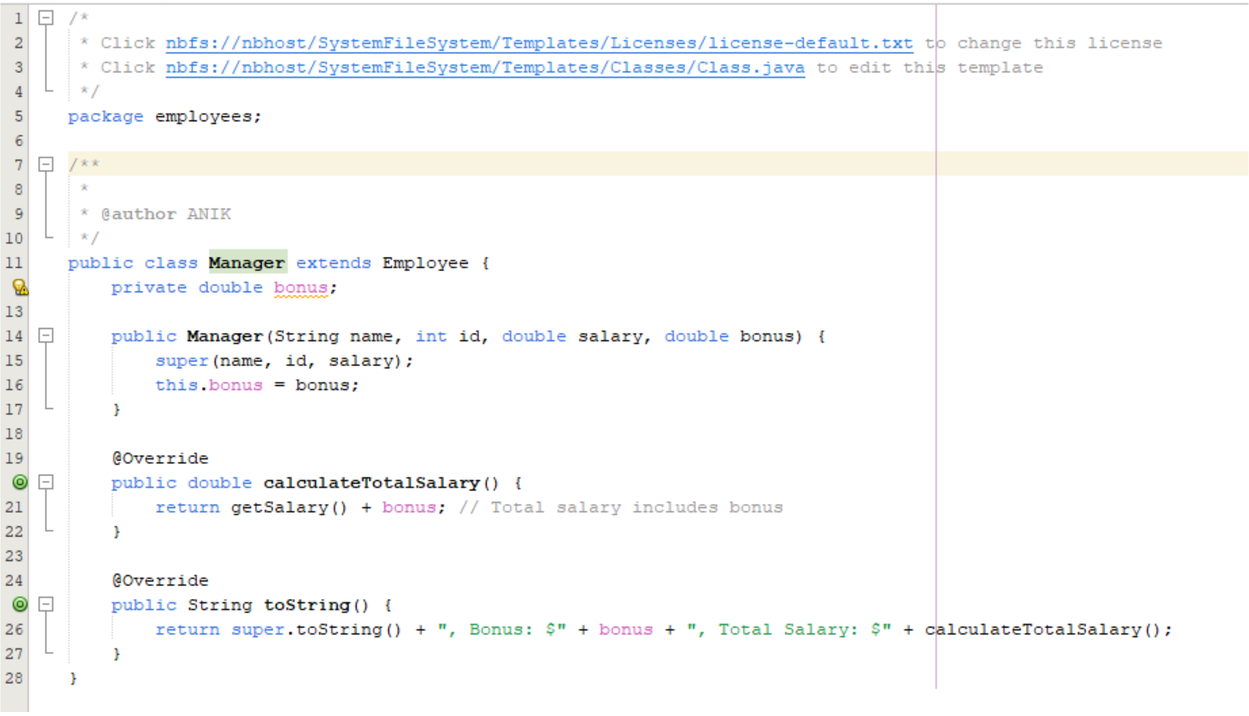
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */

package employees;
import java.util.ArrayList;
/**
 *
 * @author ANIK
 */
public class Main {

    public static void main(String[] args) {
        ArrayList<Employee> employees = new ArrayList<>();

        // Creating Manager objects
        Manager manager1 = new Manager("Alice", 101, 80000.00, 10000.00);
        Manager manager2 = new Manager("Bob", 102, 90000.00, 12000.00);

        // Creating Engineer objects
        Engineer engineer1 = new Engineer("Charlie", 201, 70000.00, 5000.00);
        Engineer engineer2 = new Engineer("Diana", 202, 75000.00, 7000.00);

        // Adding employees to the ArrayList
        employees.add(manager1);
        employees.add(manager2);
        employees.add(engineer1);
        employees.add(engineer2);

