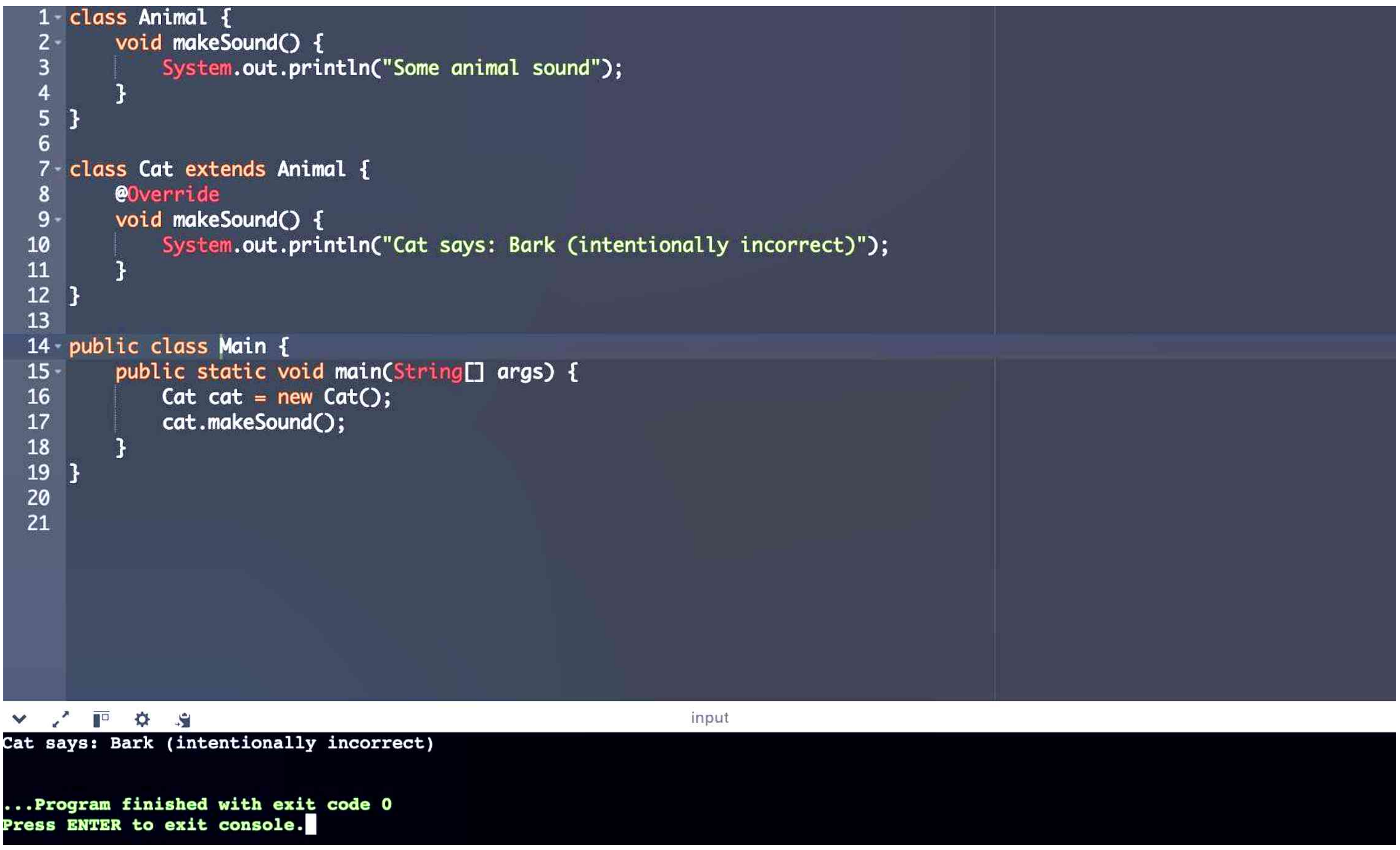
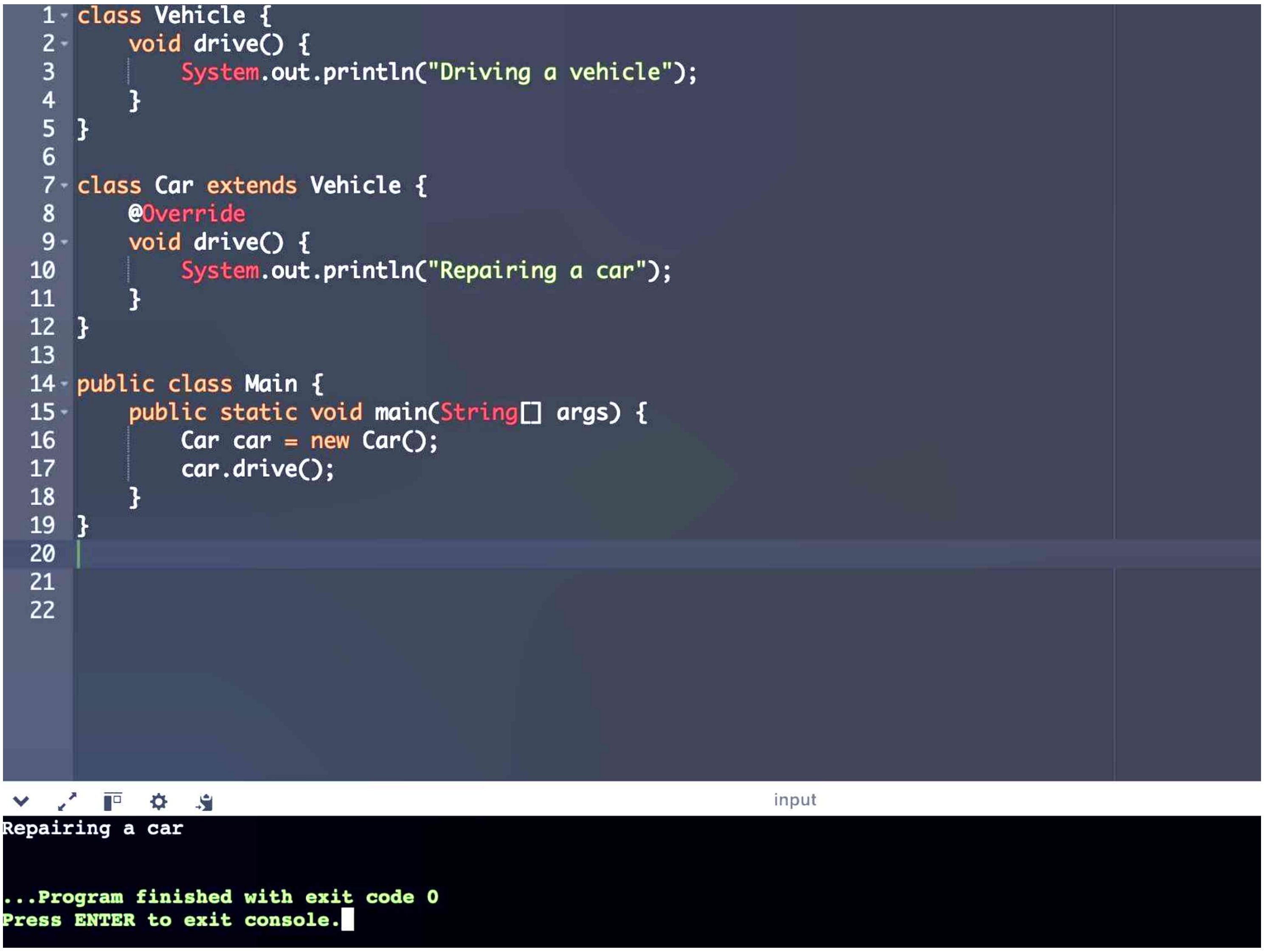
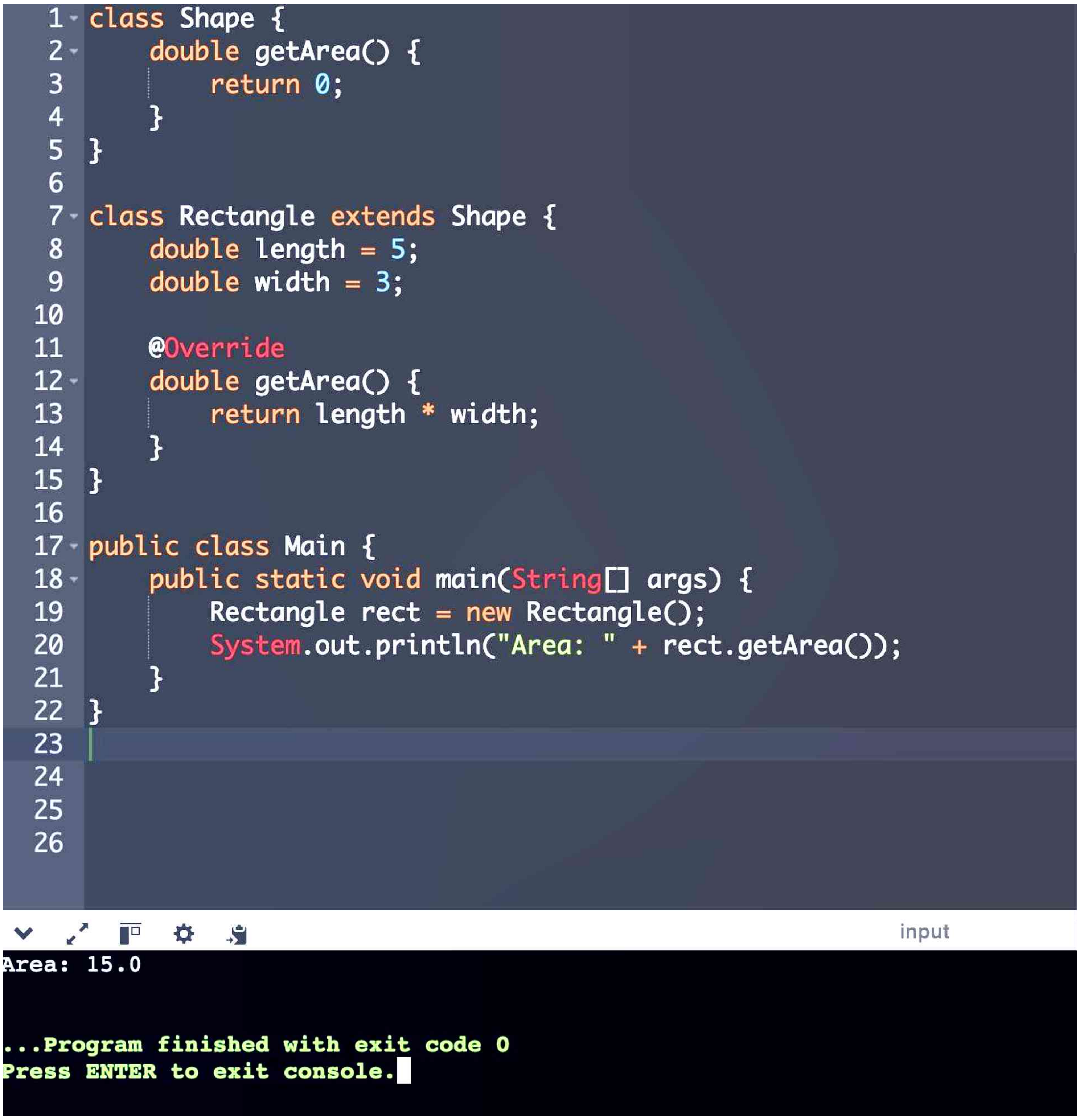

```java
class Animal {
    void makeSound() {
        System.out.println("Some animal sound");
    }
}

class Cat extends Animal {
    @Override
    void makeSound() {
        System.out.println("Cat says: Bark (intentionally incorrect)");
    }
}

public class Main {
    public static void main(String[] args) {
        Cat cat = new Cat();
        cat.makeSound();
    }
}
```

```
Cat says: Bark (intentionally incorrect)


...Program finished with exit code 0
Press ENTER to exit console.
```



