****

**VIT Bhopal University**

**Kotrikalan village,**

**Sehore Dt.Bhopal-466144**

**DEPARTMENT OF- COMPUTER SCIENCE AND ENGINEERING**

ACADEMIC YEAR: 2024-2025

**COURSE NAME**



**CSE2006-** PROGRAMMING IN JAVA LABORATORY

Winter SEMSTER

**Submitted By**

**Name: Devam Pandey**

**Reg No: 23BCE10731**

Faculty Signature

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| E.NO | EXPERIMENT NAME | Pg. No. | SIGN |
| 1 | Write a Program to call base class constructor and method using super keyword in java | 4 |  |
| 2 | Write a Java program to create a base class Shape with a method called calculateArea().  Create three subclasses: Circle, Rectangle, and Triangle. Override the calculateArea() method in each subclass to calculate and return the shape's area. | 6 |  |
| 3 | Write a Java program to display Pascal's triangle.  Test Data  Input number of rows: 5  Expected Output :  Input number of rows: 5  1  1 1  1 2 1  1 3 3 1  1 4 6 4 1 | 9 |  |
| 4 | Write a Java program to create a class called Employee with private instance variables employee\_id, employee\_name, and employee\_salary. Provide public getter and setter methods to access and modify the id and name variables, but provide a getter method for the salary variable that returns a formatted string. | 9 |  |
| 5 | Write a following Java program using Flow control statements to get the value at run time.   * Given year is Leap year or not. * Factorial of a given number using method. * Fibonacci series of a given count | 13 |  |
| 6 | 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. Implement methods for calculating fuel efficiency, distance traveled, and maximum speed | 15 |  |
| 7 | Write a Java program to create an abstract class GeometricShape with abstract methods area() and perimeter(). Create subclasses Triangle and Square that extend the GeometricShape class and implement the respective methods to calculate the area and perimeter of each shape. | 18 |  |
| 8 | Write a Java programming to create a banking system with three classes - Bank Account, SavingsAccount, and CurrentAccount. The bank should have a list of accounts and methods for adding them. Accounts should be an interface with methods to deposit, withdraw, calculate interest, and view balances. SavingsAccount and CurrentAccount should implement the Account interface and have their own unique methods. | 20 |  |
| 9 | Write a Java program to create a method that takes a string as input and throws an exception if the string does not contain vowels. | 25 |  |
| 10 | Implement the list creation using ArrayList Class, Vector and Stack with example and output. | 26 |  |
| 11 | Write a Java program that uses the CountDownLatch class to synchronize the start and finish of multiple threads(0-9 and 9-0). | 28 |  |
| 12 | Write a Java program to implement a concurrent web crawler that crawls multiple websites simultaneously using threads. | 31 |  |

1. **Write a Program to call base class constructor and method using super keyword in java**

**AIM:** To Write a Program to call base class constructor and method using super keyword in java

**ALGORITHM:**

Step1: We created a parent class NAnimal

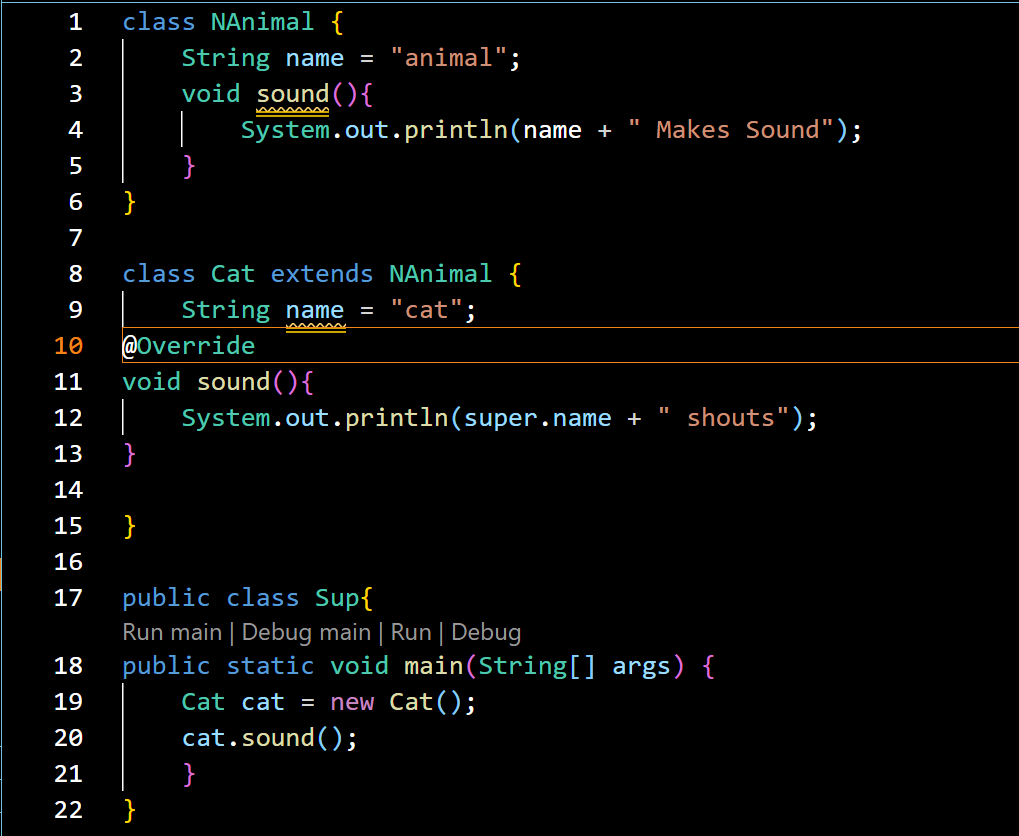
Step2: We created a string variable “name”

Step3: We inherited the NAnimal class to a child class named Cat

Step4: We used the super keyword to access the name variable from the NAnimal class

Step5: End

**PROGRAM:**



**OUTPUT:**



**RESULT:** We wrote a Program to call base class constructor and method using super keyword in java

Q2. Write a Java program to create a base class Shape with a method called calculateArea().

Create three subclasses: Circle, Rectangle, and Triangle. Override the calculateArea() method in each subclass to calculate and return the shape's area.

**AIM:** To Write a Java program to create a base class Shape with a method called calculateArea().

Create three subclasses: Circle, Rectangle, and Triangle. Override the calculateArea() method in each subclass to calculate and return the shape's area.