        // Displaying employee information
        for (Employee employee : employees) {
            System.out.println(employee.toString());
        }
    }
}
```

Employee.java



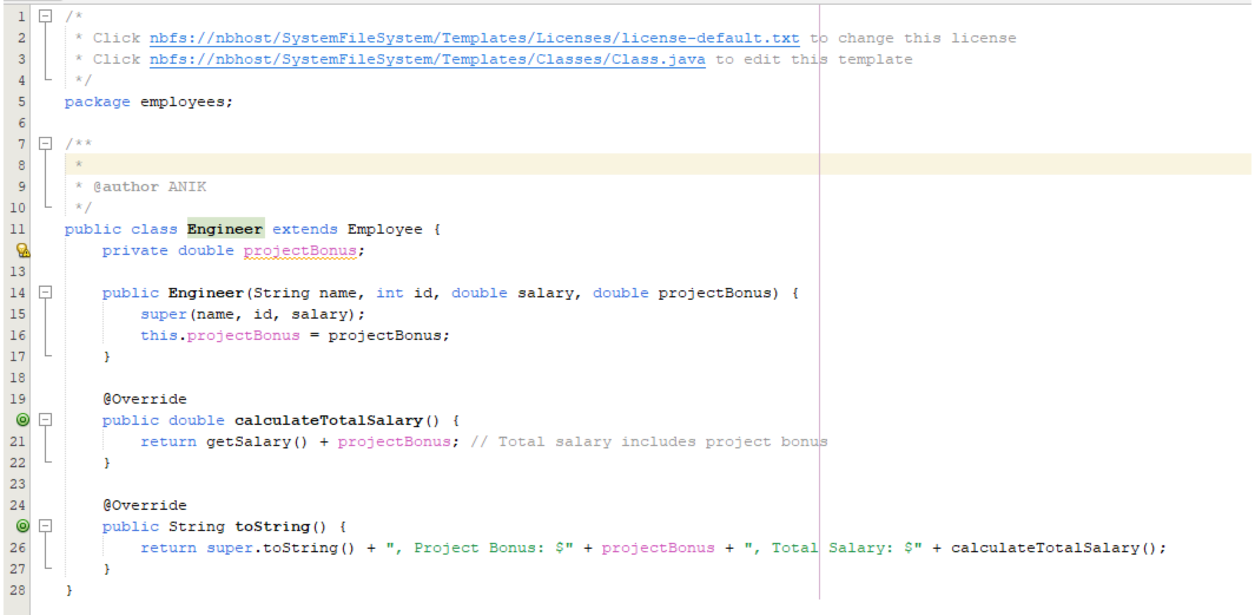
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package employees;

/**
 *
 * @author ANIK
 */
public class Employee {
    private String name;
    private int id;
    private double salary;

    public Employee(String name, int id, double salary) {
        this.name = name;
        this.id = id;
        this.salary = salary;
    }

    public String getName() {
        return name;
    }

    public int getId() {
        return id;
    }

    public double getSalary() {
        return salary;
    }

    public double calculateTotalSalary() {
        return salary; // Base salary for general employees
    }

    @Override
    public String toString() {
        return "Name: " + name + ", ID: " + id + ", Salary: $" + salary;
    }
}
```

Manager.java



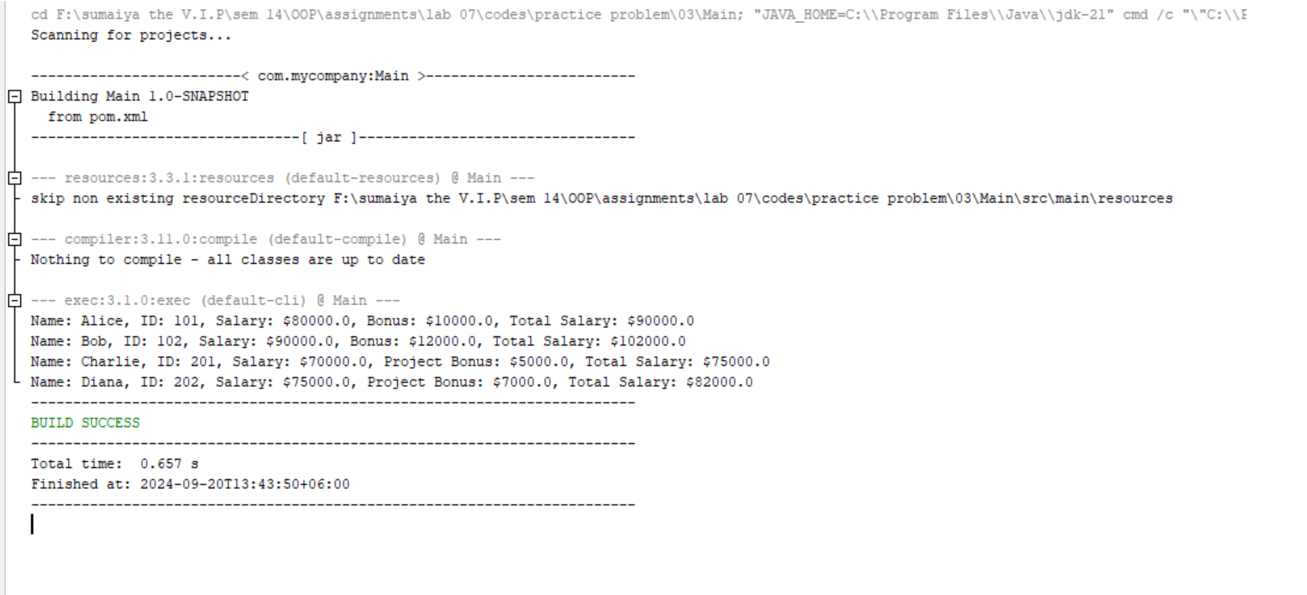
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package employees;