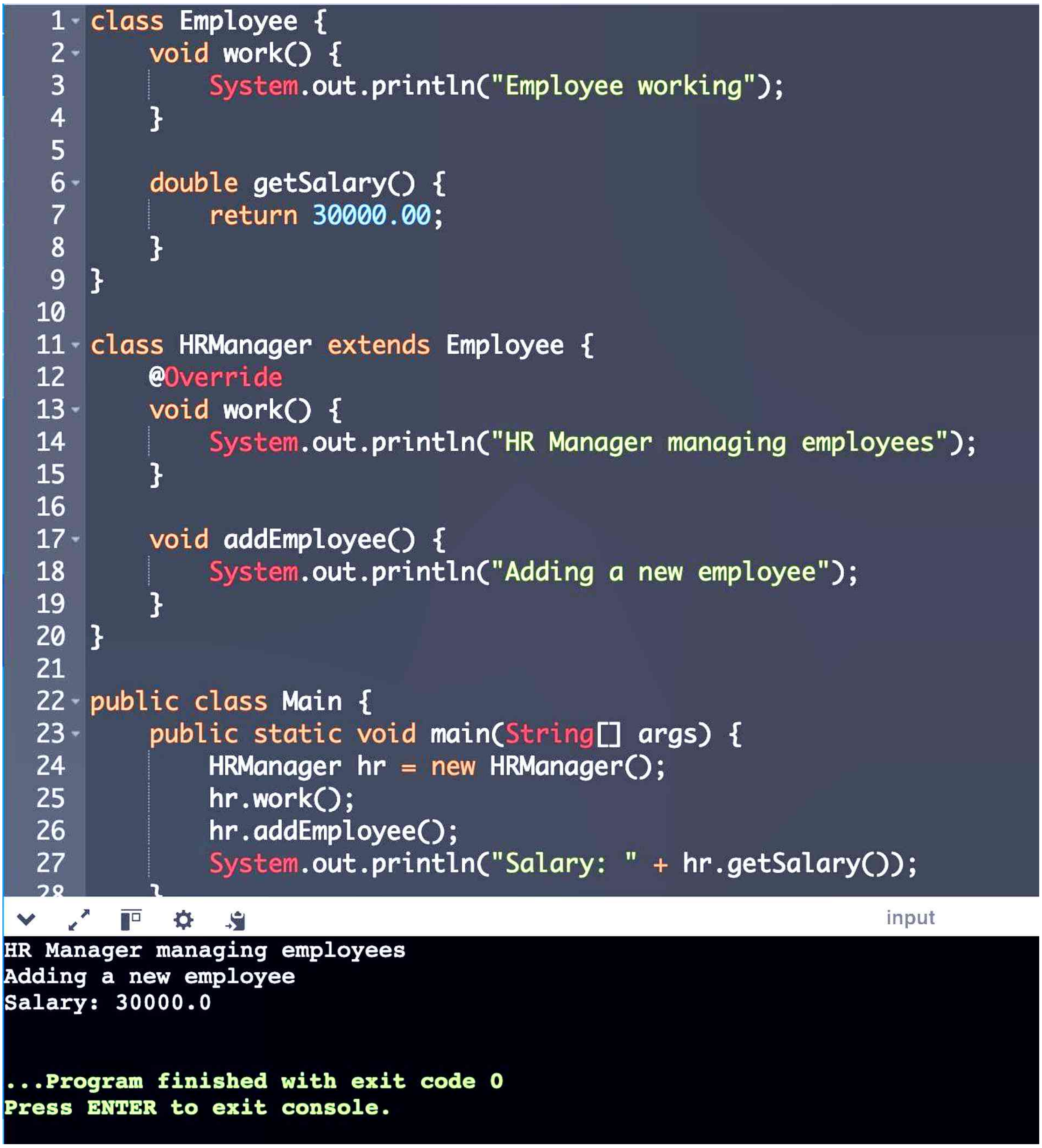
```java
class Vehicle {
    void drive() {
        System.out.println("Driving a vehicle");
    }
}

class Car extends Vehicle {
    @Override
    void drive() {
        System.out.println("Repairing a car");
    }
}

public class Main {
    public static void main(String[] args) {
        Car car = new Car();
        car.drive();
    }
}
```

```
Repairing a car


...Program finished with exit code 0
Press ENTER to exit console.
```



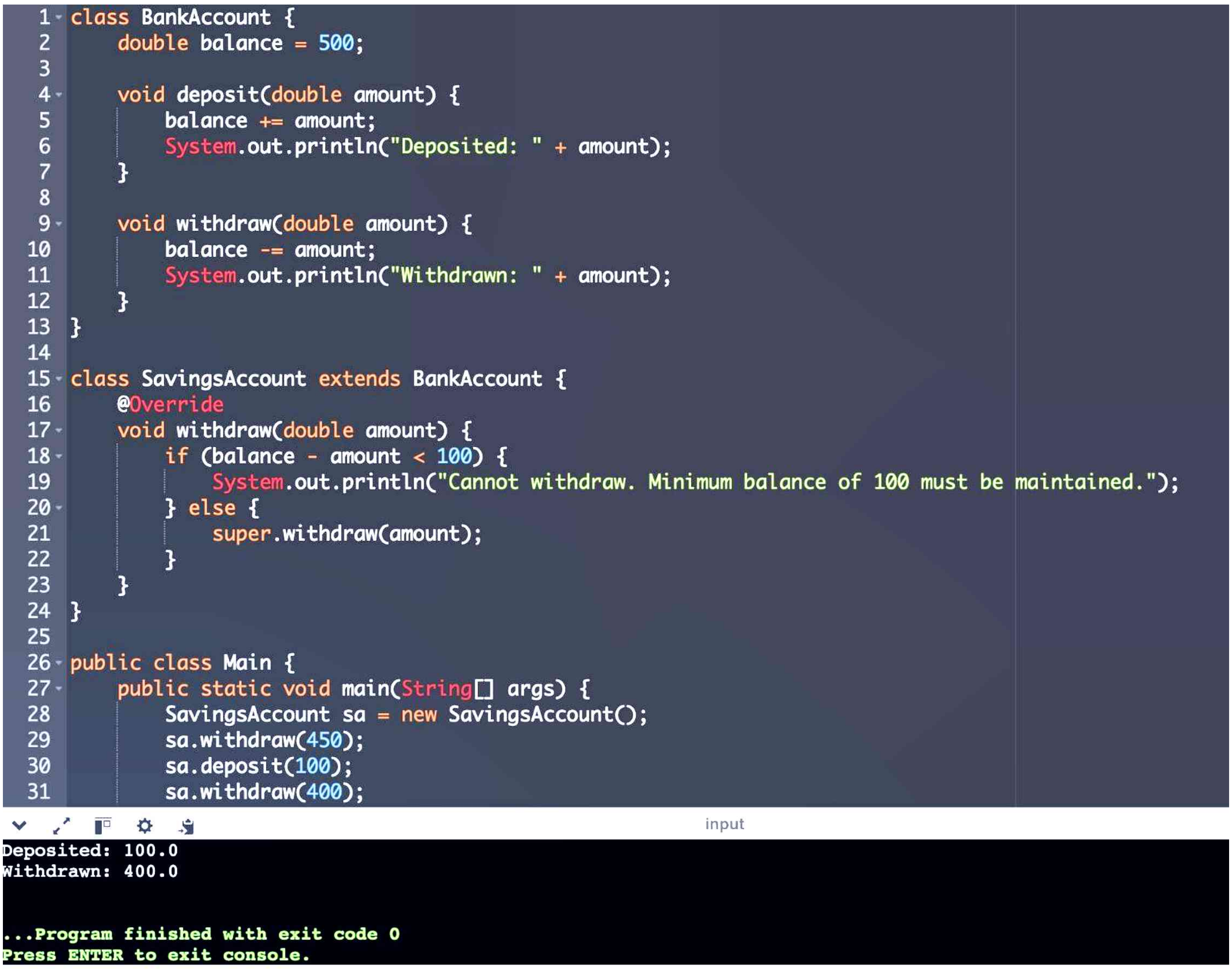
```java
class Shape {
    double getArea() {
        return 0;
    }
}

class Rectangle extends Shape {
    double length = 5;
    double width = 3;

    @Override
    double getArea() {
        return length * width;
    }
}

public class Main {
    public static void main(String[] args) {
        Rectangle rect = new Rectangle();
        System.out.println("Area:  " + rect.getArea());
    }
}
```

```
Area: 15.0


...Program finished with exit code 0
Press ENTER to exit console.
```



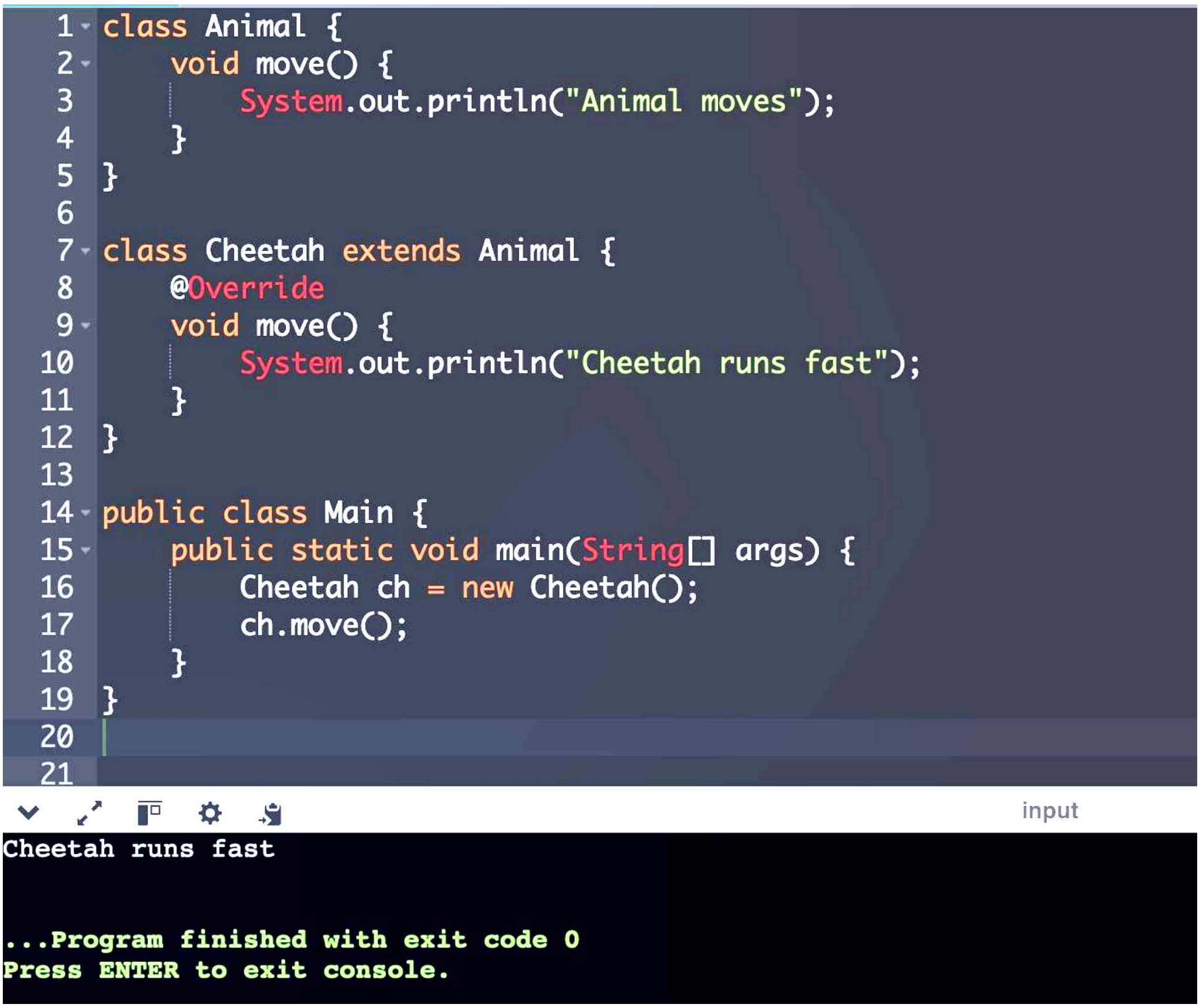
```java
class Employee {
    void work() {
        System.out.println("Employee working");
    }

    double getSalary() {
        return 30000.00;
    }
}

class HRManager extends Employee {
    @Override
    void work() {
        System.out.println("HR Manager managing employees");
    }

    void addEmployee() {
        System.out.println("Adding a new employee");
    }
}

public class Main {
    public static void main(String[] args) {
        HRManager hr = new HRManager();
        hr.work();
        hr.addEmployee();
        System.out.println("Salary: " + hr.getSalary());
```

```
HR Manager managing employees
Adding a new employee
Salary: 30000.0


...Program finished with exit code 0
Press ENTER to exit console.
```



