

1.final

final是一个关键字，可以用于类、方法和变量，表示不可改变的特性。

- **final变量：**

- 一旦赋值后，不能再改变。
- 必须在声明时或构造函数中初始化。

```
public class MyClass {  
    public final int CONSTANT = 10;  
  
    public MyClass() {  
        // CONSTANT = 20; // 编译错误，不能重新赋值  
    }  
}
```

- **final方法：**

- 不能被子类重写。

```
public class ParentClass {  
    public final void display() {  
        System.out.println("This is a final method.");  
    }  
}  
  
public class ChildClass extends ParentClass {  
    // public void display() { // 编译错误，不能重写  
    //     System.out.println("Cannot override final method.");  
    // }  
}
```

- **final类：**

- 不能被继承。

```
public final class FinalClass {  
    // 类内容  
}  
  
// public class SubClass extends FinalClass { // 编译错误，不能继承  
// }
```

2.finally

finally是一个用于异常处理的关键字，表示一个代码块，它总是会被执行，无论是否发生异常。通常用于释放资源等清理工作。

- **finally代码块：**

- 总是会执行，即使在try块中有return语句。

```

public class MyClass {
    public static void main(String[] args) {
        try {
            System.out.println("In try block");
            // return; // 即使有 return, finally 也会执行
        } catch (Exception e) {
            System.out.println("In catch block");
        } finally {
            System.out.println("In finally block");
        }
    }
}

```

3.finalize

finalize是一个方法，用于对象被垃圾回收器回收之前的清理操作。在Object类中定义，子类可以重写它来执行特定的清理操作。

- **finalize方法：**
 - 在垃圾回收器确定没有对该对象的更多引用时调用。
 - 由于垃圾回收机制的不确定性，不推荐依赖finalize方法进行重要的清理工作。

```

public class MyClass {
    @Override
    protected void finalize() throws Throwable {
        try {
            System.out.println("Finalize method called");
            // 清理代码
        } finally {
            super.finalize();
        }
    }
}

```

总结

- **final**：用于声明常量、不可重写的方法和不可继承的类。
- **finally**：用于异常处理，确保某些代码总是会执行。
- **finalize**：用于对象被垃圾回收之前的清理操作，但由于不确定性，不推荐依赖。