28-Final Keyword

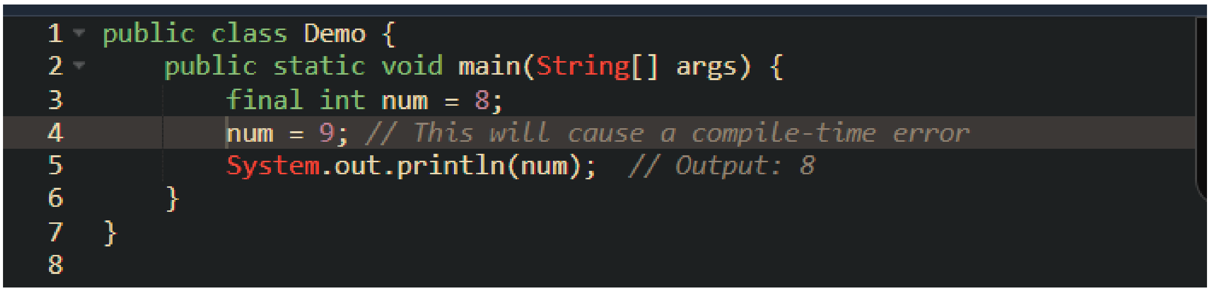
Final Keyword in Java

The final keyword in Java is used to restrict the modification of variables, methods, and classes. It plays a crucial role in defining constants and preventing inheritance or method overriding. Let's explore how final can be used in different contexts:

1. Final Variable

A variable declared with the final keyword cannot be modified once it is initialized. This makes the variable effectively a constant.

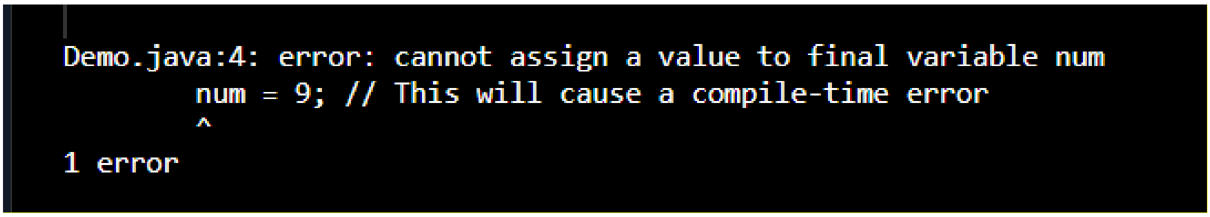
Explanation:



```java
public class Demo {
    public static void main(String[] args) {
        final int num = 8;
        num = 9; // This will cause a compile-time error
        System.out.println(num);  // Output: 8
    }
}

```

Output:



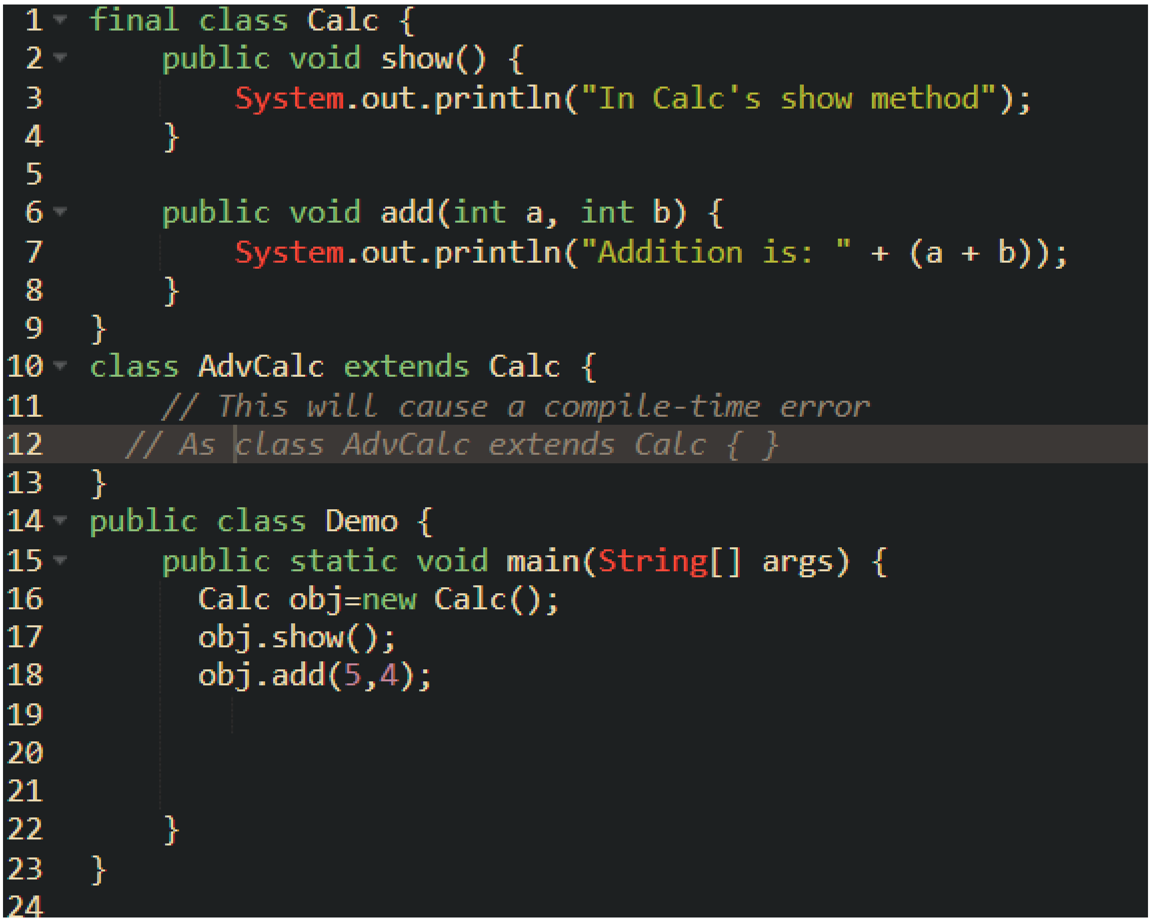
```
Demo.java:4: error: cannot assign a value to final variable num
        num = 9; // This will cause a compile-time error
        ^
1 error
```