**ALGORITHM:**

Step1: Start

Step2: We created an abstract class Shape

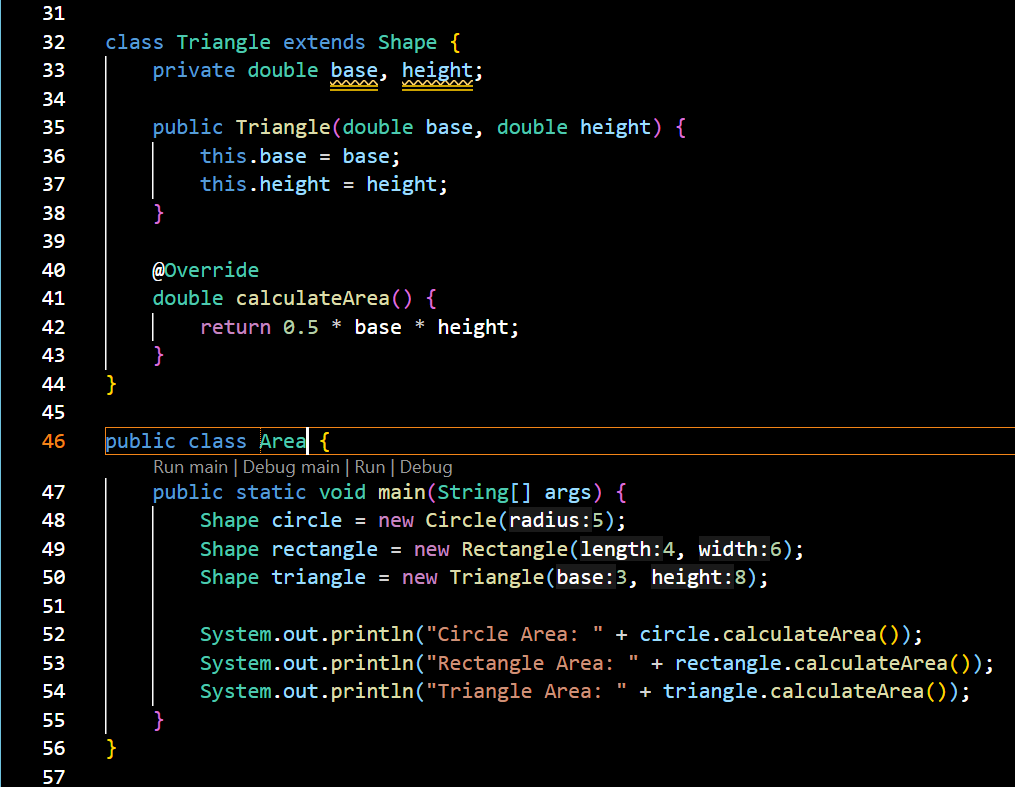
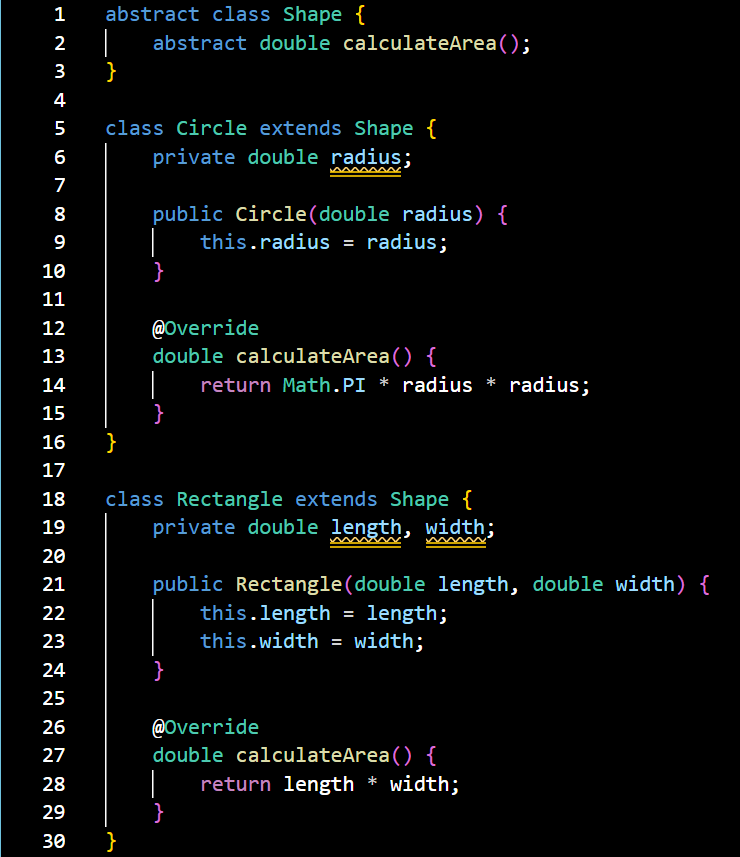
Step3: We created an abstract method calculateArea()

Step3: We inherited the Shape class to a child classes named Circle, Rectangle and Triangle

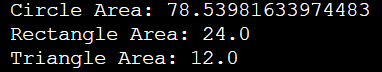
Step4: We override the calculateArea method according to the shape

Step5: End

**PROGRAM:**



**OUTPUT:**



**RESULT:**

We created a Java program to create a base class Shape with a method called calculateArea().

Create three subclasses: Circle, Rectangle, and Triangle. Override the calculateArea() method in each subclass to calculate and return the shape's area.

**Q3. Write a Java program to display Pascal's triangle.**

**Test Data**

**Input number of rows: 5**

**Expected Output :**

**Input number of rows: 5**

**1**

**1 1**

**1 2 1**

**1 3 3 1**

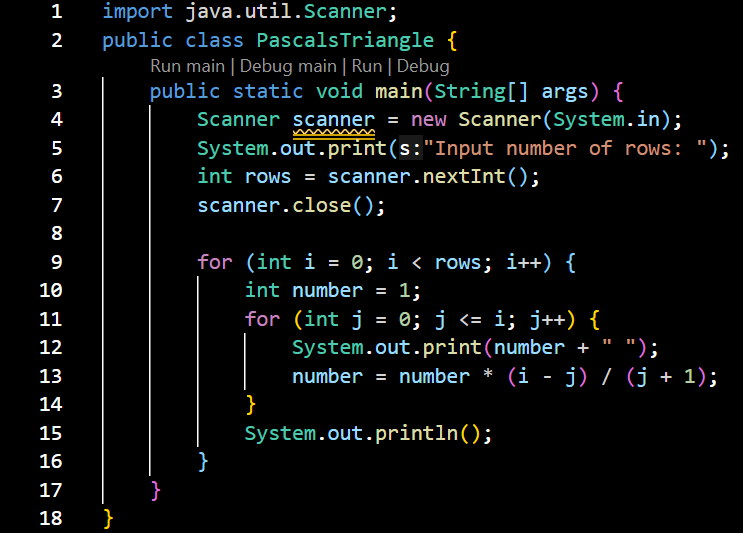
**1 4 6 4 1**

**AIM:** To Write a Java program to display Pascal's triangle.

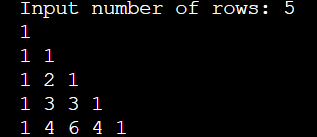
**ALGORITHM:**

1. Start
2. Take no. of rows as input
3. Run the loop till the input
4. Print the values
5. End

**PROGRAM:**

****

**OUTPUT:**



**RESULT:**

We created a Java program to display Pascal's triangle.