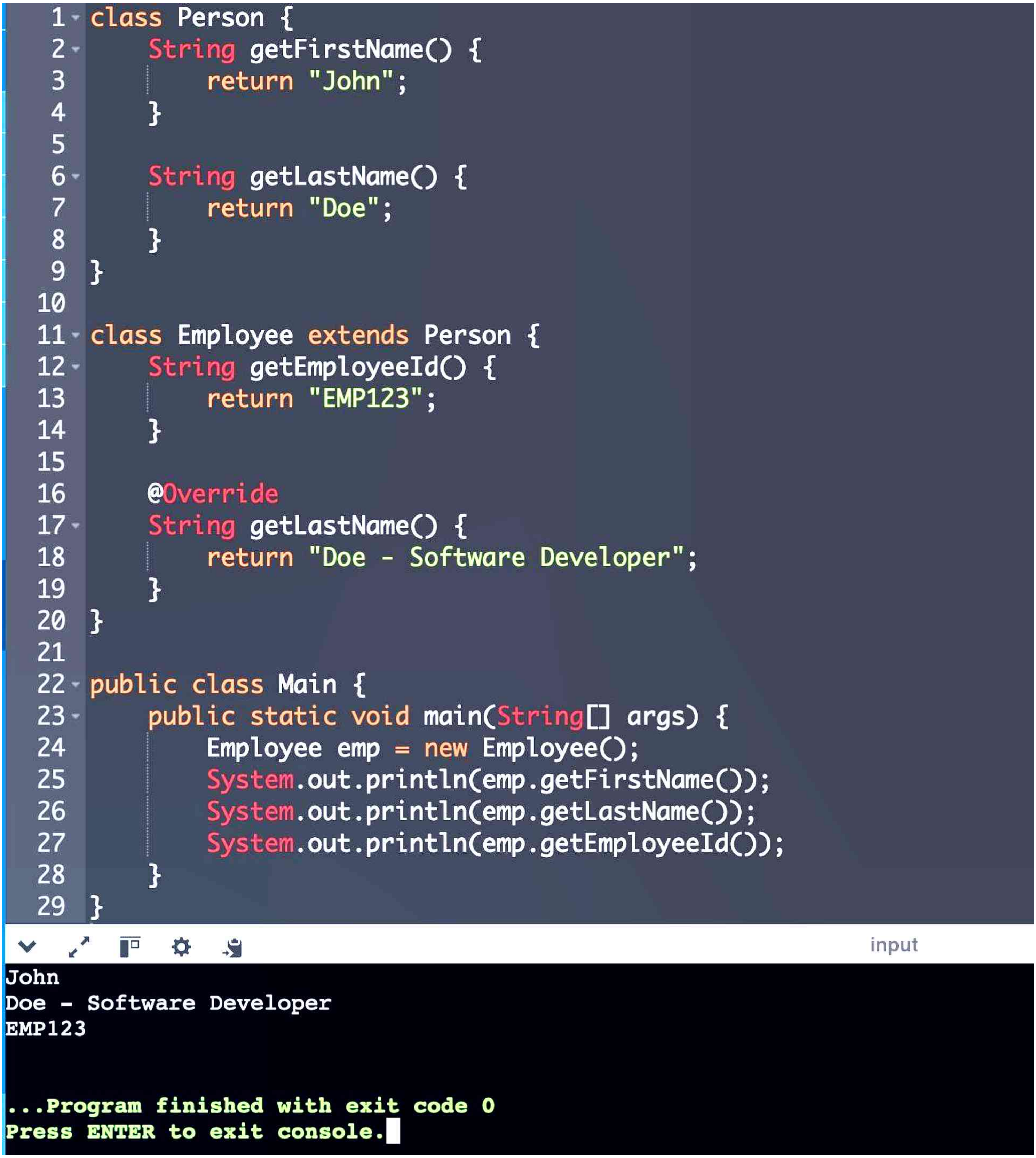
```java
class BankAccount {
    double balance = 500;

    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: " + amount);
    }

    void withdraw(double amount) {
        balance -= amount;
        System.out.println("Withdrawn: " + amount);
    }
}

class SavingsAccount extends BankAccount {
    @Override
    void withdraw(double amount) {
        if (balance - amount < 100) {
            System.out.println("Cannot withdraw. Minimum balance of 100 must be maintained.");
        } else {
            super.withdraw(amount);
        }
    }
}

public class Main {
    public static void main(String[] args) {
        SavingsAccount sa = new SavingsAccount();
        sa.withdraw(450);
        sa.deposit(100);
        sa.withdraw(400);
```

```
Deposited: 100.0
Withdrawn: 400.0


...Program finished with exit code 0
Press ENTER to exit console.
```



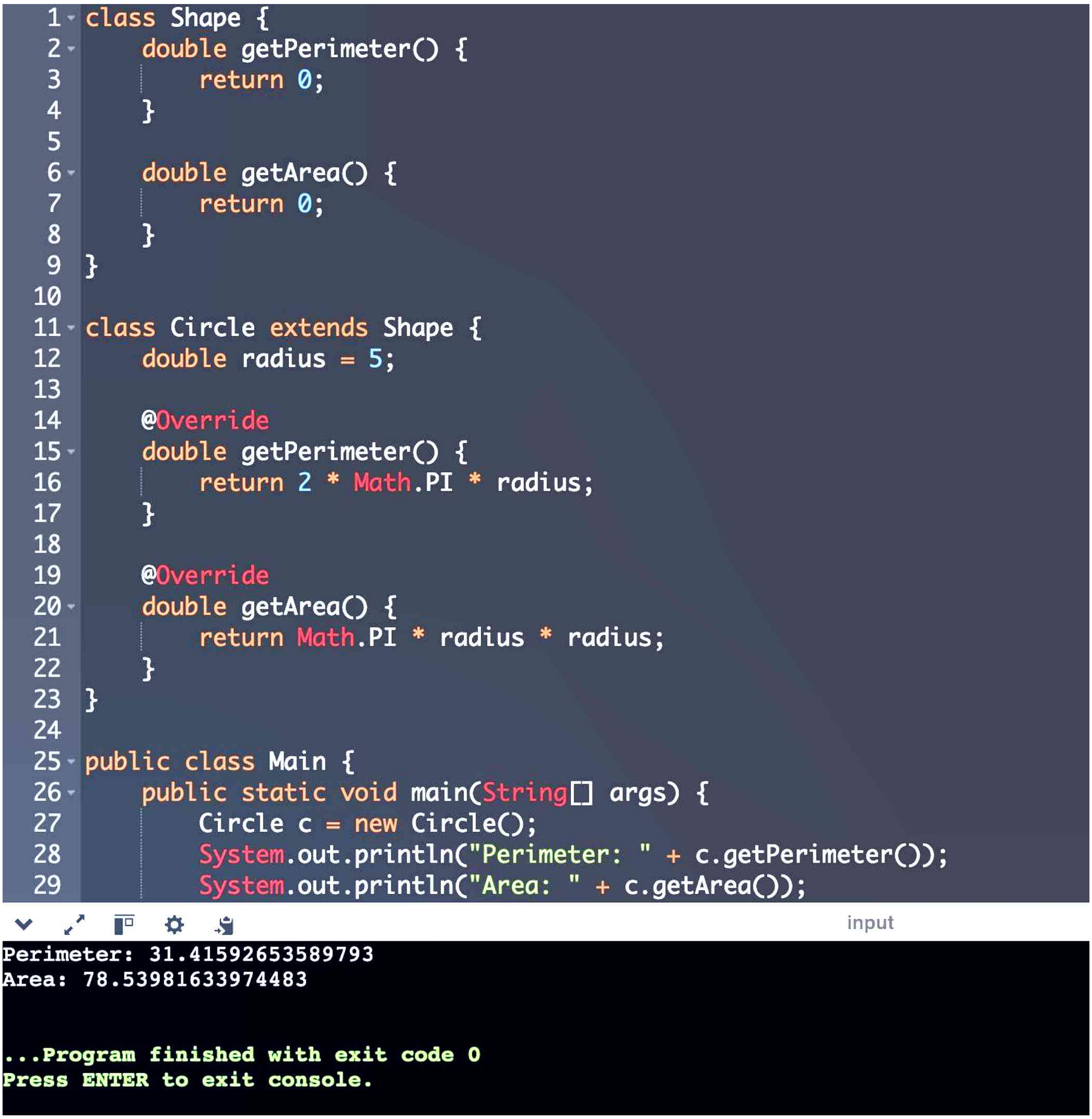
```java
class Animal {
    void move() {
        System.out.println("Animal moves");
    }
}

class Cheetah extends Animal {
    @Override
    void move() {
        System.out.println("Cheetah runs fast");
    }
}

public class Main {
    public static void main(String[] args) {
        Cheetah ch = new Cheetah();
        ch.move();
    }
}
```

```
Cheetah runs fast


...Program finished with exit code 0
Press ENTER to exit console.
```



