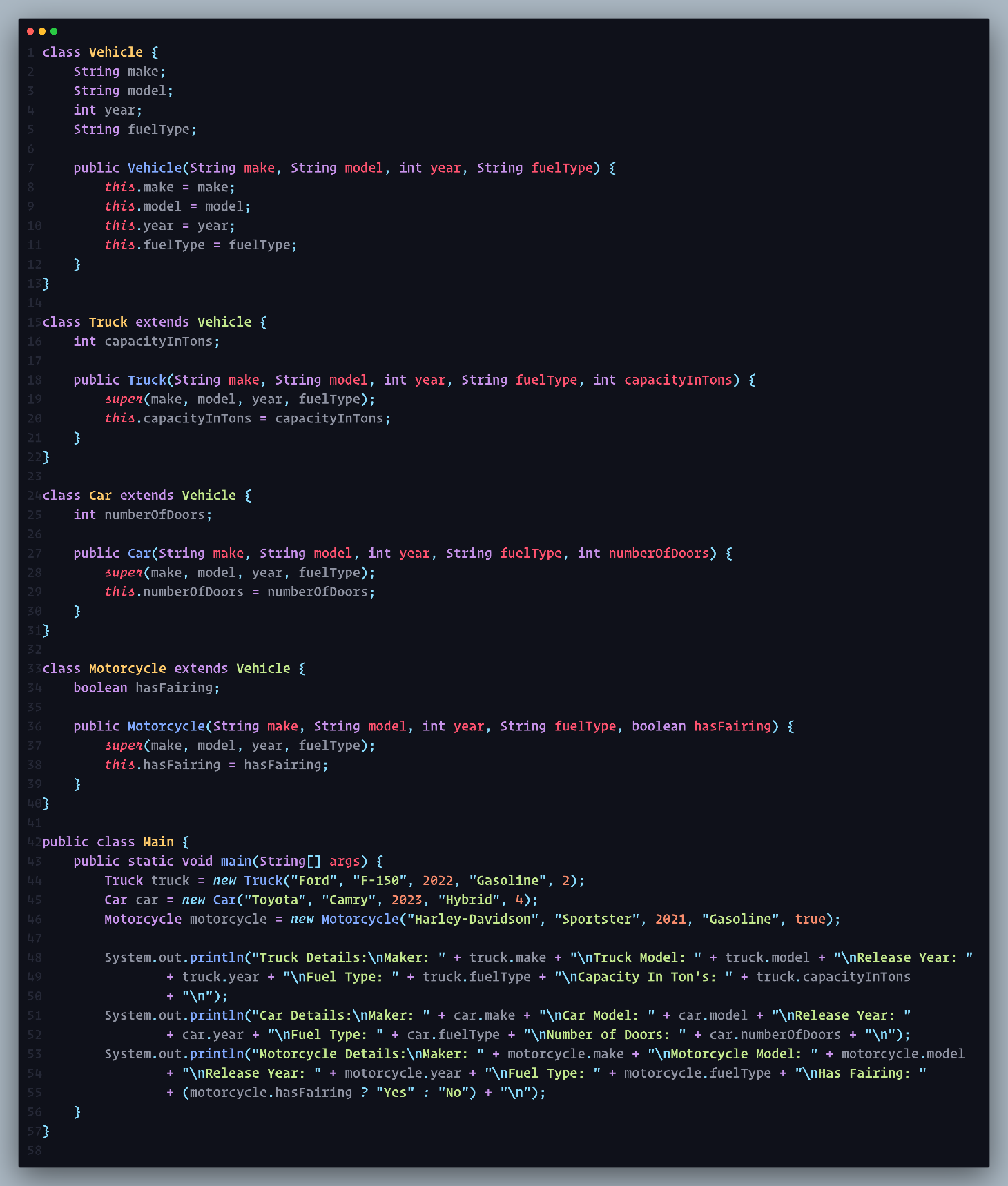
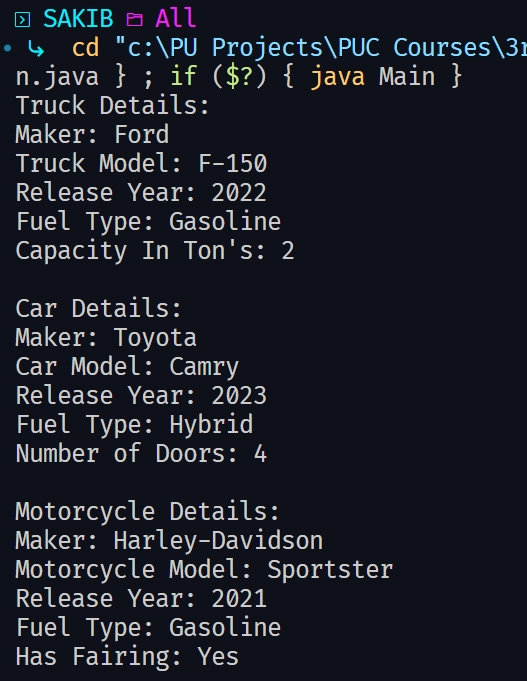
**Problem Statement :** Write a Java program to create a vehicle class hierarchy. The base class should be Vehicle, with subclasses Truck, Car and Motorcycle. Each subclass should have properties such as make, model, year, and fuel type.