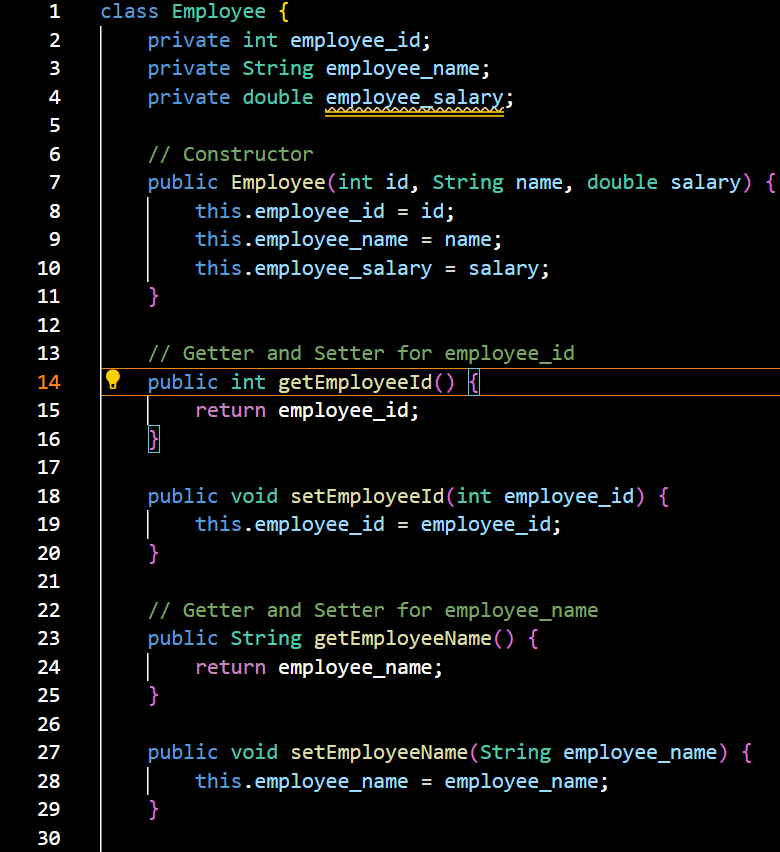
**Q4. Write a Java program to create a class called Employee with private instance variables employee\_id, employee\_name, and employee\_salary. Provide public getter and setter methods to access and modify the id and name variables, but provide a getter method for the salary variable that returns a formatted string.**

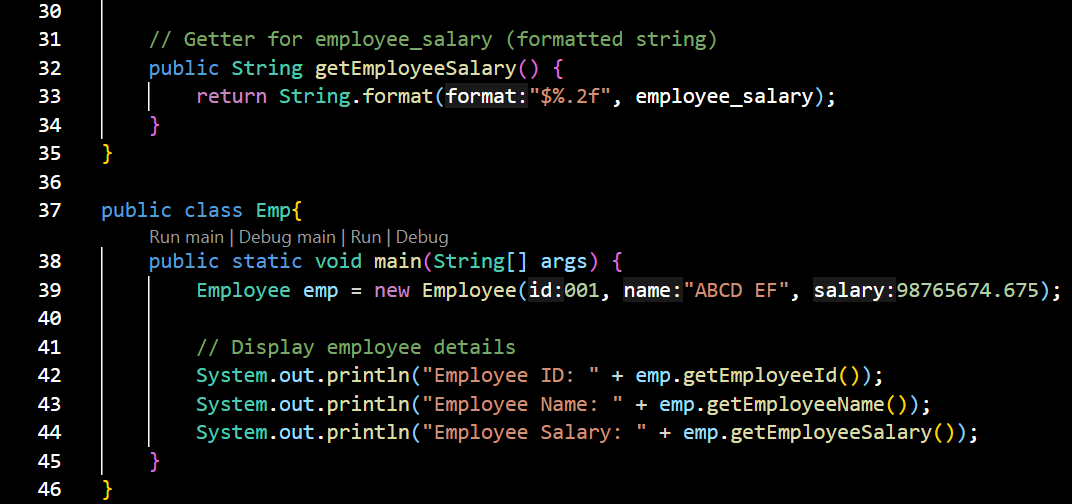
**AIM:** To Write a Java program to create a class called Employee with private instance variables employee\_id, employee\_name, and employee\_salary. Provide public getter and setter methods to access and modify the id and name variables, but provide a getter method for the salary variable that returns a formatted string.

**ALGORITHM:**

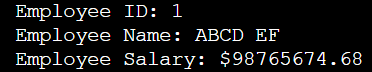
1. **Start**
2. Create a class employee
3. Create private instances for employee\_id, employee\_name, and employee\_salary
4. Create public getter and setter methods to access the id and name
5. Create only a getter method for salary that formats the output before returning
6. Create a new class emp
7. Create an object of employee
8. Use the setter methods to input the values
9. Use the getter methods to see the values
10. End

**PROGRAM:**



****

**OUTPUT:**



**RESULT:** We Created a Java program to create a class called Employee with private instance variables employee\_id, employee\_name, and employee\_salary. Provide public getter and setter methods to access and modify the id and name variables, but provide a getter method for the salary variable that returns a formatted string.

**Q5. Write a following Java program using Flow control statements to get the value at run time.**

* **Given year is Leap year or not.**
* **Factorial of a given number using method.**
* **Fibonacci series of a given count.**

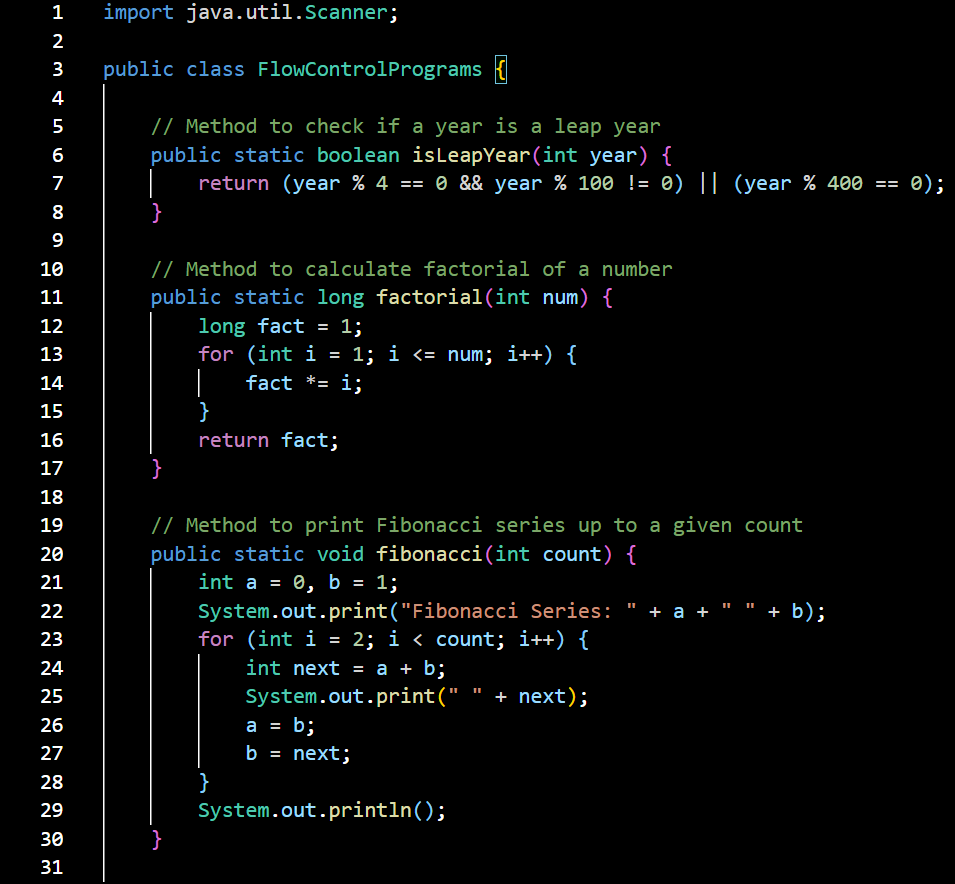
**AIM:** To Write a following Java program using Flow control statements to get the value at run time.

