1. 内部类

（一）定义

在一个类中定义一个类，如：在类A中定义一个类B，则该类B就称为内部类。

格式：

 class 外部类名{

     修饰符 class 内部类名{

     }

}

（二）特点

1. 内部类可以直接访问外部类的成员，包括私有的外部类成员变量

2. 外部类要访问内部类的成员，必须在外部类中创建内部类对象，通过内部类对象来访问内部类的成员。

（三）分类

1. 成员内部类

2. 局部内部类

3. 匿名内部类

1. 成员内部类

（一）外部类中的位置

内部类定义的位置和外部类的成员变量是一个位置。

（二）创建成员内部类格式

格式：外部类名 .内部类名 对象名 = 外部类对象.内部类对象;

举例： Outer.Inner oi = new Outer().new Inner();

（三）成员内部类的使用方法

当我们定义一个内部类时，一般是不想外界来访问，因此内部类的定义应该是private，同时，我们也要留一些退路，给外界一个调用的接口，在外部类中定义调用方法来创建内部类对象并调用。

（四）示例

class Outer {

    private int num = 10;

    private class Inner {

        public void show() {

            System.out.println(num);

        }

    }

    public void method() {

        Inner i = new Inner();

        i.show();

    }

}

public class InnerDemo {

    public static void main(String[] args) {

//Outer.Inner oi = new Outer().new Inner();

//oi.show();

        Outer o = new Outer();

        o.method();

    }

}

1. 局部内部类
2. 位置

定义在外部类的成员方法中的内部类。

1. 创建局部内部类格式

定义在外部类的成员方法中的局部内部类，外部类是无法直接使用的，需要在外部类的成员方法中创建内部类对象并使用。

1. 特点

局部内部类可以直接访问外部类的成员，也可以访问该局部内部类所在的外部类成员方法中的局部变量。

1. 示例

class Outer {

    private int num = 10;

    public void method() {

        int num2 = 20;

        class Inner {

            public void show() {

                System.out.println(num);

                System.out.println(num2);

            }

        }

        Inner i = new Inner();

        i.show();

    }

}

public class OuterDemo {

    public static void main(String[] args) {

        Outer o = new Outer();

        o.method();

    }

}

1. 匿名内部类
2. 定义

本质是一个继承了一个类或者接口的子类匿名对象

1. 前提

存在一个类或者接口，这里的类可以是具体类也可以是抽象类

1. 定义的格式

New 类名/接口名(){

重写方法

}

举例：

new Inter(){

    @Override

    public void method(){}

}

1. 匿名内部类的调用方法

匿名内部类本质是一个子类匿名对象，因此可以直接通过该对象调用成员方法或者成员对象。

如：

interface Inter{

    void method();

}

class Test{

    public static void main(String[] args){

        new Inter(){

            @Override

            public void method(){

                System.out.println("我是匿名内部类");

            }

        }.method(); // 直接调用方法

    }

}

1. 匿名内部类的应用时机

当某个方法需要抽象类或者接口的子类对象时，我们可以传递一个匿名内部类过去，简化一般的方法。即传递的是一个子类匿名对象，重写方法以实现不同的功能。