Explanation: Once num is declared as final, any attempt to change its value will result in a compilation error.

2. Final Class

A class declared as final cannot be extended (inherited). This is useful when you want to prevent other classes from inheriting and modifying its behavior.

Example:



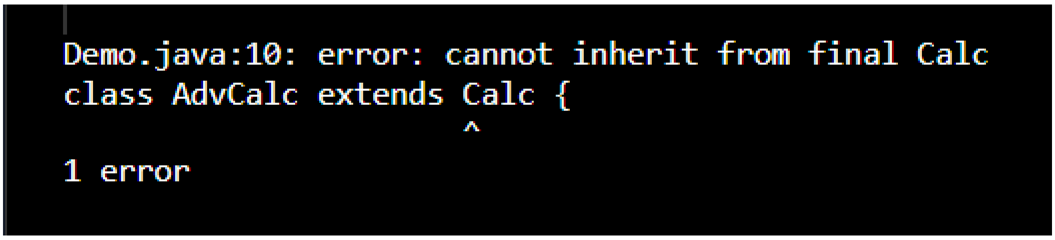
```java
final class Calc {
    public void show() {
        System.out.println("In Calc's show method");
    }

    public void add(int a, int b) {
        System.out.println("Addition is: " + (a + b));
    }
}
class AdvCalc extends Calc {
    // This will cause a compile-time error
  // As class AdvCalc extends Calc { }
}
public class Demo {
    public static void main(String[] args) {
      Calc obj=new Calc();
      obj.show();
      obj.add(5,4);



    }
}
```

Output:



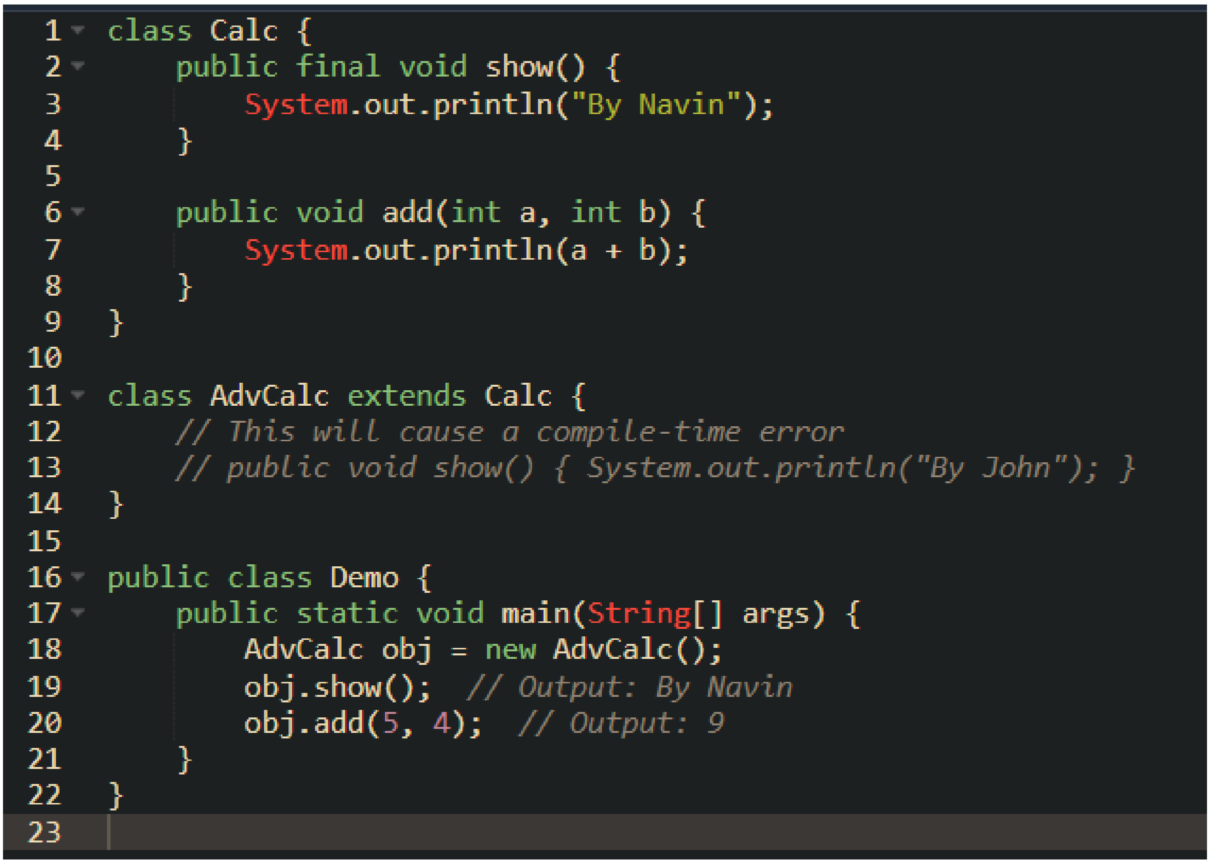
```
Demo.java:10: error: cannot inherit from final Calc
class AdvCalc extends Calc {
                      ^
1 error
```