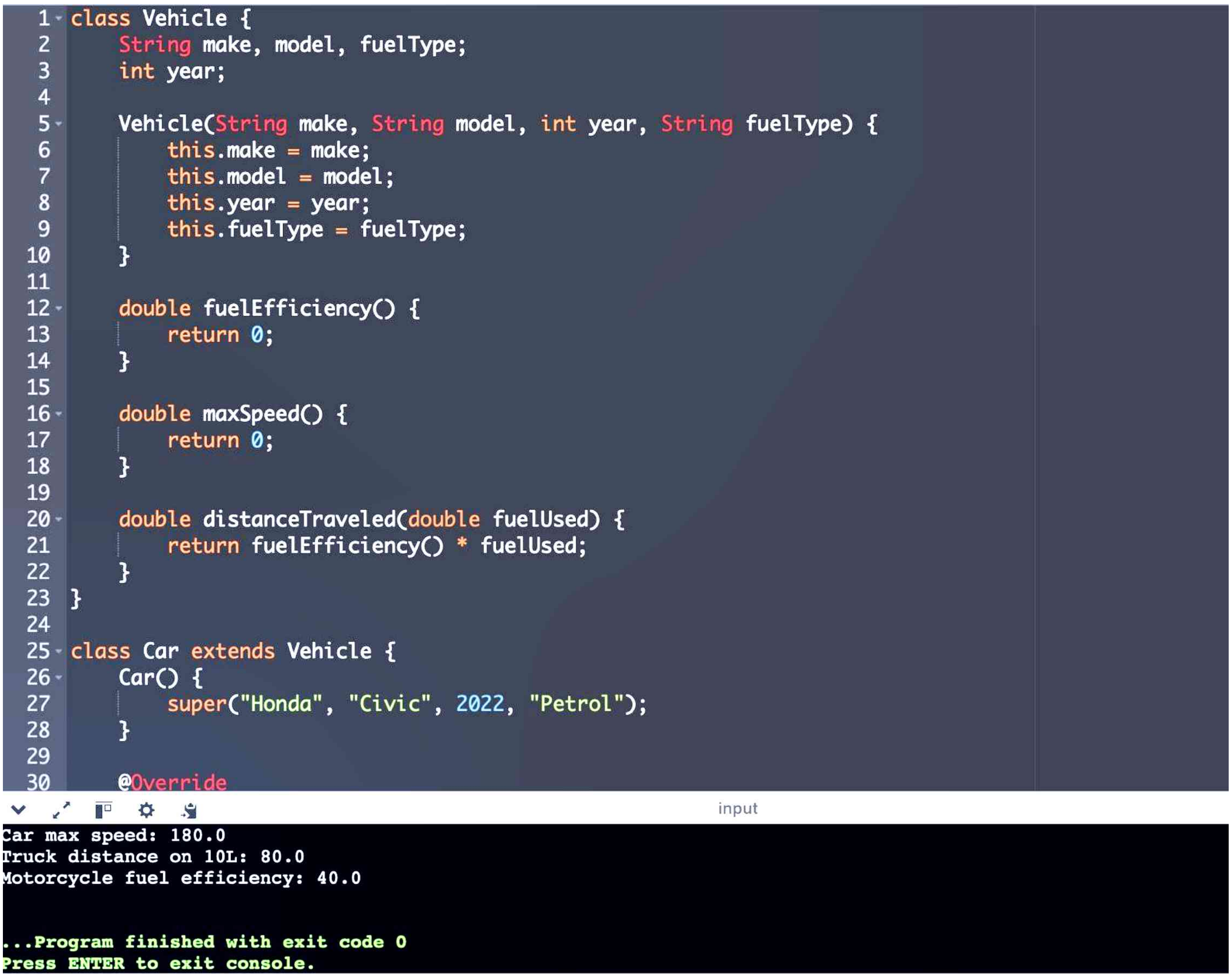
```java
class Person {
    String getFirstName() {
        return "John";
    }

    String getLastName() {
        return "Doe";
    }
}

class Employee extends Person {
    String getEmployeeId() {
        return "EMP123";
    }

    @Override
    String getLastName() {
        return "Doe - Software Developer";
    }
}

public class Main {
    public static void main(String[] args) {
        Employee emp = new Employee();
        System.out.println(emp.getFirstName());
        System.out.println(emp.getLastName());
        System.out.println(emp.getEmployeeId());
    }
}
```

```
John
Doe - Software Developer
EMP123


...Program finished with exit code 0
Press ENTER to exit console.
```



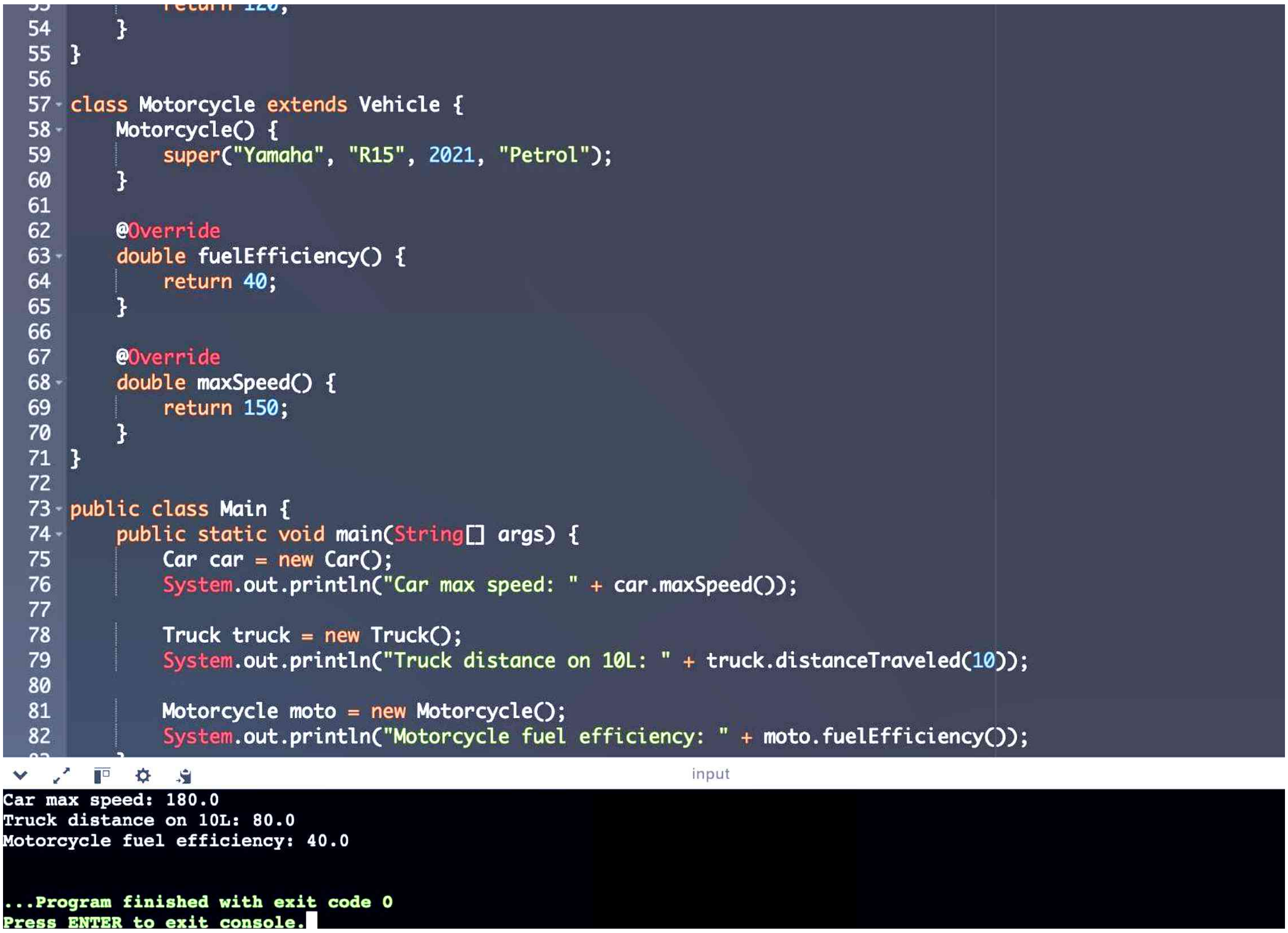
```java
class Shape {
    double getPerimeter() {
        return 0;
    }

    double getArea() {
        return 0;
    }
}

class Circle extends Shape {
    double radius = 5;

    @Override
    double getPerimeter() {
        return 2 * Math.PI * radius;
    }

    @Override
    double getArea() {
        return Math.PI * radius * radius;
    }
}

public class Main {
    public static void main(String[] args) {
        Circle c = new Circle();
        System.out.println("Perimeter: " + c.getPerimeter());
        System.out.println("Area: " + c.getArea());
```

```
Perimeter: 31.41592653589793
Area: 78.53981633974483


...Program finished with exit code 0
Press ENTER to exit console.
```



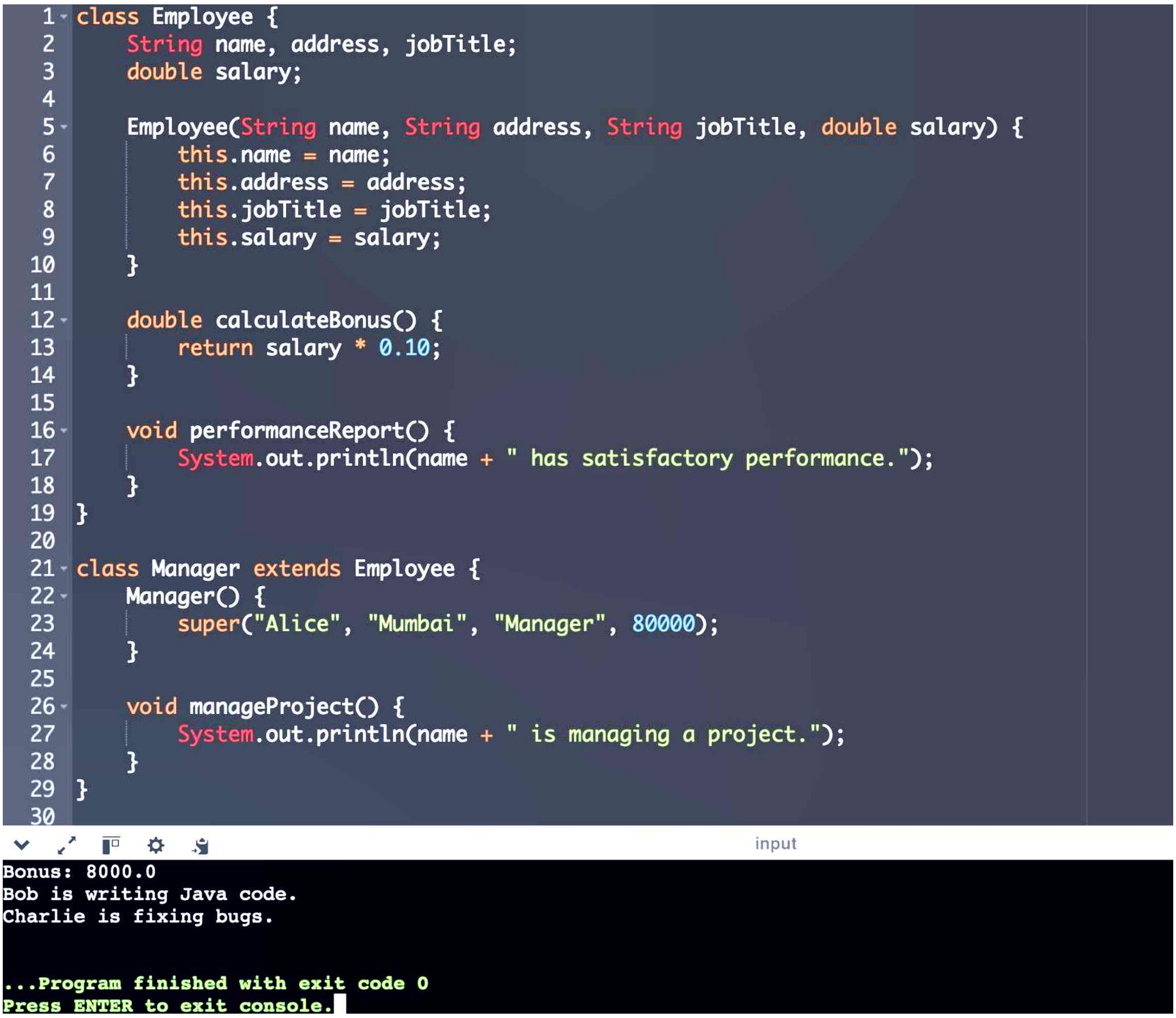
```
class Vehicle {
    String make, model, fuelType;
    int year;

    Vehicle(String make, String model, int year, String fuelType) {
        this.make = make;
        this.model = model;
        this.year = year;
        this.fuelType = fuelType;
    }

    double fuelEfficiency() {
        return 0;
    }

    double maxSpeed() {
        return 0;
    }

    double distanceTraveled(double fuelUsed) {
        return fuelEfficiency() * fuelUsed;
    }
}

class Car extends Vehicle {
    Car() {
        super("Honda", "Civic", 2022, "Petrol");
    }

    @Override
```

```
Car max speed: 180.0
Truck distance on 10L: 80.0
Motorcycle fuel efficiency: 40.0


...Program finished with exit code 0
Press ENTER to exit console.
```