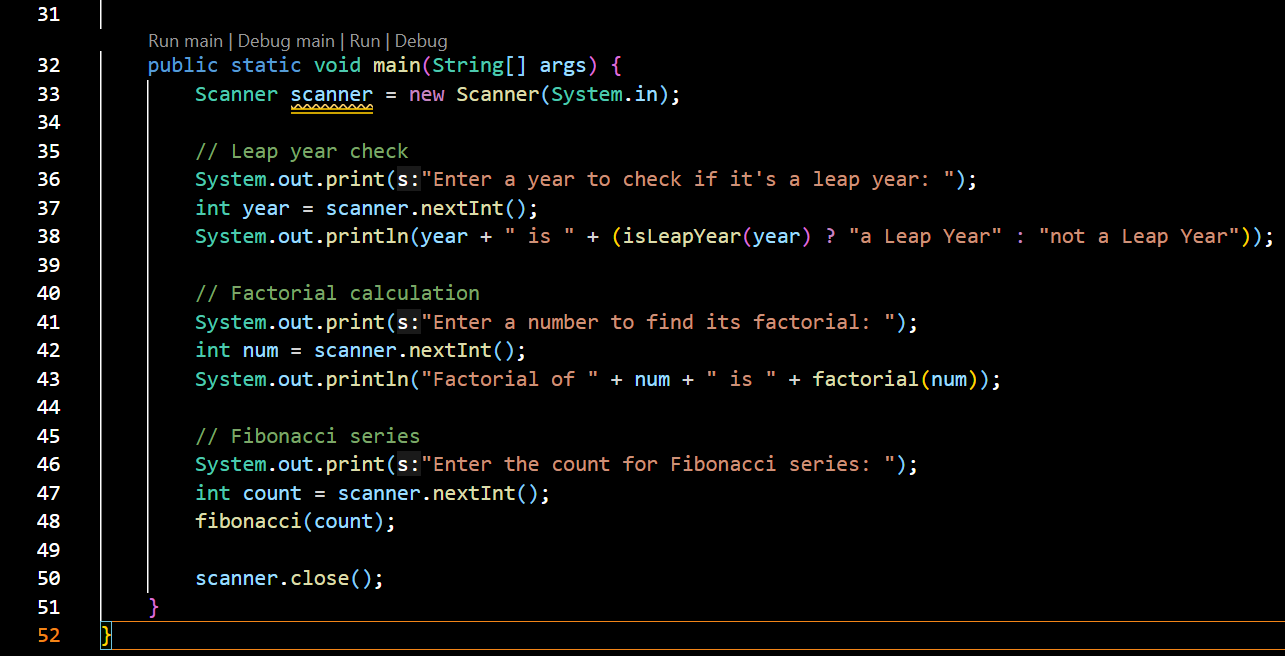
* Given year is Leap year or not.
* Factorial of a given number using method.
* Fibonacci series of a given count.

**ALGORITHM:**

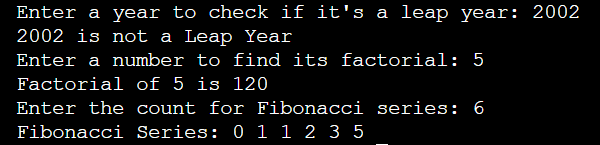
1. Start
2. Create public class
3. Create a method for Leap year Checking
4. Create a method for factorial calculation
5. Create a method for Fibonacci series
6. Create a main function
7. Call all the methods
8. End

**PROGRAM:**





**OUTPUT:**

**RESULT:** We created Write a following Java program using Flow control statements to get the value at run time.

* Given year is Leap year or not.
* Factorial of a given number using method.
* Fibonacci series of a given count.

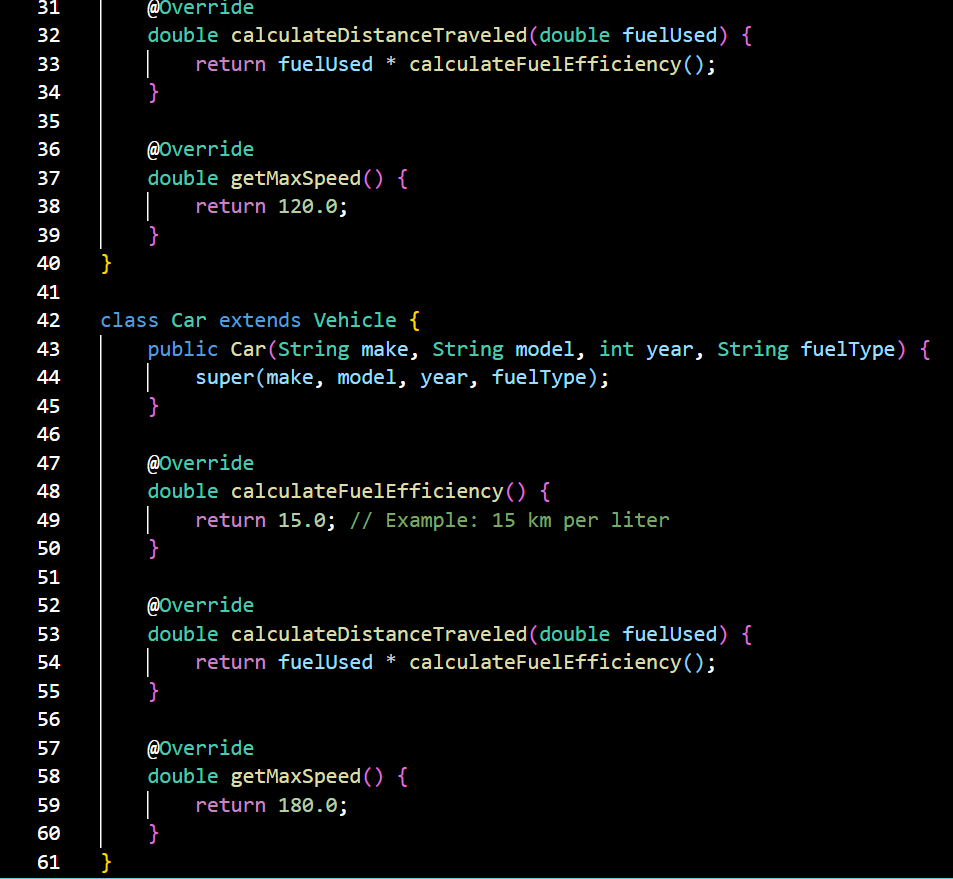
**Q6. Write a Java program to create a vehicle class hierarchy. The base class should be Vehicle, with subclasses Truck, Car and Motorcycle. Each subclass should have properties such as make, model, year, and fuel type. Implement methods for calculating fuel efficiency, distance traveled, and maximum speed.**

**AIM:** To 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. Implement methods for calculating fuel efficiency, distance traveled, and maximum speed.