Explanation: The final keyword prevents the Calc class from being extended by any other class.

3. Final Method

A method declared with the final keyword cannot be overridden by subclasses. This ensures that the method's implementation remains unchanged in all derived classes.

Example:



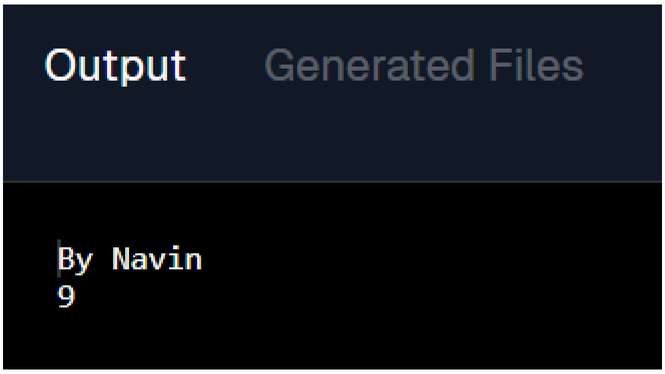
```
class Calc {
    public final void show() {
        System.out.println("By Navin");
    }

    public void add(int a, int b) {
        System.out.println(a + b);
    }
}

class AdvCalc extends Calc {
    // This will cause a compile-time error
    // public void show() { System.out.println("By John"); }
}

public class Demo {
    public static void main(String[] args) {
        AdvCalc obj = new AdvCalc();
        obj.show();  // Output: By Navin
        obj.add(5, 4);  // Output: 9
    }
}
```

Output



```
By Navin
9
```

Explanation: The show method in Calc is marked as final, so it cannot be overridden by AdvCalc.

Characteristics of final Keyword

- **Final Variables**: Once initialized, their value cannot be changed. This is useful for constants.

- **Final Methods**: These cannot be overridden, ensuring that the implementation stays consistent across all subclasses.
- **Final Classes**: These cannot be extended, which is useful when a class is meant to be used as-is without modification.
- **Initialization**: Final variables must be initialized when declared or within the constructor, ensuring they are assigned exactly once.
- **Performance**: Using final can improve performance as the compiler can optimize the code knowing that the value or method won't change.
- **Security**: The final keyword helps in protecting data and methods from being modified by malicious code.

Conclusion

The final keyword in Java is a powerful tool for creating constants, preventing inheritance, and ensuring methods are not overridden. It provides security, performance, and clarity in the codebase by restricting unwanted modifications.

FAQs about the final Keyword in Java

1. **What is a final method in Java?**

   A final method is a method that cannot be overridden by subclasses. It ensures that the method's behavior remains unchanged across the inheritance hierarchy.

2. **Is a final method inherited?**

   Yes, a final method is inherited by the subclass, but it cannot be overridden or modified.

3. **Can a final class be instantiated?**

   Yes, a final class can be instantiated, just like any other class. However, it cannot be extended.

4. **What is the difference between final and static keywords in Java?**
   - Final Keyword: Used to declare variables, methods, or classes as unmodifiable.
   - Static Keyword: Used to declare class members (variables and methods) that belong to the class itself, not to instances of the class.

5. **Can we declare a String variable as final?**

   Yes, you can declare a String variable as final, which means that once it is initialized, it cannot be changed.