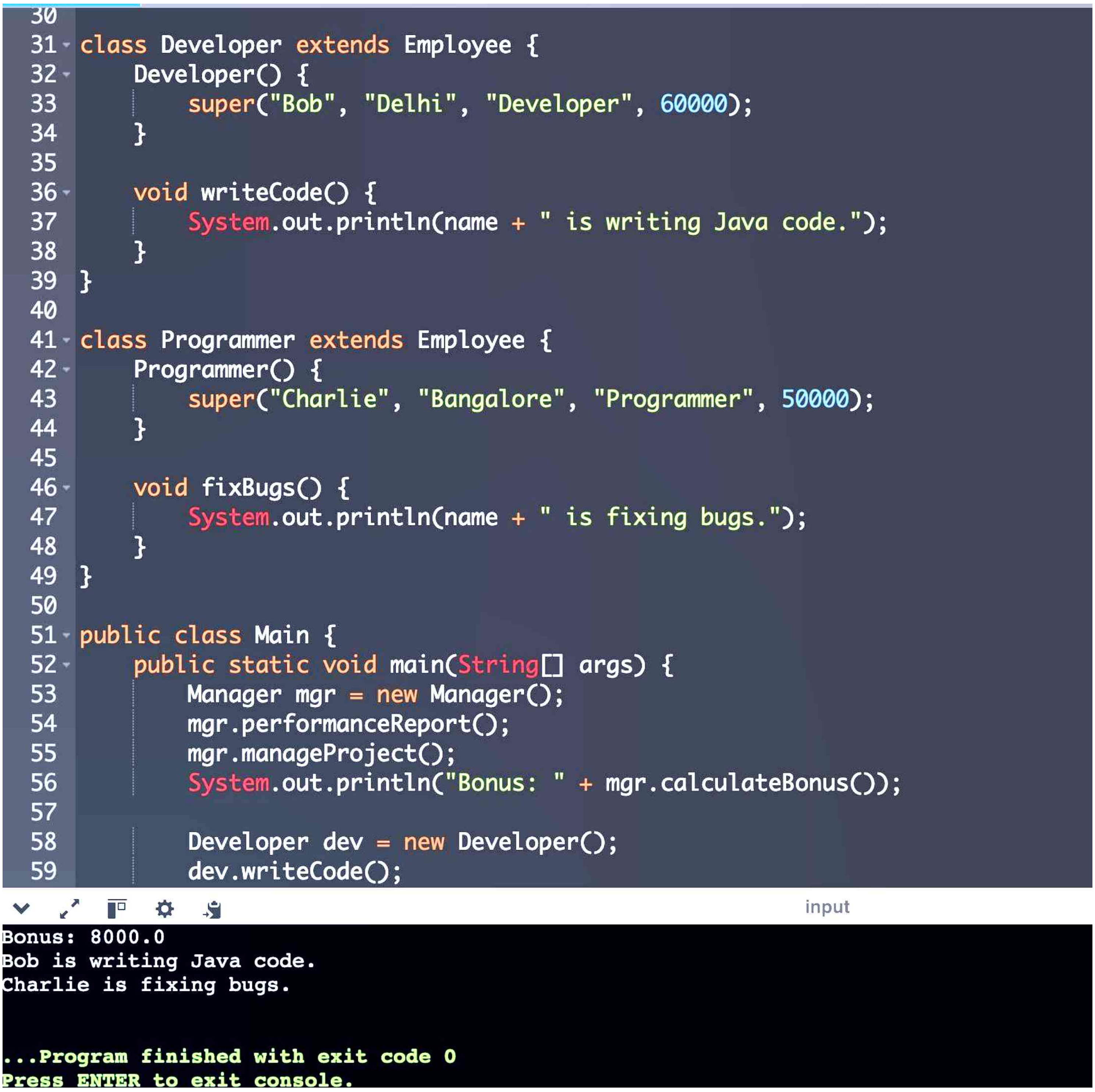
```java
        return 120;
    }
}

class Motorcycle extends Vehicle {
    Motorcycle() {
        super("Yamaha", "R15", 2021, "Petrol");
    }

    @Override
    double fuelEfficiency() {
        return 40;
    }

    @Override
    double maxSpeed() {
        return 150;
    }
}

public class Main {
    public static void main(String[] args) {
        Car car = new Car();
        System.out.println("Car max speed: " + car.maxSpeed());

        Truck truck = new Truck();
        System.out.println("Truck distance on 10L: " + truck.distanceTraveled(10));

        Motorcycle moto = new Motorcycle();
        System.out.println("Motorcycle fuel efficiency: " + moto.fuelEfficiency());
```

```
Car max speed: 180.0
Truck distance on 10L: 80.0
Motorcycle fuel efficiency: 40.0


...Program finished with exit code 0
Press ENTER to exit console.
```



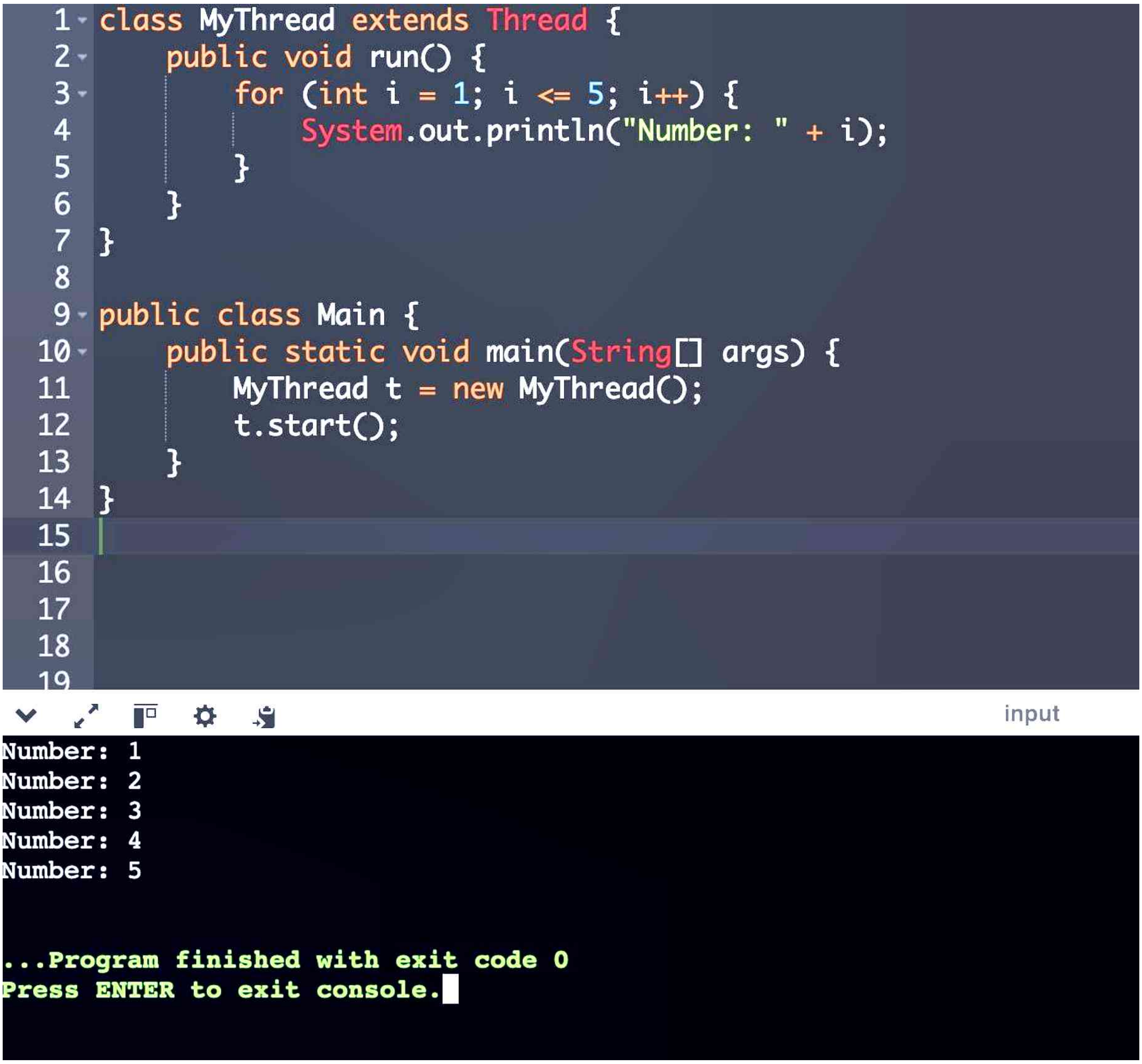
```java
class Employee {
    String name, address, jobTitle;
    double salary;

    Employee(String name, String address, String jobTitle, double salary) {
        this.name = name;
        this.address = address;
        this.jobTitle = jobTitle;
        this.salary = salary;
    }

    double calculateBonus() {
        return salary * 0.10;
    }

    void performanceReport() {
        System.out.println(name + " has satisfactory performance.");
    }
}

class Manager extends Employee {
    Manager() {
        super("Alice", "Mumbai", "Manager", 80000);
    }

    void manageProject() {
        System.out.println(name + " is managing a project.");
    }
}
```

```
Bonus: 8000.0
Bob is writing Java code.
Charlie is fixing bugs.


...Program finished with exit code 0
Press ENTER to exit console.
```



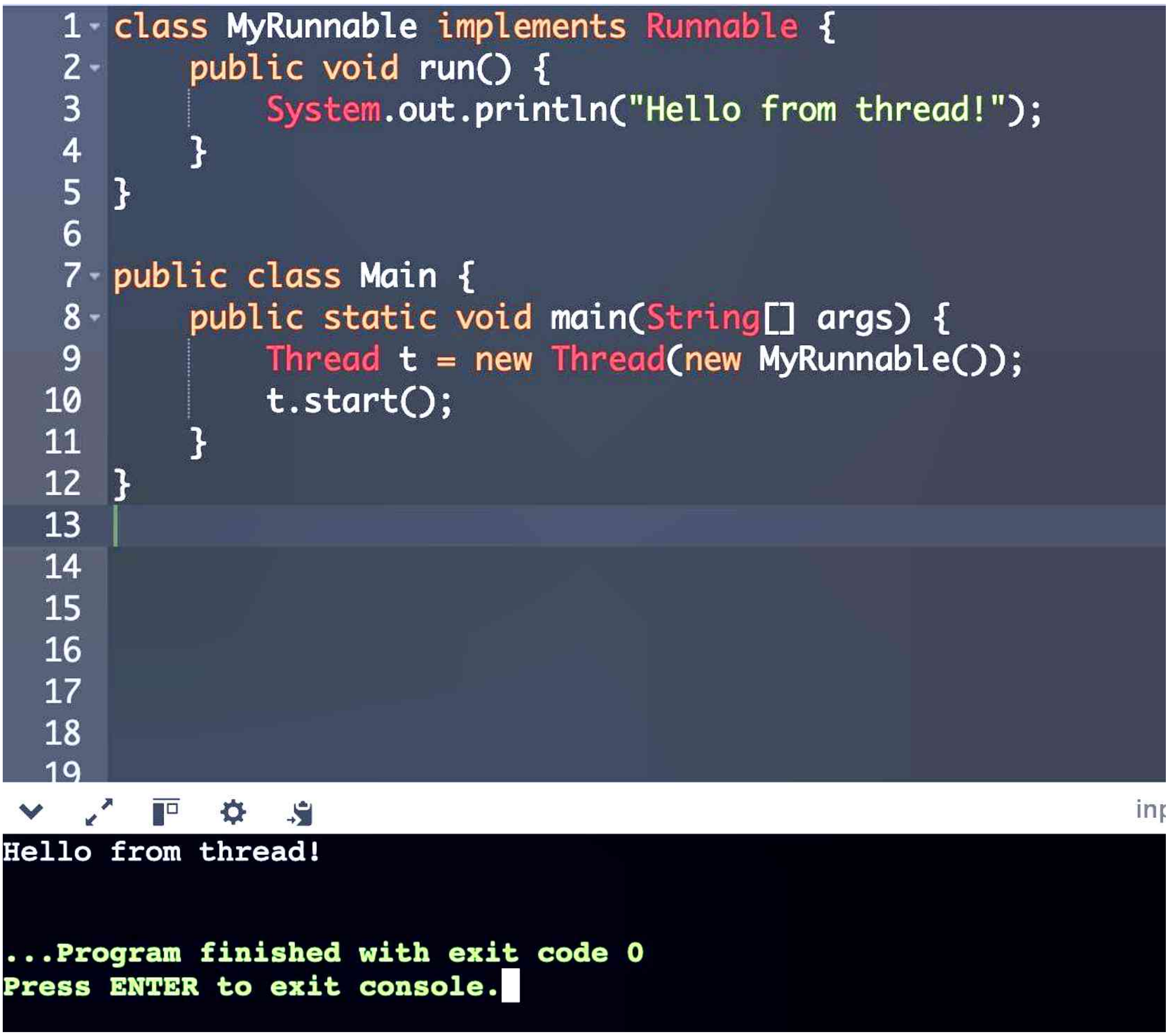
```java

class Developer extends Employee {
    Developer() {
        super("Bob", "Delhi", "Developer", 60000);
    }

    void writeCode() {
        System.out.println(name + " is writing Java code.");
    }
}

class Programmer extends Employee {
    Programmer() {
        super("Charlie", "Bangalore", "Programmer", 50000);
    }

    void fixBugs() {
        System.out.println(name + " is fixing bugs.");
    }
}

public class Main {
    public static void main(String[] args) {
        Manager mgr = new Manager();
        mgr.performanceReport();
        mgr.manageProject();
        System.out.println("Bonus: " + mgr.calculateBonus());

        Developer dev = new Developer();
        dev.writeCode();
```

```
Bonus: 8000.0
Bob is writing Java code.
Charlie is fixing bugs.


...Program finished with exit code 0
Press ENTER to exit console.
```