/**
 *
 * @author ANIK
 */
public class Manager extends Employee {
    private double bonus;

    public Manager(String name, int id, double salary, double bonus) {
        super(name, id, salary);
        this.bonus = bonus;
    }

    @Override
    public double calculateTotalSalary() {
        return getSalary() + bonus; // Total salary includes bonus
    }

    @Override
    public String toString() {
        return super.toString() + ", Bonus: $" + bonus + ", Total Salary: $" + calculateTotalSalary();
    }
}
```

Engineer.java



```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package employees;

/**
 *
 * @author ANIK
 */
public class Engineer extends Employee {
    private double projectBonus;

    public Engineer(String name, int id, double salary, double projectBonus) {
        super(name, id, salary);
        this.projectBonus = projectBonus;
    }

    @Override
    public double calculateTotalSalary() {
        return getSalary() + projectBonus; // Total salary includes project bonus
    }

    @Override
    public String toString() {
        return super.toString() + ", Project Bonus: $" + projectBonus + ", Total Salary: $" + calculateTotalSalary();
    }
}
```

Output:



```
cd F:\sumaiya the V.I.P\sem 14\OOP\assignments\lab 07\codes\practice problem\03\Main; "JAVA_HOME=C:\\Program Files\\Java\\jdk-21" cmd /c "\"C:\\
Scanning for projects...

------------------------< com.mycompany:Main >-------------------------
Building Main 1.0-SNAPSHOT
  from pom.xml
--------------------------------[ jar ]---------------------------------

--- resources:3.3.1:resources (default-resources) @ Main ---
skip non existing resourceDirectory F:\sumaiya the V.I.P\sem 14\OOP\assignments\lab 07\codes\practice problem\03\Main\src\main\resources

--- compiler:3.11.0:compile (default-compile) @ Main ---
Nothing to compile - all classes are up to date

