

cod

Problem Statement

John is curious to know how many leap years he has lived through. He wants a program where he can input his birth year and the current year, and the program calculates the total number of leap years between them.

```
import java.util.Scanner;

class AgeCalculatorFunctions {

    private int birthYear;

    private int currentYear;

    // Constructor to initialize years

    public AgeCalculatorFunctions(int birthYear, int currentYear) {
        this.birthYear = birthYear;
        this.currentYear = currentYear;
    }

    // Method to calculate leap years

    public int calculateLeapYears() {
        int count = 0;
        for (int year = birthYear; year <= currentYear; year++) {
            if (isLeapYear(year)) {
                count++;
            }
        }
        return count;
    }
}
```

```
// Helper method to check leap year

private boolean isLeapYear(int year){

    if (year % 400 == 0) {

        return true;

    } else if (year % 100 == 0) {

        return false;

    } else {

        return year % 4 == 0;

    }

}

class Main {

    public static void main(String[] args){

        Scanner scanner = new Scanner(System.in);

        int birthYear = scanner.nextInt();

        int currentYear = scanner.nextInt();

        AgeCalculatorFunctions calculator = new AgeCalculatorFunctions(birthYear,
        currentYear);

        System.out.println(calculator.calculateLeapYears());

        scanner.close();

    }

}
```

Problem Statement

Linda is developing a program that computes the volume of either a cube or a pyramid based on user input. The user specifies the shape (1 for a cube, 2 for a pyramid) and provides the necessary dimensions like side length for a cube and base and height for a pyramid.

```
import java.util.Scanner;

class GeometryCalculator {

    // Overloaded method for cube

    public double calculateVolume(double side) {

        return side * side * side;
    }

    // Overloaded method for pyramid

    public double calculateVolume(double base, double height) {

        return (base * height) / 3.0;
    }
}

public class Main {

    public static void main(String[] args) {

        GeometryCalculator calculator = new GeometryCalculator();

        Scanner scanner = new Scanner(System.in);

        int shape = scanner.nextInt();
    }
}
```

```

if (shape == 1){

    double sideLength = scanner.nextDouble();

    double cubeVolume = calculator.calculateVolume(sideLength);

    System.out.printf("Volume of the cube: %.2f%n", cubeVolume);

}

else if (shape == 2){

    double baseArea = scanner.nextDouble();

    double height = scanner.nextDouble();

    double pyramidVolume = calculator.calculateVolume(baseArea, height);

    System.out.printf("Volume of the pyramid: %.2f%n", pyramidVolume);

}

else {

    System.out.println("Invalid");

}

scanner.close();

}

```

Problem Statement:

Imagine you're developing a Java class called `GeometricSequence` that offers constructors to calculate either the Nth term or the sum of the first N terms in a geometric sequence. The class provides two constructors:

```

import java.util.Scanner;

class GeometricSequence {

```

```

int a, r, N;
String op;

GeometricSequence(int a, int r, int N) { this(a, r, N, "term"); }

GeometricSequence(int a, int r, int N, String op) { this.a=a; this.r=r; this.N=N; this.op=op; }

double calculate() {
    switch (op.toLowerCase()) {
        case "term": return a * Math.pow(r, N-1);
        case "sum": return (r==1) ? a*N : a * (Math.pow(r, N)-1)/(r-1);
        default:
            System.out.println("Invalid operation specified.");
            return -1;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int a=sc.nextInt(), r=sc.nextInt(), N=sc.nextInt(); sc.nextLine();
        String op = sc.hasNextLine()? sc.nextLine().trim() : "";
        GeometricSequence gs = (op.isEmpty())? new GeometricSequence(a,r,N) : new
        GeometricSequence(a,r,N,op));
        double res = gs.calculate();
        if(res!=-1) System.out.println("Result: " + res);
    }
}

```

```
    sc.close();  
}  
}
```

Problem Statement

Sharon, a software developer, is working on a project to automate velocity calculations for various objects. She wants to create a class named VelocityCalculator with overloaded methods calculateVelocity to calculate the velocity. One method will accept distance in meters and time in seconds as integers, while another will accept distance and time as doubles.

```
import java.util.Scanner;  
  
class VelocityCalculator {  
  
    // Overloaded method for integer inputs  
  
    public static int calculateVelocity(int distance, int time) {  
  
        return distance / time; // integer division  
    }  
  
    // Overloaded method for double inputs  
  
    public static double calculateVelocity(double distance, double time) {  
  
        return distance / time; // double division  
    }  
  
}  
  
public class Main {  
  
    public static void main(String[] args) {  
  
        Scanner scanner = new Scanner(System.in);  
    }  
}
```

```

int distanceInt = scanner.nextInt();

int timeInt = scanner.nextInt();

double distanceDouble = scanner.nextDouble();

double timeDouble = scanner.nextDouble();

int velocityInt = VelocityCalculator.calculateVelocity(distanceInt, timeInt);

double velocityDouble = VelocityCalculator.calculateVelocity(distanceDouble,
timeDouble);

System.out.println("Velocity with integer inputs: " + velocityInt + " m/s");

System.out.printf("Velocity with double inputs: %.2f m/s", velocityDouble);

scanner.close();

}

}

```

LAB Ex

Problem Statement

One day, a creative architect received an unusual request from a client. The client needed a software program to calculate the area of rectangle-shaped garden plots. The client had the idea of using rectangular plots with various widths and heights to create unique gardens.

```

import java.util.Scanner;

class Rectangle {

    private double width;

```

```
private double height;

// Constructor

public Rectangle(double width, double height) {
    this.width = width;
    this.height = height;
}

// Method to calculate area

public double getArea() {
    return width * height;
}

}

class Main {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        double width = scanner.nextDouble();
        double height = scanner.nextDouble();

        Rectangle rectangle = new Rectangle(width, height);
        double area = rectangle.getArea();

        System.out.printf("%.2f", area);
        scanner.close();
    }
}
```

}