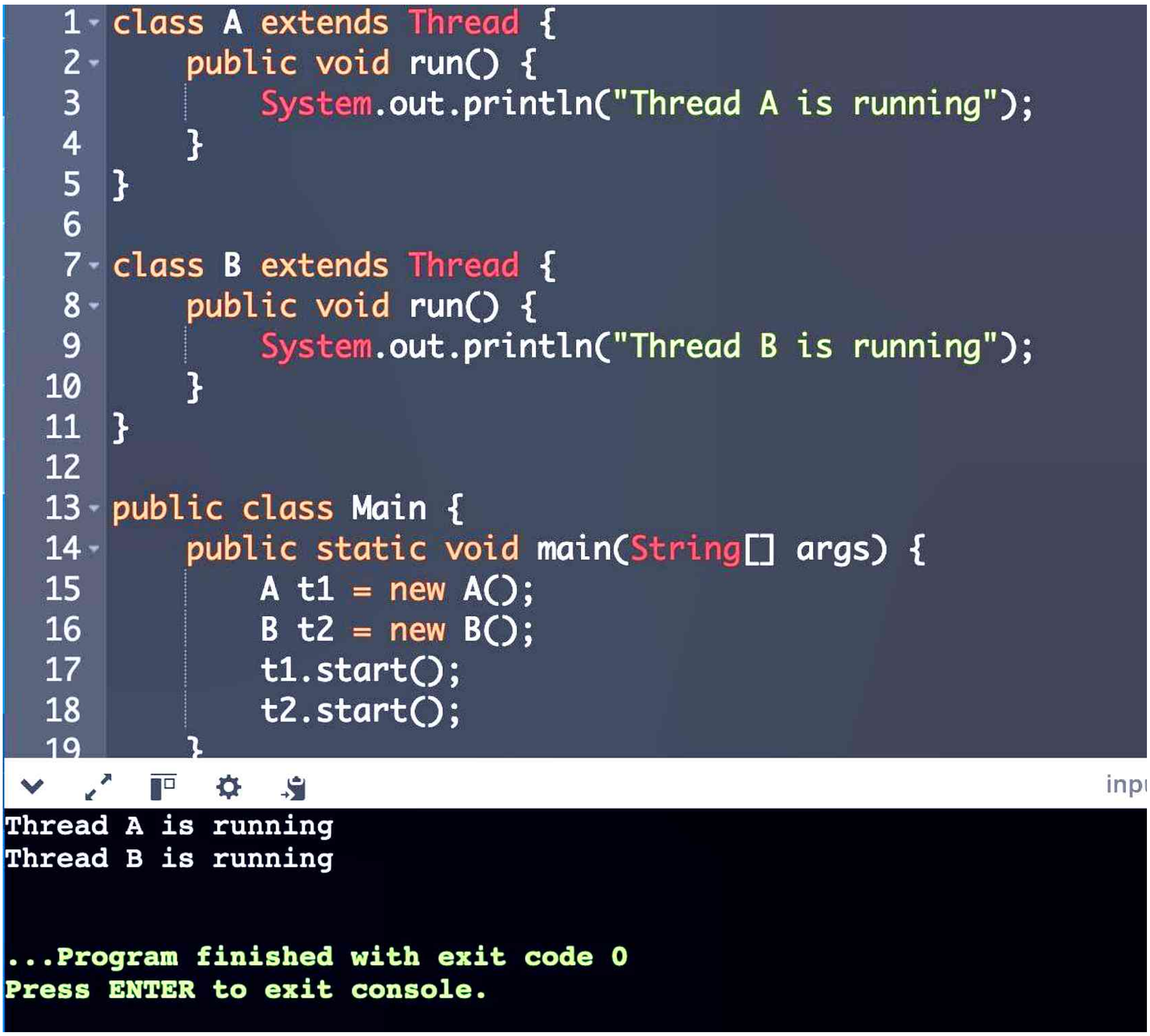
```java
class MyThread extends Thread {
    public void run() {
        for (int i = 1; i <= 5; i++) {
            System.out.println("Number: " + i);
        }
    }
}

public class Main {
    public static void main(String[] args) {
        MyThread t = new MyThread();
        t.start();
    }
}
```

```
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5


...Program finished with exit code 0
Press ENTER to exit console.
```



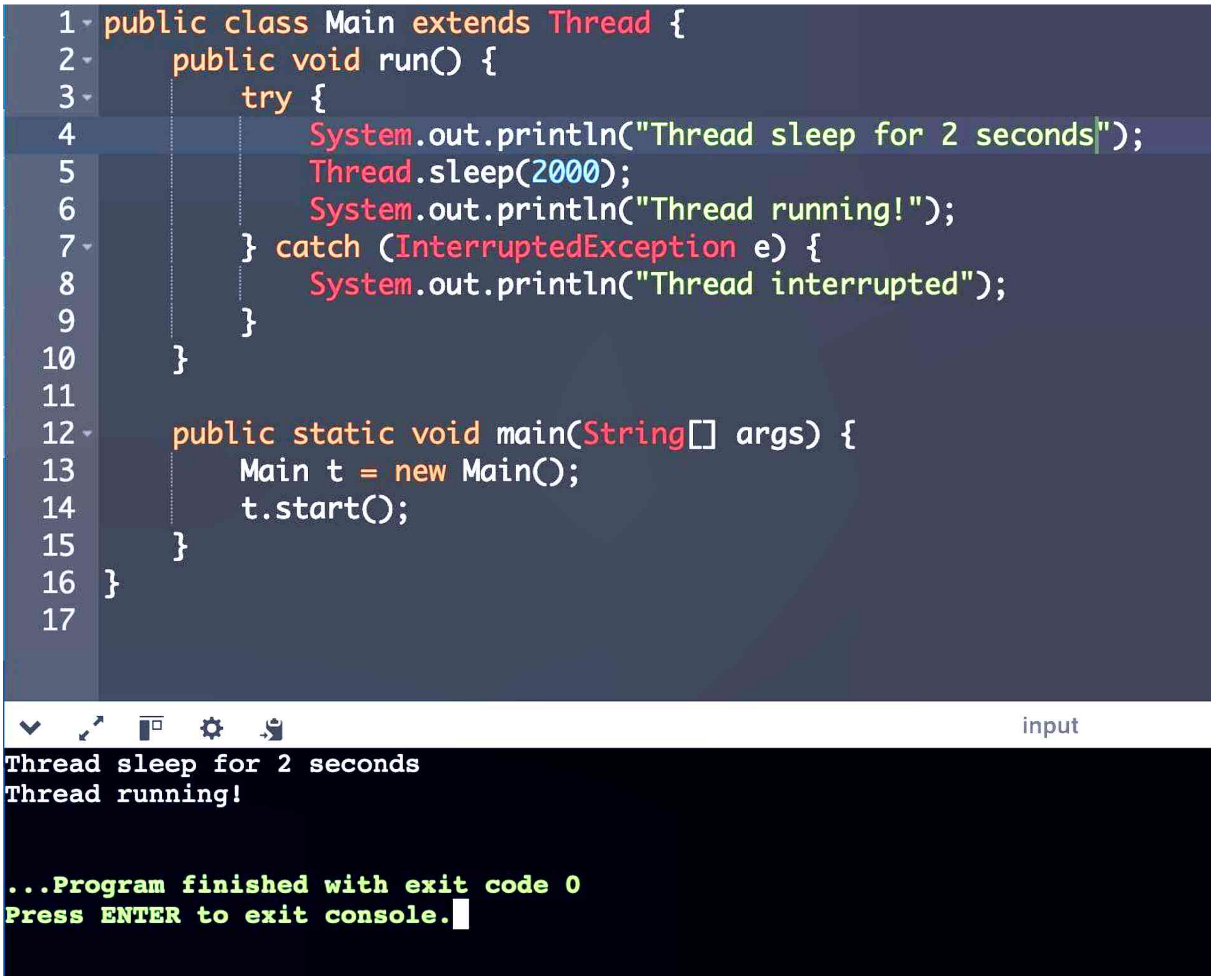
```java
class MyRunnable implements Runnable {
    public void run() {
        System.out.println("Hello from thread!");
    }
}

public class Main {
    public static void main(String[] args) {
        Thread t = new Thread(new MyRunnable());
        t.start();
    }
}
```

```
Hello from thread!


...Program finished with exit code 0
Press ENTER to exit console.
```



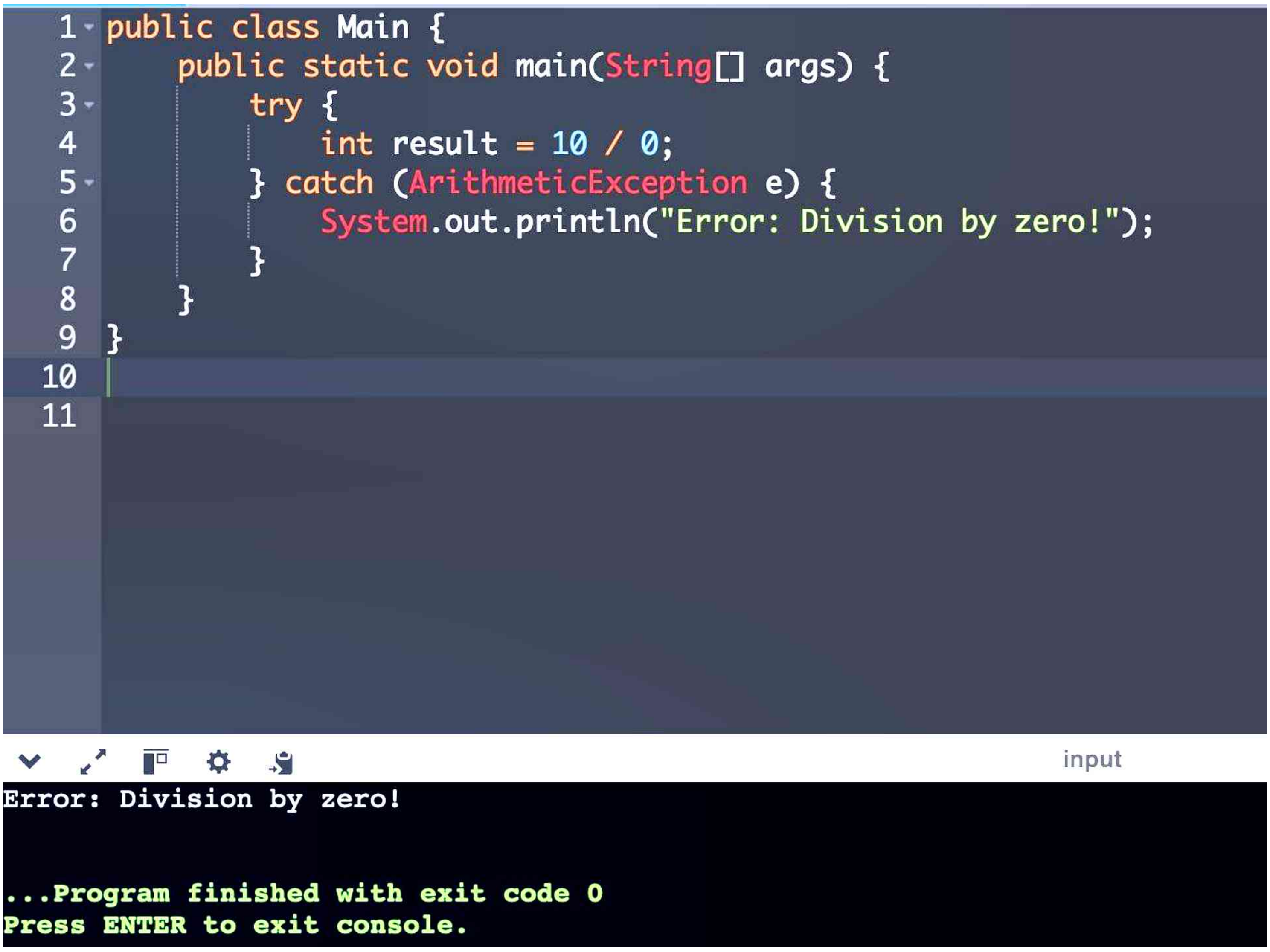
```java
class A extends Thread {
    public void run() {
        System.out.println("Thread A is running");
    }
}

class B extends Thread {
    public void run() {
        System.out.println("Thread B is running");
    }
}

public class Main {
    public static void main(String[] args) {
        A t1 = new A();
        B t2 = new B();
        t1.start();
        t2.start();
    }
```

```
Thread A is running
Thread B is running


...Program finished with exit code 0
Press ENTER to exit console.
```