--- exec:3.1.0:exec (default-cli) @ Main ---
Name: Alice, ID: 101, Salary: $80000.0, Bonus: $10000.0, Total Salary: $90000.0
Name: Bob, ID: 102, Salary: $90000.0, Bonus: $12000.0, Total Salary: $102000.0
Name: Charlie, ID: 201, Salary: $70000.0, Project Bonus: $5000.0, Total Salary: $75000.0
Name: Diana, ID: 202, Salary: $75000.0, Project Bonus: $7000.0, Total Salary: $82000.0
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.657 s
Finished at: 2024-09-20T13:43:50+06:00
------------------------------------------------------------------------
```

Practice Problem 04

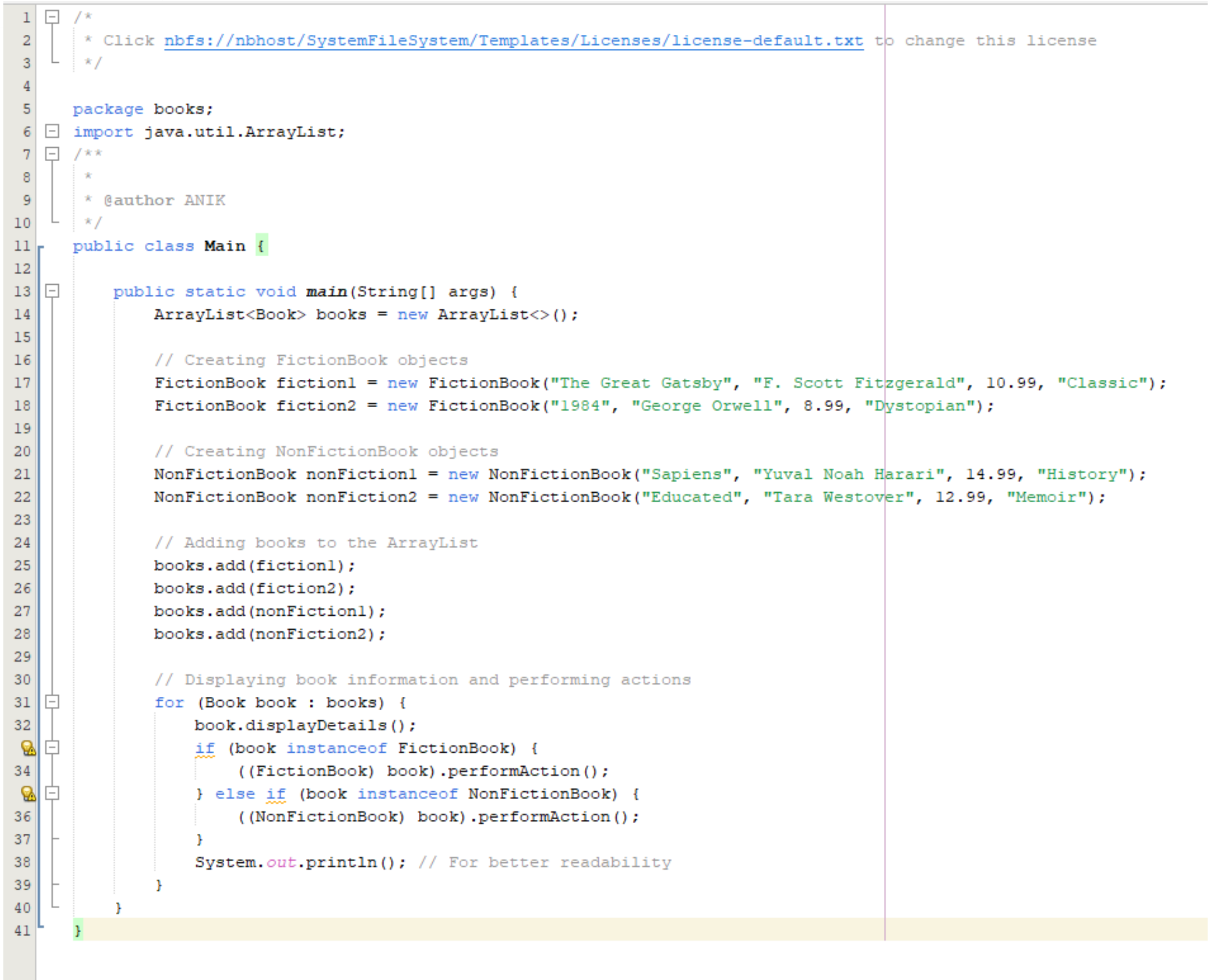
Code Explanation:

This Java program models books using a base class called 'Book', which includes common attributes like title, author, and price. Two subclasses, 'FictionBook' and 'NonFictionBook', extend this base class, adding specific