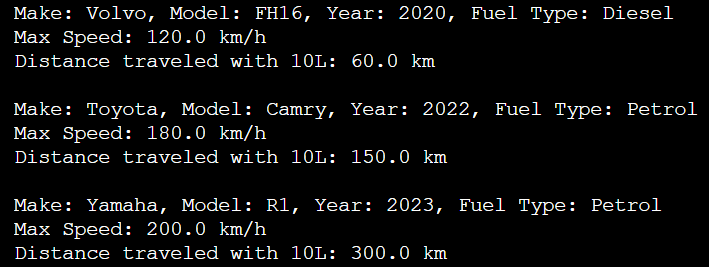
**ALGORITHM:**

1. Start
2. Create an abstract class vehicle
3. Create protected instances make, model, year, and fuel type
4. Create abstract methods for calculating fuel efficiency, distance traveled, and maximum speed
5. Create child classes Truck, Car and Motorcycle inheriting the abstract class
6. Override the abstract methods according to the different vehicles
7. End

**PROGRAM:** 



**OUTPUT:**



**RESULT:** We Created 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. Implement methods for calculating fuel efficiency, distance traveled, and maximum speed.

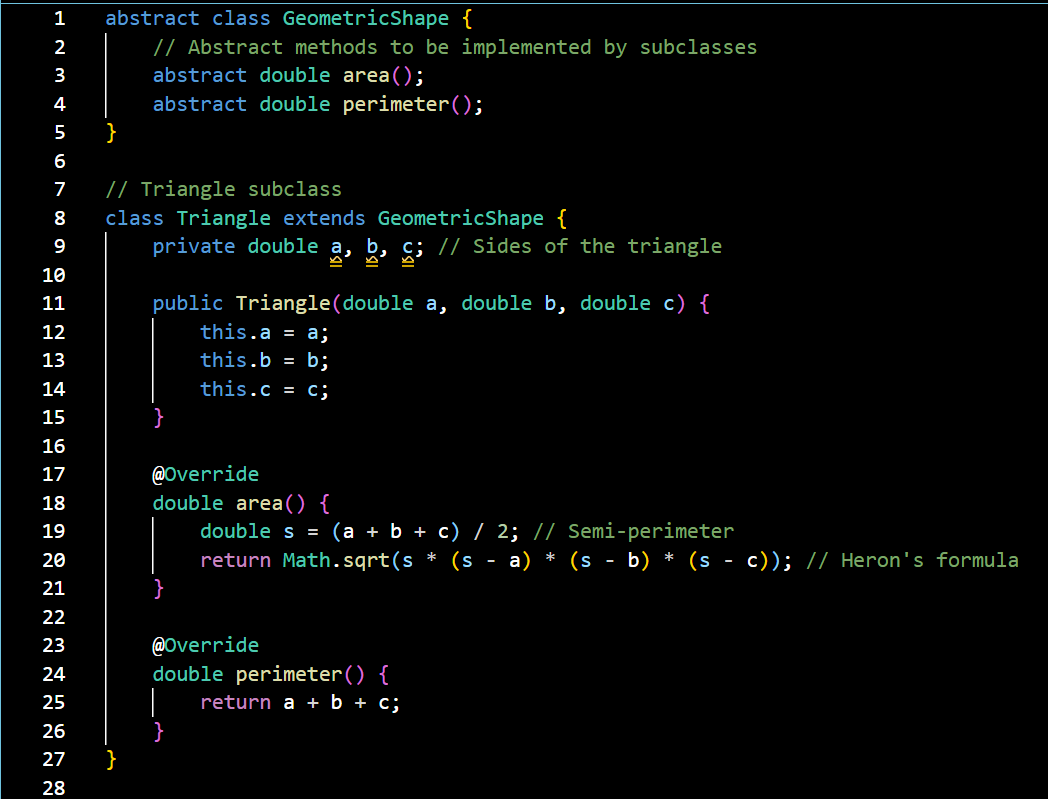
**Q7. Write a Java program to create an abstract class GeometricShape with abstract methods area() and perimeter(). Create subclasses Triangle and Square that extend the GeometricShape class and implement the respective methods to calculate the area and perimeter of each shape.**

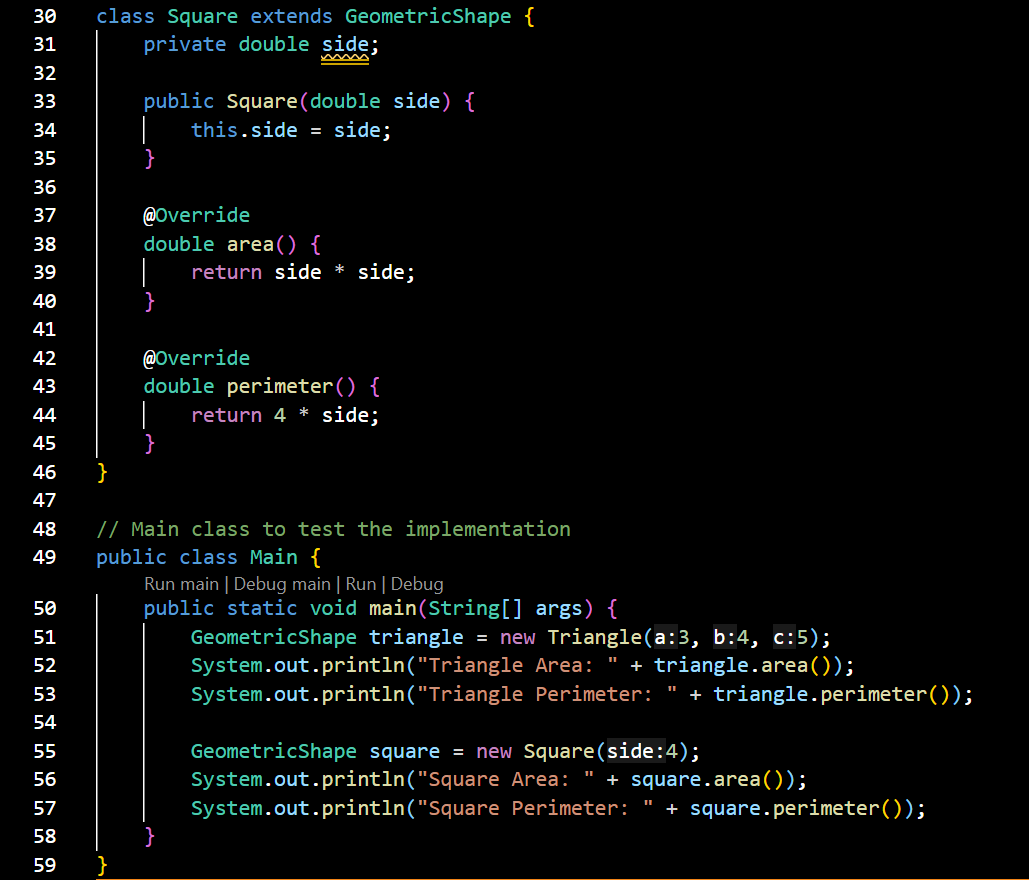
**AIM:** To Write a Java program to create an abstract class GeometricShape with abstract methods area() and perimeter(). Create subclasses Triangle and Square that extend the GeometricShape class and implement the respective methods to calculate the area and perimeter of each shape.

