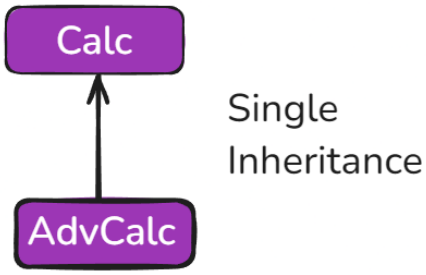
22-SIngle and Multilevel inheritance

Single and Multilevel Inheritance in Java

Inheritance is a powerful concept in Java that allows one class to inherit the properties and behaviors of another class. This concept is fundamental in Object-Oriented Programming (OOP) and helps in creating a hierarchical structure for classes.

Single Inheritance

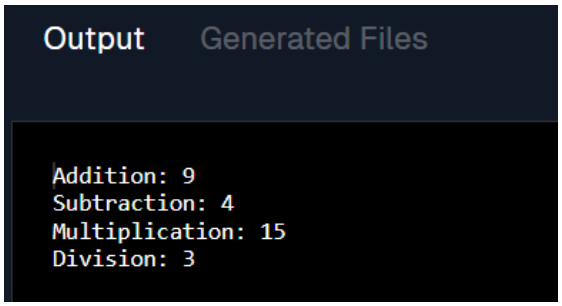
Definition: Single inheritance occurs when a class (child class) inherits from only one parent class. This allows the child class to access the fields and methods of the parent class.



Example: Let's look at a simple example of single inheritance:

```java
class Calc {
    public int add(int n1, int n2) {
        return n1 + n2;
    }

    public int sub(int n1, int n2) {
        return n1 - n2;
    }
}

class AdvCalc extends Calc {
    public int multi(int n1, int n2) {
        return n1 * n2;
    }

    public int div(int n1, int n2) {
        return n1 / n2;
    }
}

public class Demo {
    public static void main(String[] args) {
        AdvCalc obj = new AdvCalc();
        int r1 = obj.add(4, 5);
        int r2 = obj.sub(7, 3);
        int r3 = obj.multi(5, 3);
        int r4 = obj.div(15, 4);

        System.out.println("Addition: " + r1);
        System.out.println("Subtraction: " + r2);
        System.out.println("Multiplication: " + r3);
        System.out.println("Division: " + r4);
    }
}

```

Output:



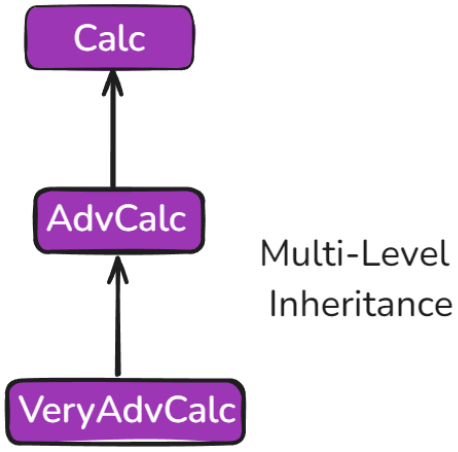
Explanation:

- This example demonstrates how AdvCalc, the child class, inherits basic arithmetic operations from Calc, the parent class.
- AdvCalc adds its own methods (multi and div) while reusing the methods from Calc (add and sub).

This approach prevents code duplication and makes the code easier to maintain and extend.

Multilevel Inheritance

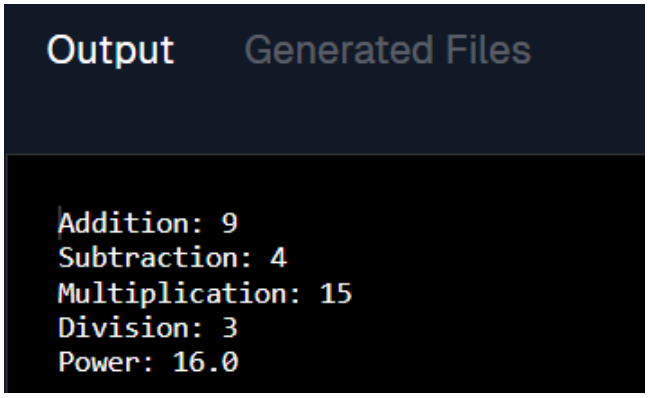
Definition: Multilevel inheritance is a scenario where a class inherits from a child class, making a chain of inheritance. This allows a subclass to inherit properties and methods from more than one level of parent classes.



Example: Consider the following example to understand multilevel inheritance:

```java
class Calc {
    public int add(int n1, int n2) {
        return n1 + n2;
    }

    public int sub(int n1, int n2) {
        return n1 - n2;
    }
}

class AdvCalc extends Calc {
    public int multi(int n1, int n2) {
        return n1 * n2;
    }

    public int div(int n1, int n2) {
        return n1 / n2;
    }
}

class VeryAdvCalc extends AdvCalc {
    public double power(int n1, int n2) {
        return Math.pow(n1, n2);
    }
}

public class Demo {
    public static void main(String[] args) {
        VeryAdvCalc obj = new VeryAdvCalc();
        int r1 = obj.add(4, 5);
        int r2 = obj.sub(7, 3);
        int r3 = obj.multi(5, 3);
        int r4 = obj.div(15, 4);
        double r5 = obj.power(4, 2);

        System.out.println("Addition: " + r1);
        System.out.println("Subtraction: " + r2);
        System.out.println("Multiplication: " + r3);
        System.out.println("Division: " + r4);
        System.out.println("Power: " + r5);
    }
```

Output:



Explanation:

- In this example, VeryAdvCalc inherits from AdvCalc, which in turn inherits from Calc. This creates a multilevel inheritance structure.

- VeryAdvCalc has access to methods from both its parent (AdvCalc) and grandparent (Calc) classes, along with its own method power.