properties such as genre and subject. Each subclass has methods to display book details and perform actions related to their type. The main class creates instances of these books, stores them in an 'ArrayList', and showcases their information and unique actions.

Code Screenshot:

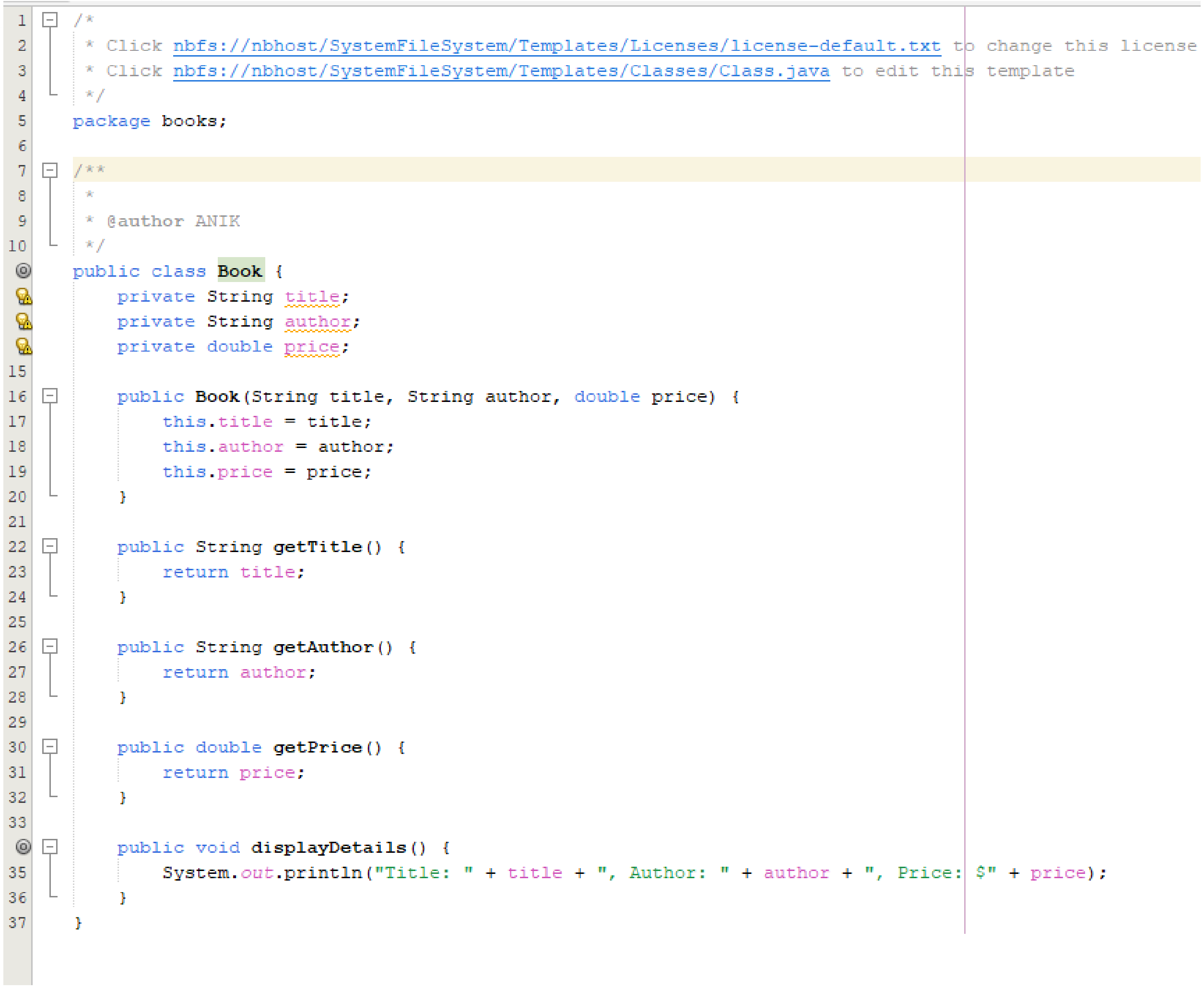
Input:

Main.java



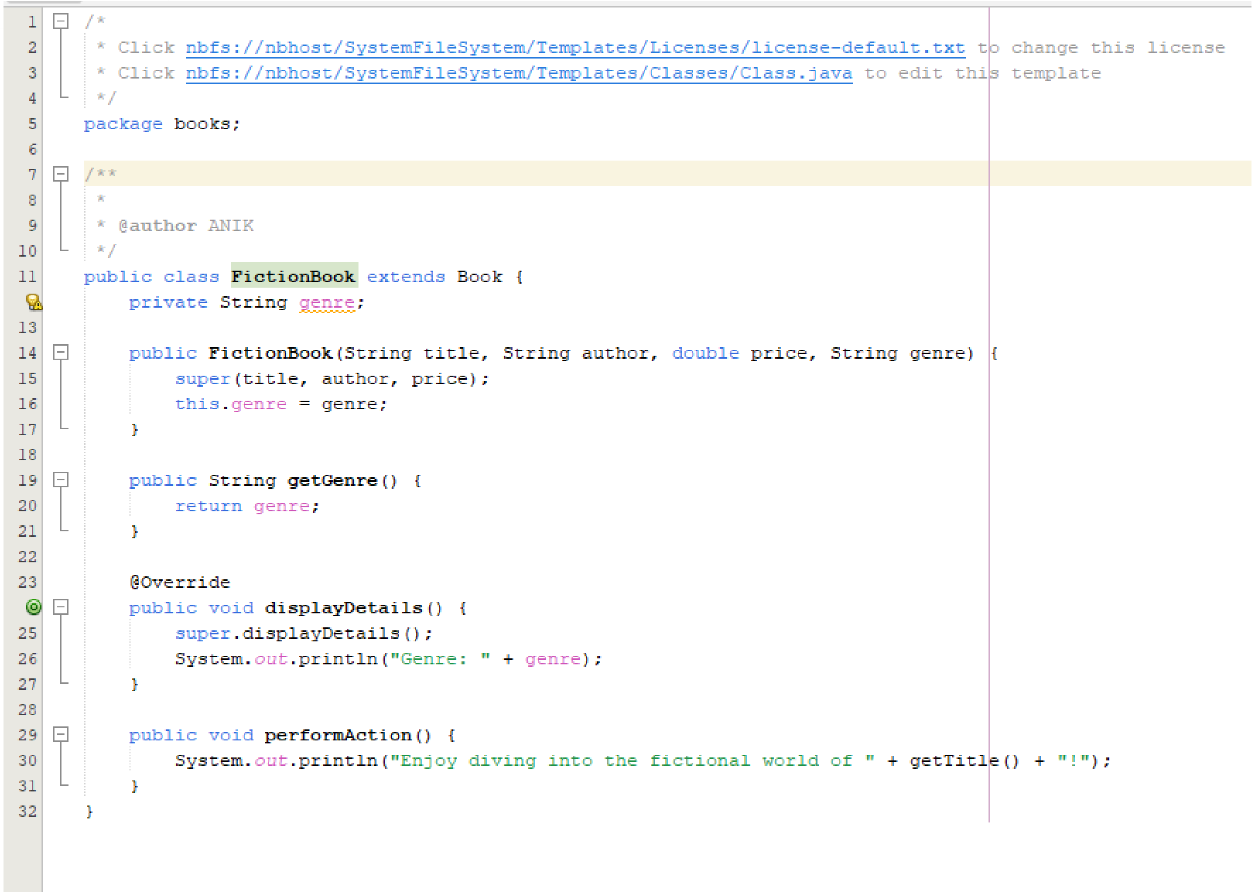
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */

package books;
import java.util.ArrayList;
/**
 *
 * @author ANIK
 */
public class Main {

    public static void main(String[] args) {
        ArrayList<Book> books = new ArrayList<>();

        // Creating FictionBook objects
        FictionBook fiction1 = new FictionBook("The Great Gatsby", "F. Scott Fitzgerald", 10.99, "Classic");
        FictionBook fiction2 = new FictionBook("1984", "George Orwell", 8.99, "Dystopian");

        // Creating NonFictionBook objects
        NonFictionBook nonFiction1 = new NonFictionBook("Sapiens", "Yuval Noah Harari", 14.99, "History");
        NonFictionBook nonFiction2 = new NonFictionBook("Educated", "Tara Westover", 12.99, "Memoir");

        // Adding books to the ArrayList
        books.add(fiction1);
        books.add(fiction2);
        books.add(nonFiction1);
        books.add(nonFiction2);

        // Displaying book information and performing actions
        for (Book book : books) {
            book.displayDetails();
            if (book instanceof FictionBook) {
                ((FictionBook) book).performAction();
            } else if (book instanceof NonFictionBook) {
                ((NonFictionBook) book).performAction();
            }
            System.out.println(); // For better readability
        }
    }
}
```