**ALGORITHM:**

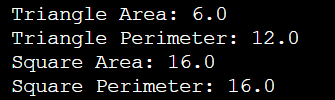
1. Start
2. Create an abstract class GeometricShape
3. Create abstract methods perimeter and area
4. Create sub classes triangle and circle extending the geometricshape class
5. Override the area and perimeter methods according to the shape
6. Call the methods and display the output
7. End

**PROGRAM:**

****

****

**OUTPUT:**



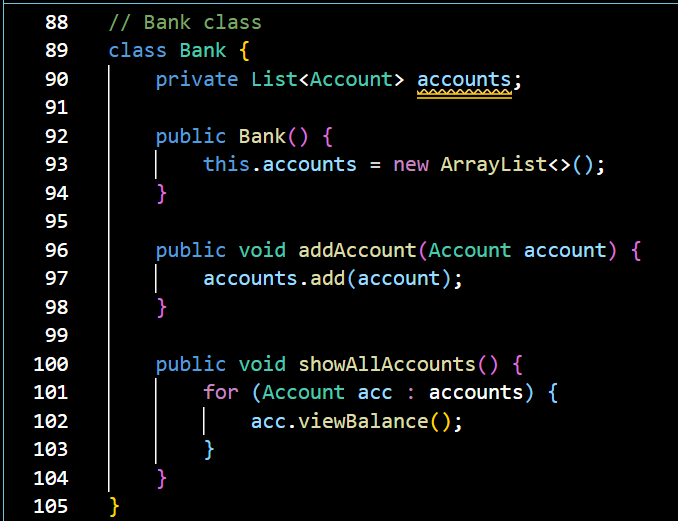
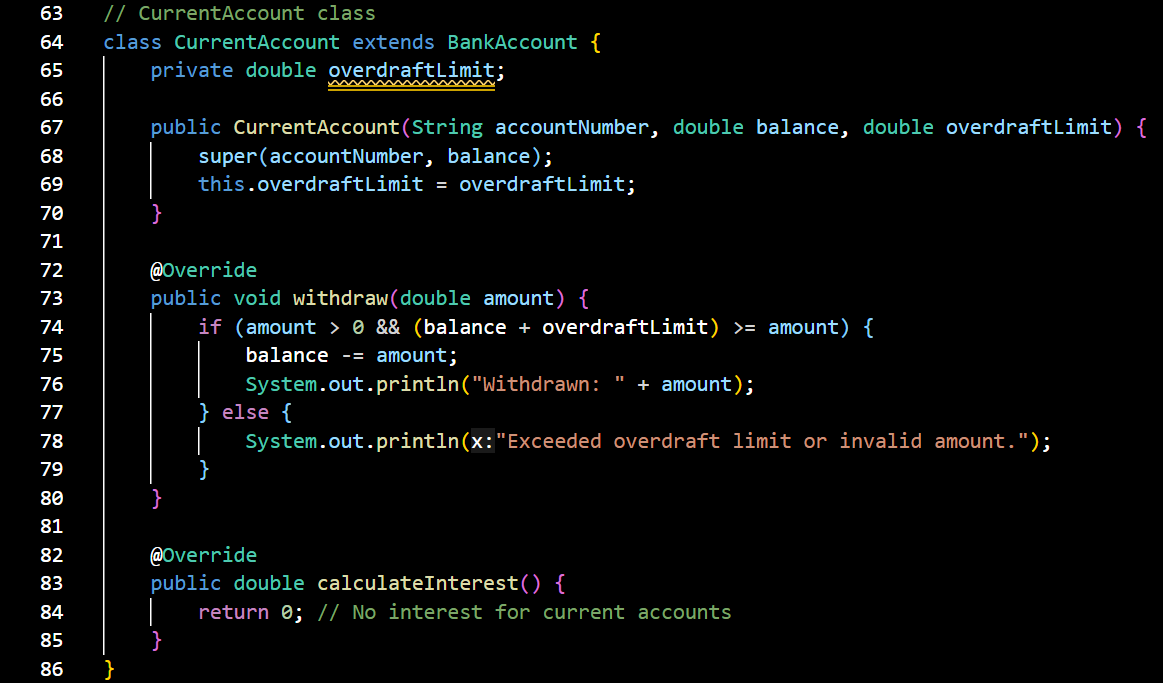
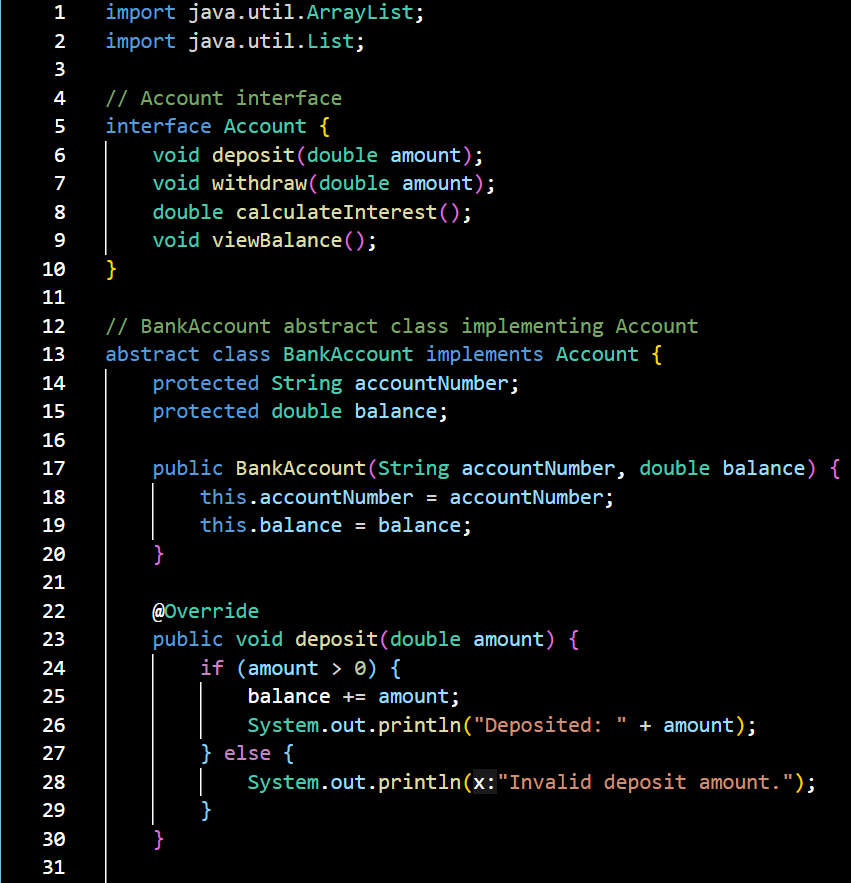
**RESULT:** We created a Java program to create an abstract class GeometricShape with abstract methods area() and perimeter(). Create subclasses Triangle and Square that extend the GeometricShape class and implement the respective methods to calculate the area and perimeter of each shape.

Q8. **Write a Java programming to create a banking system with three classes - Bank Account, SavingsAccount, and CurrentAccount. The bank should have a list of accounts and methods for adding them. Accounts should be an interface with methods to deposit, withdraw, calculate interest, and view balances. SavingsAccount and CurrentAccount should implement the Account interface and have their own unique methods.**

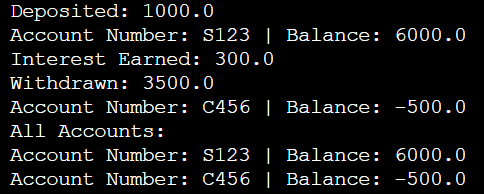
**AIM:** To Write a Java programming to create a banking system with three classes - Bank Account, SavingsAccount, and CurrentAccount. The bank should have a list of accounts and methods for adding them. Accounts should be an interface with methods to deposit, withdraw, calculate interest, and view balances. SavingsAccount and CurrentAccount should implement the Account interface and have their own unique methods.