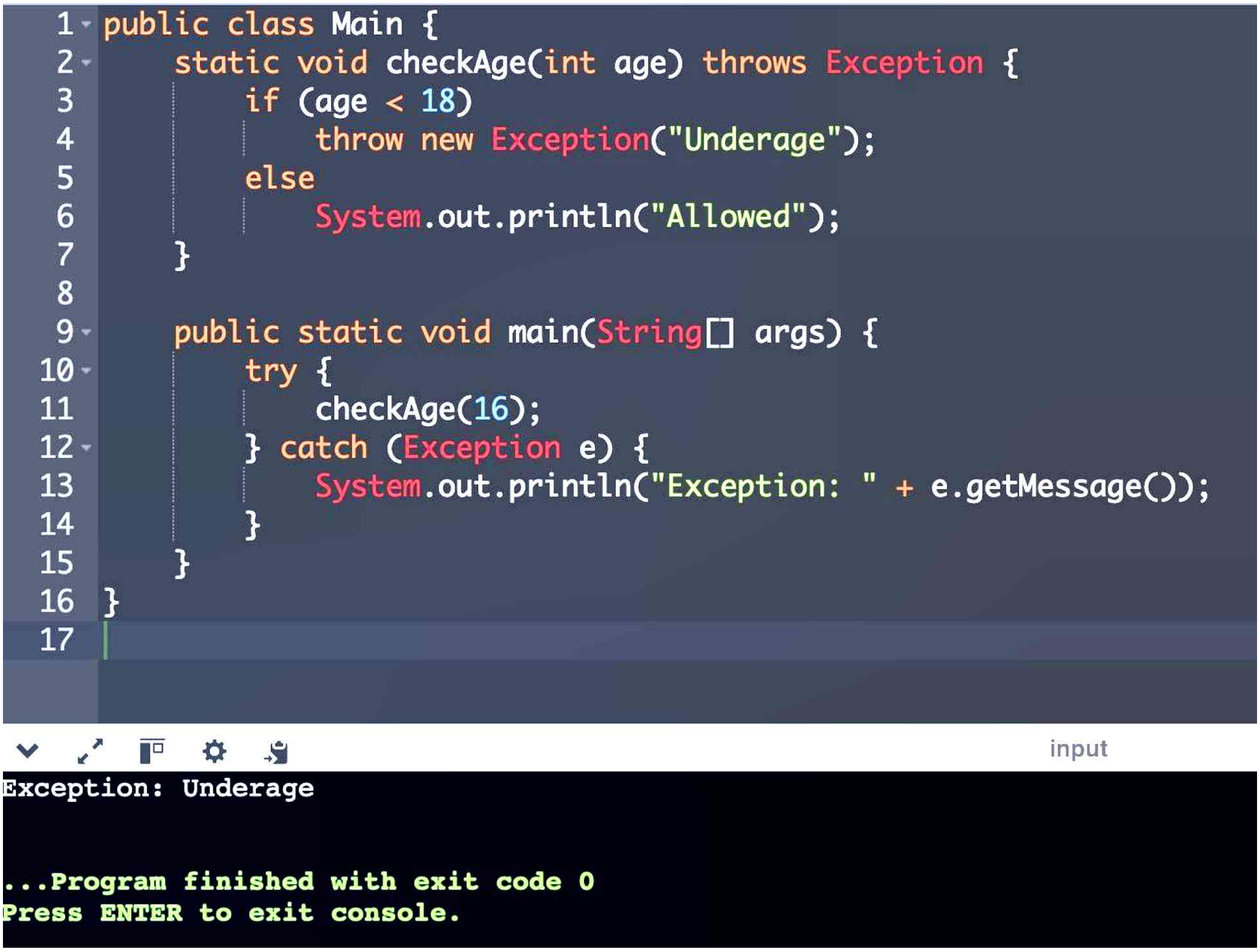
```java
public class Main extends Thread {
    public void run() {
        try {
            System.out.println("Thread sleep for 2 seconds");
            Thread.sleep(2000);
            System.out.println("Thread running!");
        } catch (InterruptedException e) {
            System.out.println("Thread interrupted");
        }
    }

    public static void main(String[] args) {
        Main t = new Main();
        t.start();
    }
}
```

```
Thread sleep for 2 seconds
Thread running!


...Program finished with exit code 0
Press ENTER to exit console.
```



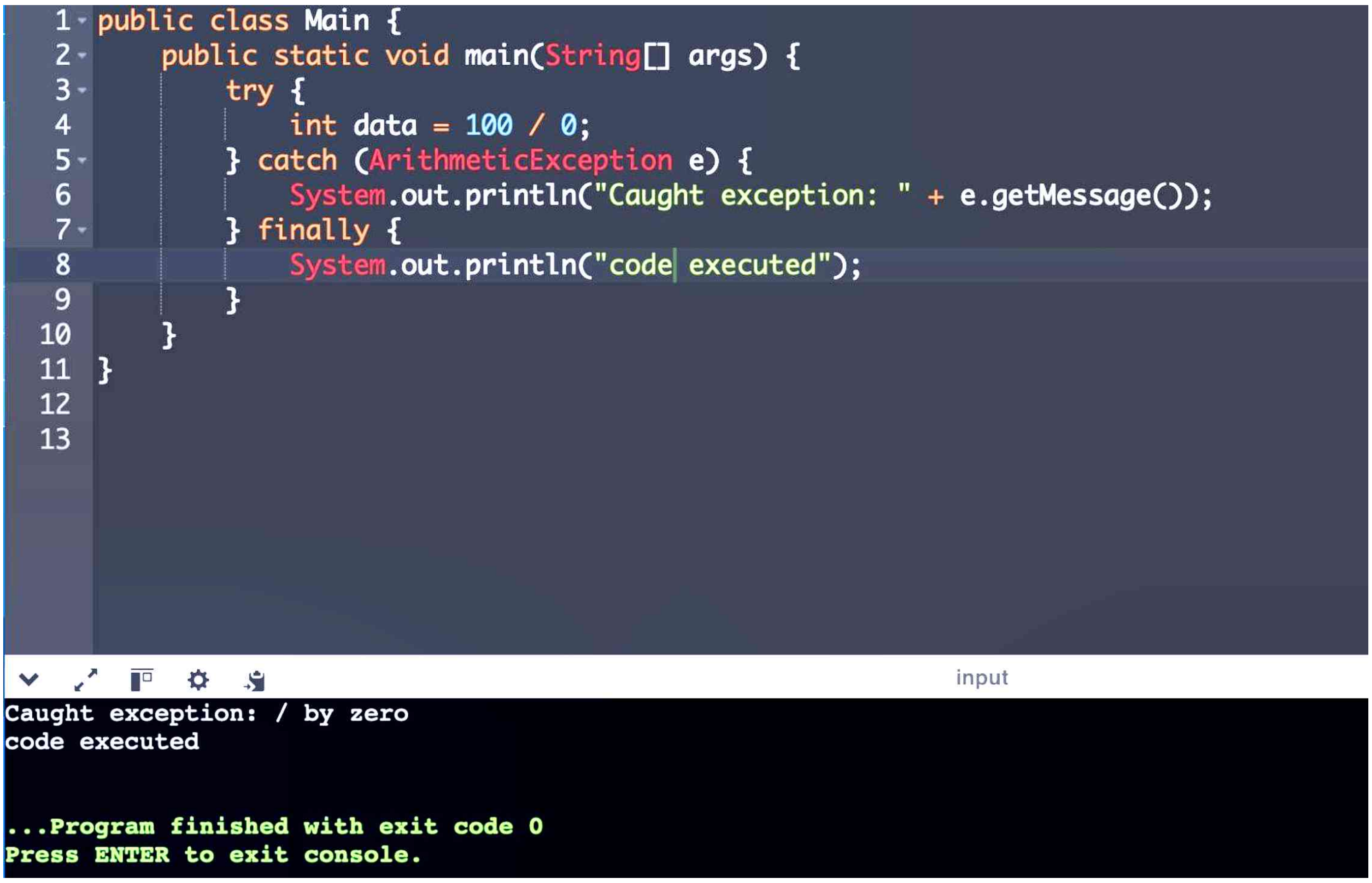
```java
public class Main {
    public static void main(String[] args) {
        try {
            int result = 10 / 0;
        } catch (ArithmeticException e) {
            System.out.println("Error: Division by zero!");
        }
    }
}
```

```
Error: Division by zero!


...Program finished with exit code 0
Press ENTER to exit console.
```