Book.java



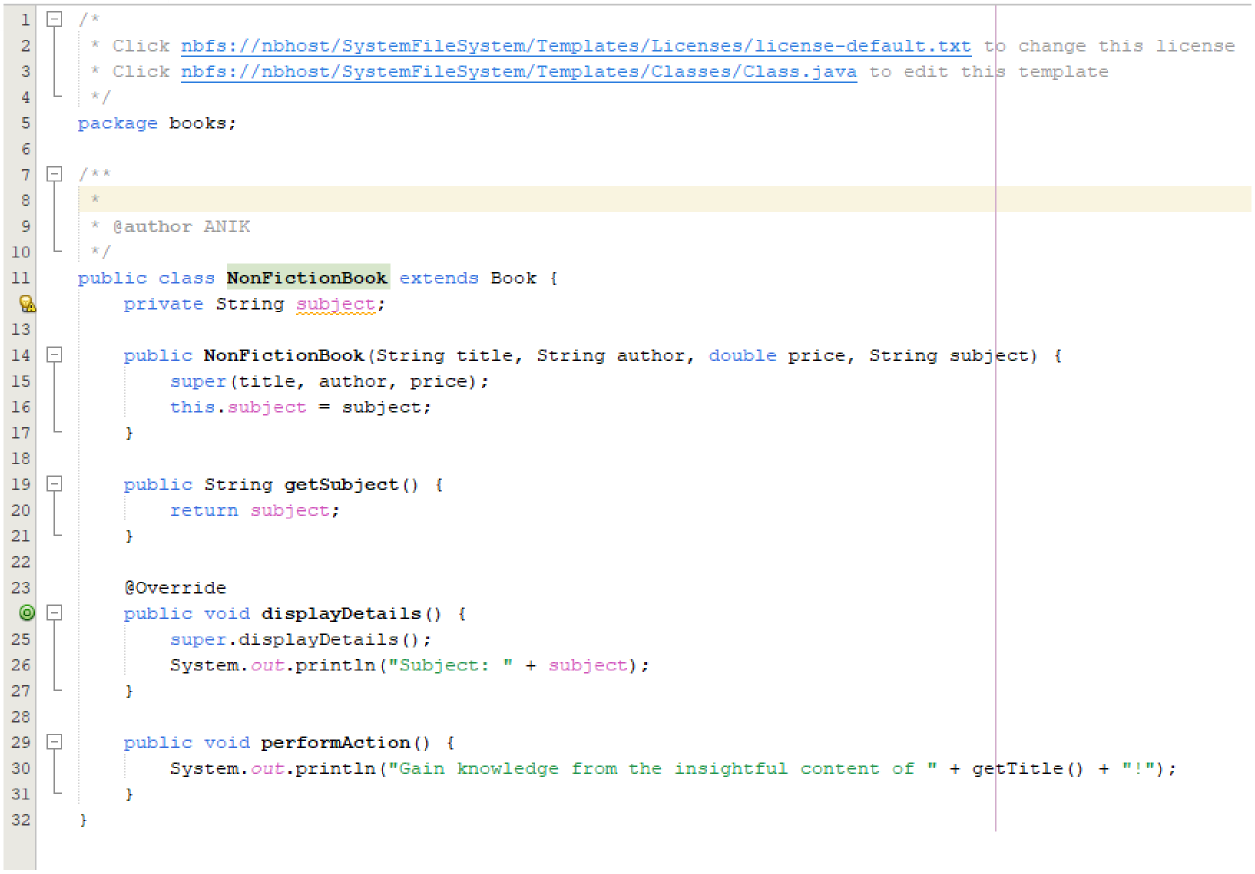
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package books;

/**
 *
 * @author ANIK
 */
public class Book {
    private String title;
    private String author;
    private double price;

    public Book(String title, String author, double price) {
        this.title = title;
        this.author = author;
        this.price = price;
    }