**ALGORITHM:**

1. Start
2. Create an interface account
3. Create interface methods for deposit, withdraw, calculate interest, and view balances
4. Create an abstract class bankaccount
5. Inherit all the methods from the account interface and override them.
6. Create a savingsaccount class extend bankaccount
7. Inherit all the methods and override the calculate interest method according to it
8. Create a currentaccount class extending bankaccount
9. Inherit all the methods and override the withdrawal and calculate interest methods according to it
10. Use these class by creating their objects and using there methods
11. End

**PROGRAM:**

**OUTPUT:**



**RESULT:** we created a Java programming to create a banking system with three classes - Bank Account, SavingsAccount, and CurrentAccount. The bank should have a list of accounts and methods for adding them. Accounts should be an interface with methods to deposit, withdraw, calculate interest, and view balances. SavingsAccount and CurrentAccount should implement the Account interface and have their own unique methods.

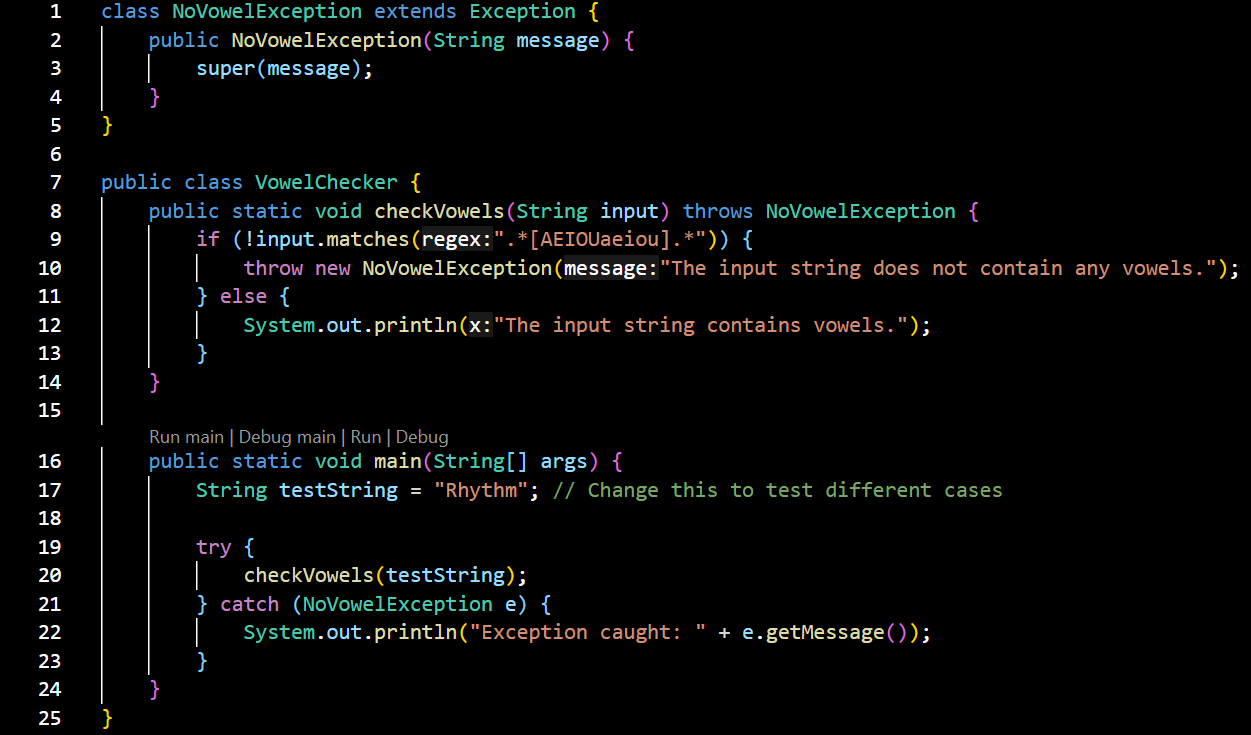
**Q9. Write a Java program to create a method that takes a string as input and throws an exception if the string does not contain vowels.**

**AIM:** To Write a Java program to create a method that takes a string as input and throws an exception if the string does not contain vowels.

**ALGORITHM:**

1. Start
2. Create an NoVowelException class extending exception class
3. Create a vowelchecker class and create a public function to check for vowels
4. Then create a main function and call in the method
5. End

**PROGRAM:**

****

**OUTPUT:**



**RESULT:** We Created a Java program to create a method that takes a string as input and throws an exception if the string does not contain vowels.

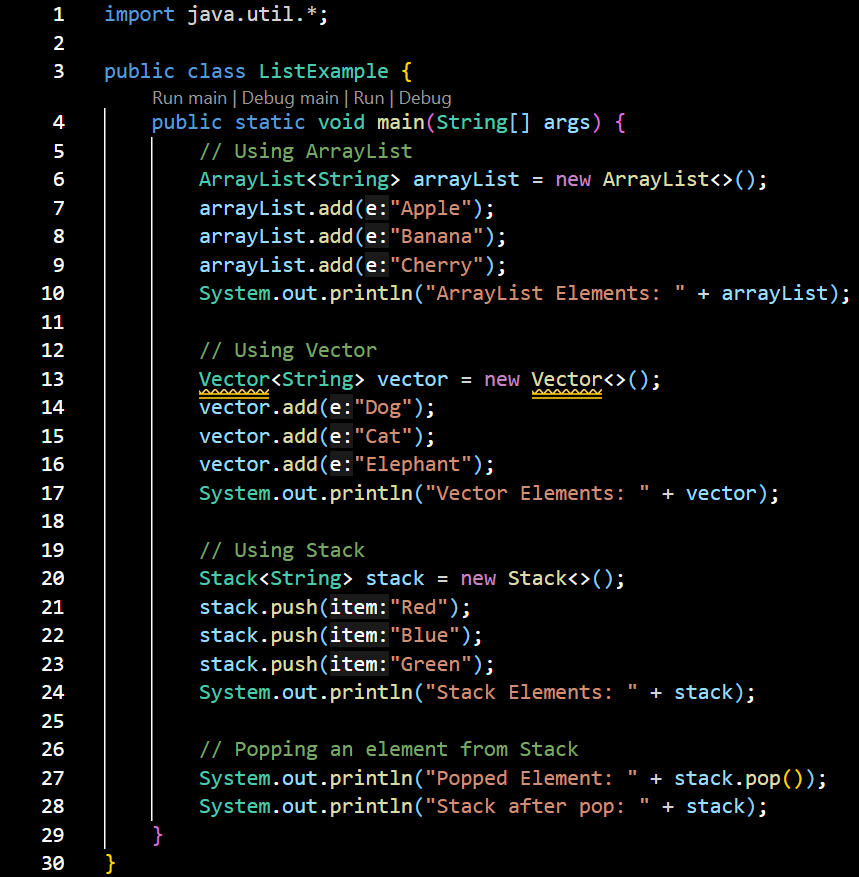
**Q10. Implement the list creation using ArrayList Class, Vector and Stack with example and output.**

**AIM:** To Implement the list creation using ArrayList Class, Vector and Stack with example and output.

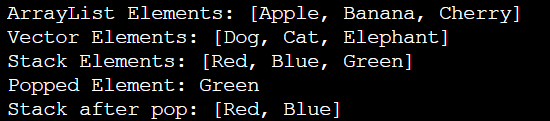
**ALGORITHM:**

1. Start
2. Import the lists
3. Create an object for each list
4. Implement the list methods on it
5. end

**PROGRAM:**

****

**OUTPUT:**

**RESULT:** We Implemented the list creation using ArrayList Class, Vector and Stack with example and output.

**Q11. Write a Java program that uses the CountDownLatch class to synchronize the start and finish of multiple threads(0-9 and 9-0).**

**AIM:** To Write a Java program that uses the CountDownLatch class to synchronize the start and finish of multiple threads(0-9 and 9-0).