**Source Code :**

****

**Output :**

****

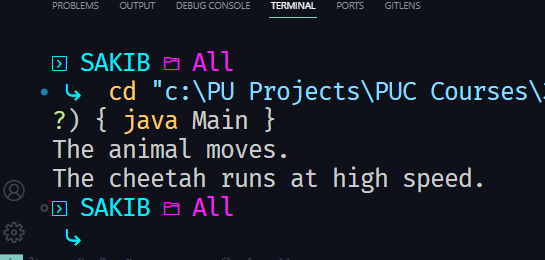
**Problem Statement :** Write a Java program to create a class called Animal with a

method named move (). Create a subclass called Cheetah that overrides the move () method to run.

**Source Code :**

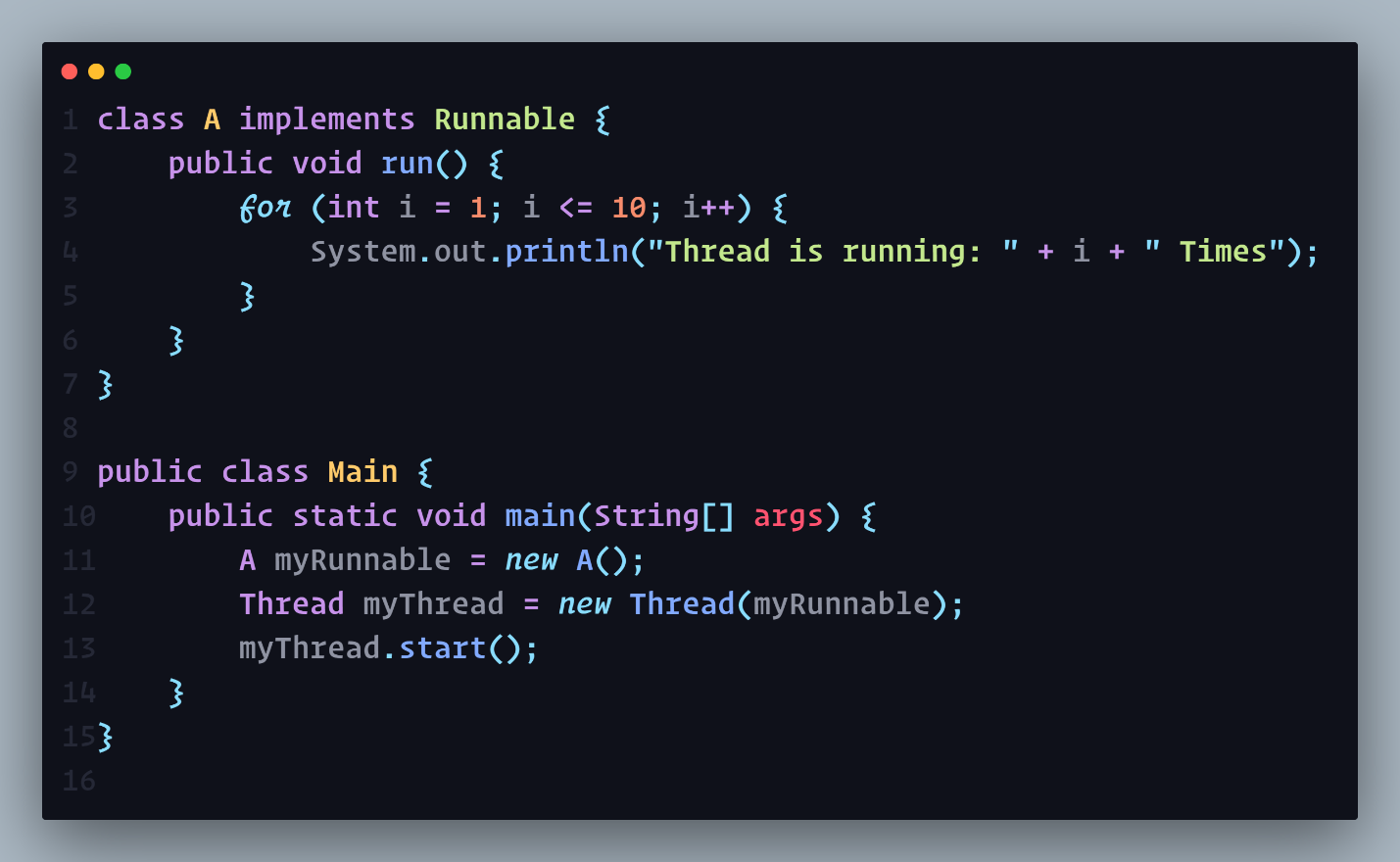
****

**Output :**

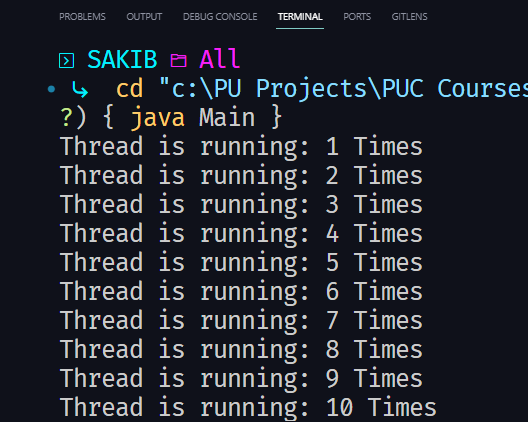
****

**Problem Statement :** Write a Java Program to Create a Basic Java Thread(by implementing Runnable Interface) that prints from number 1 to 10.

**Source Code :**



**Output:**

****

**Problem Statement :**

You are designing a simple program to calculate and display the areas of different shapes. Implement the necessary classes and interfaces to achieve this functionality using Java interfaces.

Create an interface named Shape with the following method:

double calculateArea(): This method calculates and returns the area of the shape.

Create two classes that implement the Shape interface:

Circle: This class should have an additional attribute: radius (double): The radius of the circle.

Rectangle: This class should have two additional attributes: width (double): The width of the rectangle.

height (double): The height of the rectangle.

Implement the calculateArea() method in both the Circle and Rectangle classes to

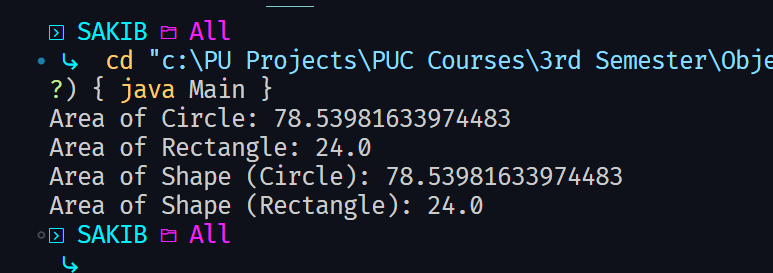
calculate and return the respective areas.