    public String getTitle() {
        return title;
    }

    public String getAuthor() {
        return author;
    }

    public double getPrice() {
        return price;
    }

    public void displayDetails() {
        System.out.println("Title: " + title + ", Author: " + author + ", Price: $" + price);
    }
}
```

FictionBook.java



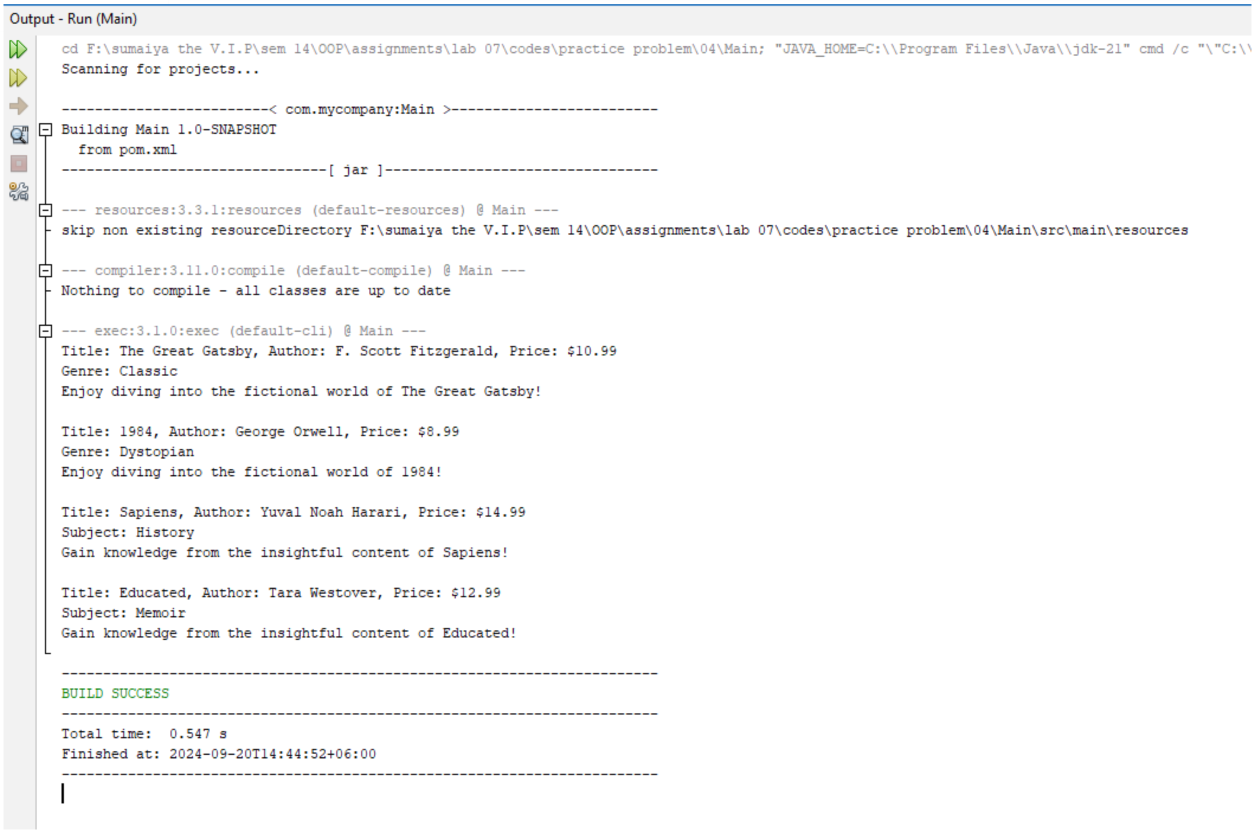
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package books;

/**
 *
 * @author ANIK
 */
public class FictionBook extends Book {
    private String genre;

    public FictionBook(String title, String author, double price, String genre) {
        super(title, author, price);
        this.genre = genre;
    }

    public String getGenre() {
        return genre;
    }

    @Override
    public void displayDetails() {
        super.displayDetails();
        System.out.println("Genre: " + genre);
    }

    public void performAction() {
        System.out.println("Enjoy diving into the fictional world of " + getTitle() + "!");
    }
}
```

NonFictionBook.java



```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package books;

/**
 *
 * @author ANIK
 */
public class NonFictionBook extends Book {
    private String subject;

    public NonFictionBook(String title, String author, double price, String subject) {
        super(title, author, price);
        this.subject = subject;
    }

    public String getSubject() {
        return subject;
    }

    @Override
    public void displayDetails() {
        super.displayDetails();
        System.out.println("Subject: " + subject);
    }

    public void performAction() {
        System.out.println("Gain knowledge from the insightful content of " + getTitle() + "!");
    }
}
```

Output:



Output - Run (Main)

```
cd F:\sumaiya the V.I.P\sem 14\OOP\assignments\lab 07\codes\practice problem\04\Main; "JAVA_HOME=C:\\Program Files\\Java\\jdk-21" cmd /c "\"C:\\
Scanning for projects...

------------------------< com.mycompany:Main >------------------------
Building Main 1.0-SNAPSHOT
  from pom.xml
--------------------------------[ jar ]---------------------------------

--- resources:3.3.1:resources (default-resources) @ Main ---
skip non existing resourceDirectory F:\sumaiya the V.I.P\sem 14\OOP\assignments\lab 07\codes\practice problem\04\Main\src\main\resources

--- compiler:3.11.0:compile (default-compile) @ Main ---
Nothing to compile - all classes are up to date

--- exec:3.1.0:exec (default-cli) @ Main ---
Title: The Great Gatsby, Author: F. Scott Fitzgerald, Price: $10.99
Genre: Classic
Enjoy diving into the fictional world of The Great Gatsby!

Title: 1984, Author: George Orwell, Price: $8.99
Genre: Dystopian
Enjoy diving into the fictional world of 1984!

Title: Sapiens, Author: Yuval Noah Harari, Price: $14.99
Subject: History
Gain knowledge from the insightful content of Sapiens!

Title: Educated, Author: Tara Westover, Price: $12.99
Subject: Memoir
Gain knowledge from the insightful content of Educated!

------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.547 s
Finished at: 2024-09-20T14:44:52+06:00
------------------------------------------------------------------------
```