**ALGORITHM:**

1. Start
2. Create a CountDownLatch named startSignal initialized to 1 (to start both threads).
3. Create another CountDownLatch named doneSignal initialized to 2 (to wait for both threads to finish).
4. Define Thread-1 (Count Up):

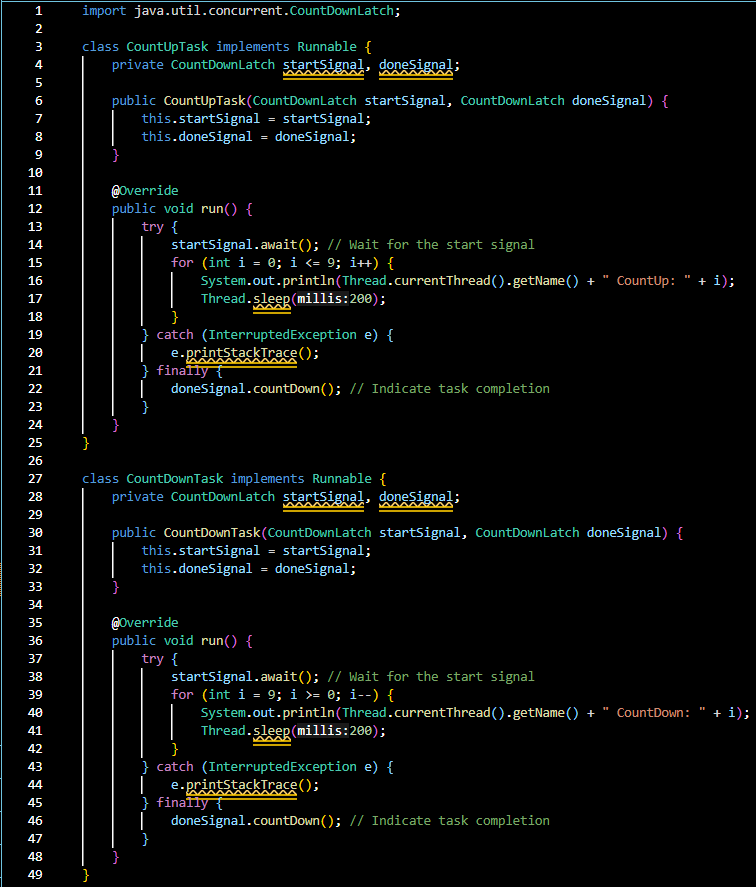
* Waits for startSignal.
* Counts from 0 to 9.
* Calls doneSignal.countDown() when done.

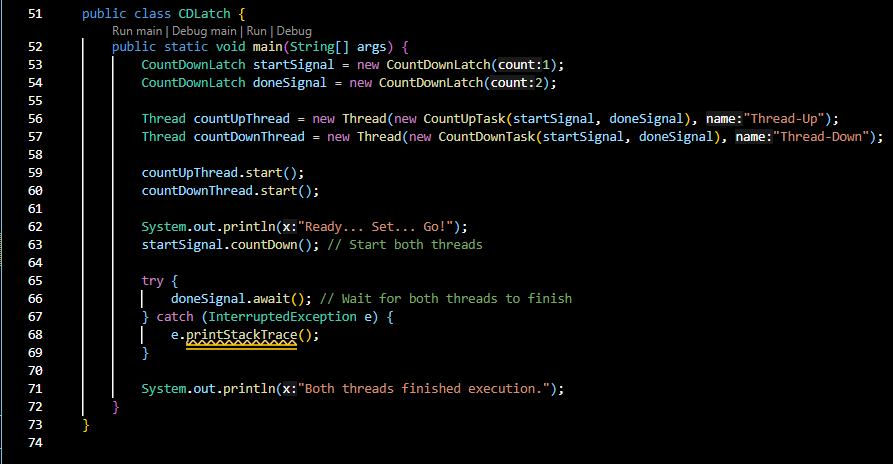
1. Define Thread-2 (Count Down):

* Waits for startSignal.
* Counts from 9 to 0.
* Calls doneSignal.countDown() when done.

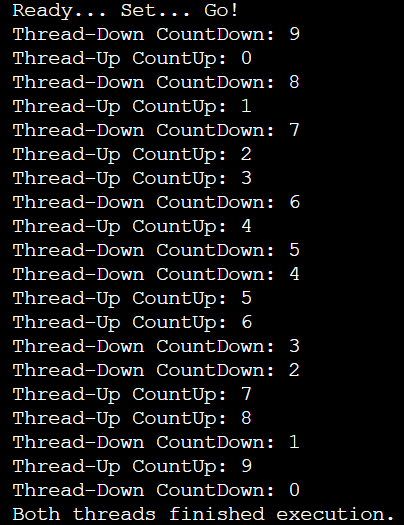
1. Start both threads.
2. Trigger start by calling startSignal.countDown().
3. Main thread waits for both threads to complete using doneSignal.await().
4. Print message when both threads are finished.
5. End

**PROGRAM:**





**OUTPUT:**



**RESULT:** We Created Write a Java program that uses the CountDownLatch class to synchronize the start and finish of multiple threads(0-9 and 9-0).

**Q12. Write a Java program to implement a concurrent web crawler that crawls multiple websites simultaneously using threads.**

**AIM:** To Write a Java program to implement a concurrent web crawler that crawls multiple websites simultaneously using threads.

**ALGORITHM:**

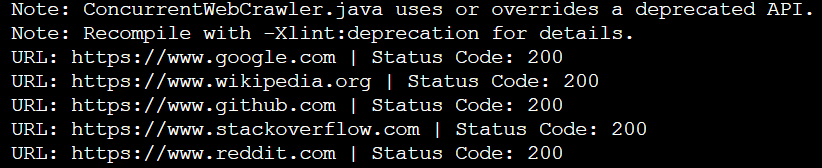
1. Start
2. Input: List of website URLs.
3. Initialize an ExecutorService with a fixed number of threads (thread pool).
4. For each URL:
   1. Create a Callable task that:
      1. Opens an HTTP connection to the URL.
      2. Gets the HTTP response status code.
      3. Returns the result as a string (URL and status).
5. Submit all tasks using invokeAll() to the executor.
6. Wait for all threads to complete and collect results using Future.
7. Print each result (URL and status code).
8. Shutdown the executor.
9. End

**PROGRAM:**





**OUTPUT:**

**RESULT:** We Created a Java program to implement a concurrent web crawler that crawls multiple websites simultaneously using threads.