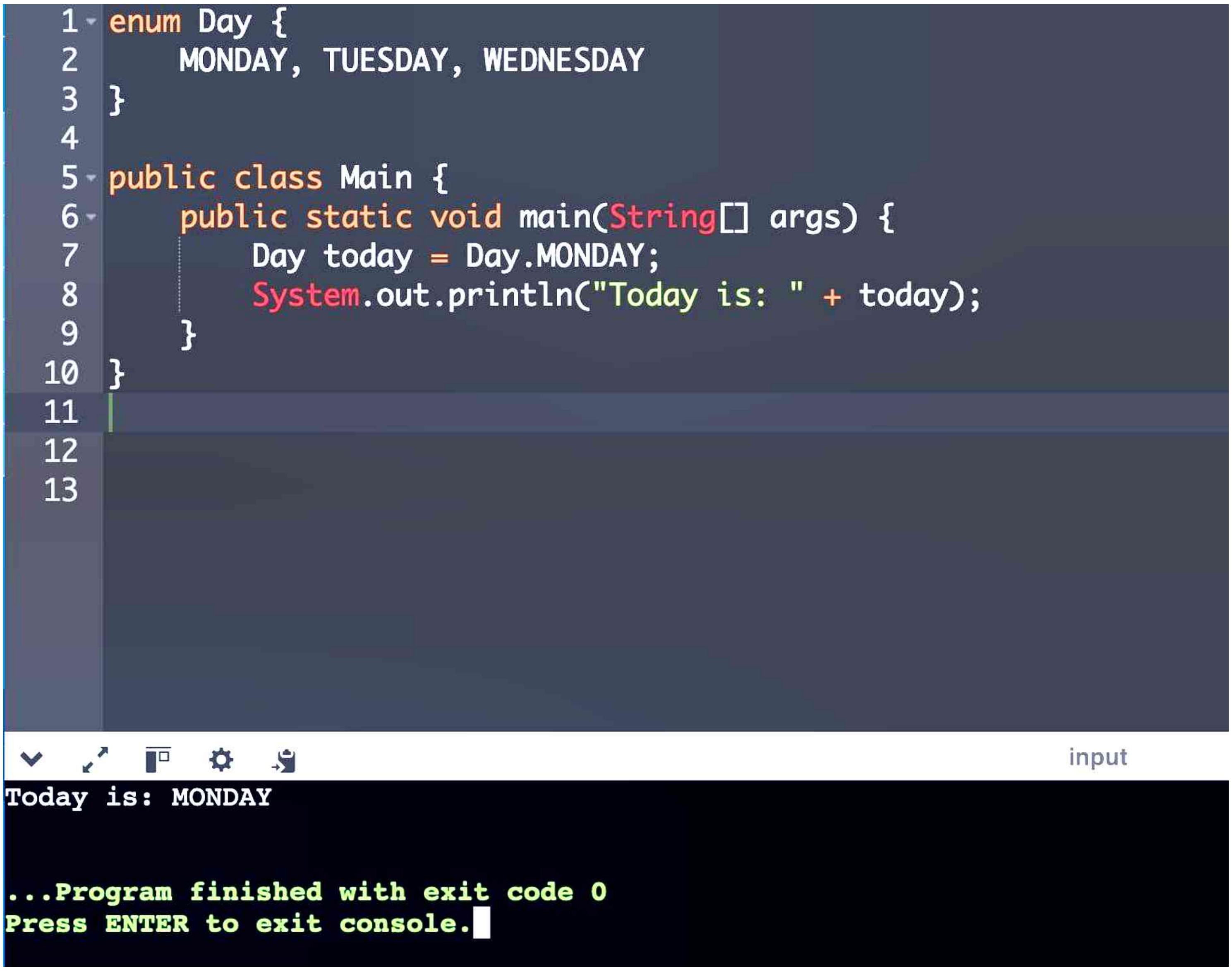
```java
public class Main {
    static void checkAge(int age) throws Exception {
        if (age < 18)
            throw new Exception("Underage");
        else
            System.out.println("Allowed");
    }

    public static void main(String[] args) {
        try {
            checkAge(16);
        } catch (Exception e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}
```

```
Exception: Underage


...Program finished with exit code 0
Press ENTER to exit console.
```



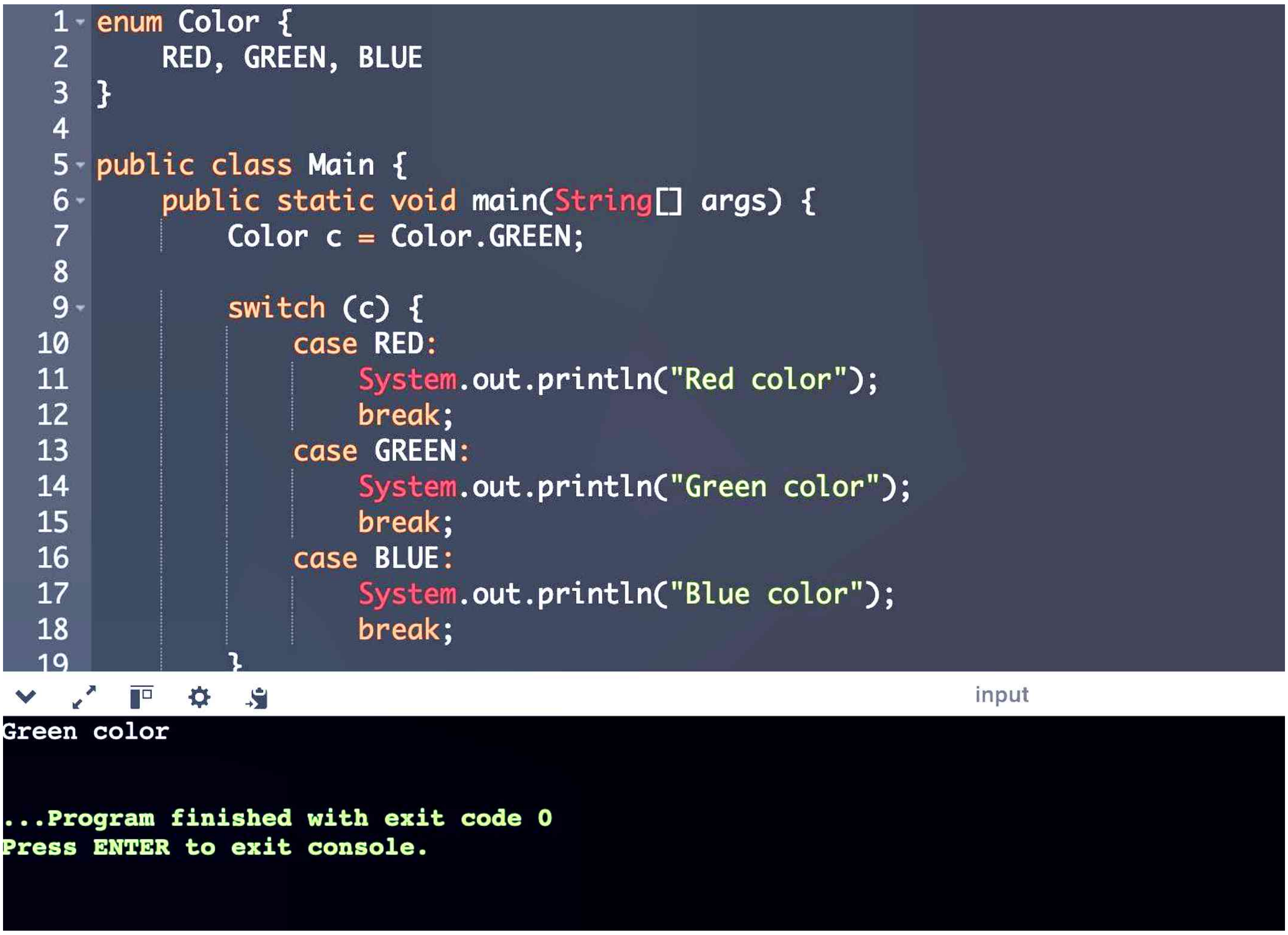
```java
public class Main {
    public static void main(String[] args) {
        try {
            int data = 100 / 0;
        } catch (ArithmeticException e) {
            System.out.println("Caught exception: " + e.getMessage());
        } finally {
            System.out.println("code executed");
        }
    }
}
```

```
Caught exception: / by zero
code executed


...Program finished with exit code 0
Press ENTER to exit console.
```



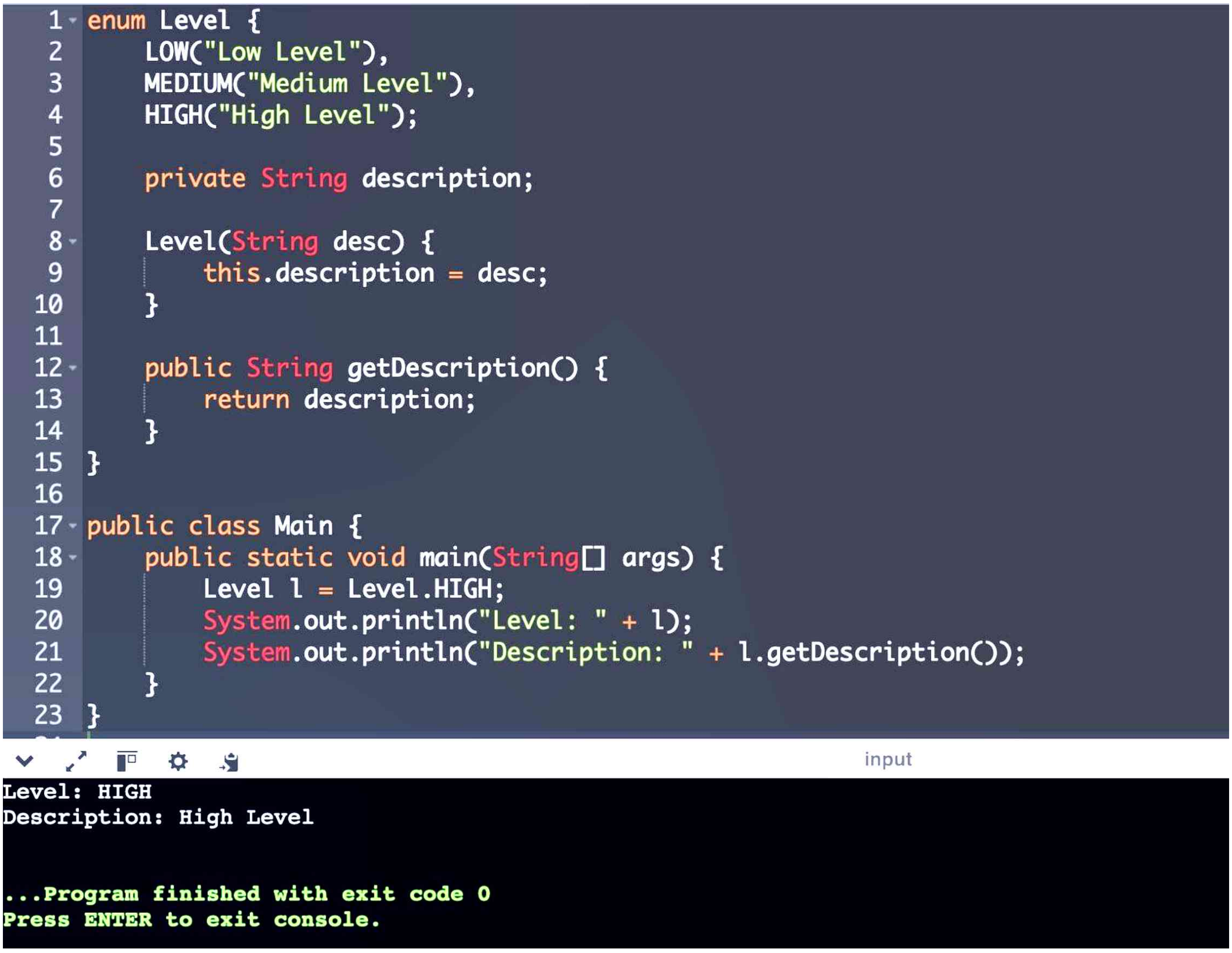
```java
enum Day {
    MONDAY, TUESDAY, WEDNESDAY
}

public class Main {
    public static void main(String[] args) {
        Day today = Day.MONDAY;
        System.out.println("Today is: " + today);
    }
}
```

```
Today is: MONDAY


...Program finished with exit code 0
Press ENTER to exit console.
```



```java
enum Color {
    RED, GREEN, BLUE
}

public class Main {
    public static void main(String[] args) {
        Color c = Color.GREEN;

        switch (c) {
            case RED:
                System.out.println("Red color");
                break;
            case GREEN:
                System.out.println("Green color");
                break;
            case BLUE:
                System.out.println("Blue color");
                break;
        }
```

```
Green color


...Program finished with exit code 0
Press ENTER to exit console.
```



```java
enum Level {
    LOW("Low Level"),
    MEDIUM("Medium Level"),
    HIGH("High Level");

    private String description;

    Level(String desc) {
        this.description = desc;
    }

    public String getDescription() {
        return description;
    }
}

public class Main {
    public static void main(String[] args) {
        Level l = Level.HIGH;
        System.out.println("Level: " + l);
        System.out.println("Description: " + l.getDescription());
    }
}
```

```
Level: HIGH
Description: High Level


...Program finished with exit code 0
Press ENTER to exit console.
```