Define the Shape interface and implement it in the Circle and Rectangle classes. Demonstrate the usage of interfaces by creating instances of both shapes and calculating their areas. Ensure that your implementation highlights the concept of interfaces, method implementation, and polymorphism in Java.

**Source Code :**

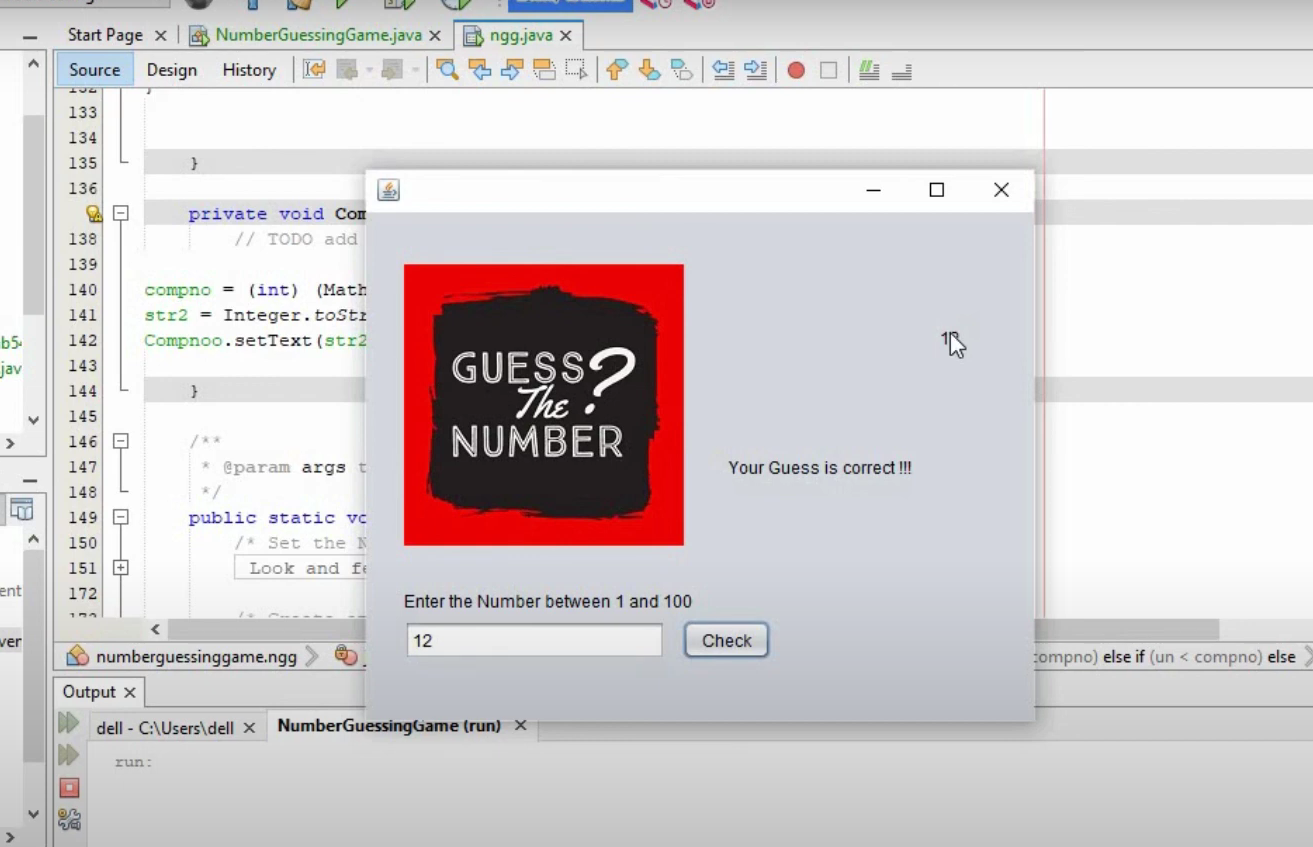
****

**Output :**

****

**Problem Statement :** Create a Number Guessing Game in Java.

**Source Code & Output :**

****