

2024 年微光后端招新 *Java* 05

by 杨皓涵

2024.10.12

Task 1. 对象和类

问：

1. 请你为这个 `Person` 类添加构造方法实现复制对象，并在题解附上你的 `Person` 类代码。你的构造方法用到 `this` 关键字了吗？请说说它的作用。
2. 在主类的 `main` 方法中创建 `Person` 类的一个对象，并给它的字段赋值（可以用构造函数，也可以用引用变量）。说说对象和类的关系。
3. 学习访问修饰符，为你的 `Person` 类的字段和方法添加你认为合适的访问修饰符。尝试在不同的位置（当前类，相同包的其它类，包的外部等）访问这些字段和方法，并总结出各种访问修饰符的限制范围。

答：

1. 用到了 `this`。`this` 关键字在构造函数中用于引用当前对象的地址，可以避免与传入的参数名称冲突。
2. 创建对象并调用构造器赋值：

```
Person person = new Person("yang", 18,  
1); //假设1表示男性
```

类是相似对象的共有抽象模板，对象是类的实例。

3. 代码中部分属性和方法修改如下:

```
private String name;  
private int age;  
private int sex;  
protected void finalize() throws Throwable {}  
public void eat() {}  
void sleep() {}  
private void dadoudou() {}
```

访问修饰符的限制范围:

private: 只能被当前类访问。

默认: 可以被同一个包内的类访问。

protected: 可以被同一个包内的类以及所有子类访问。

public: 可以被任何类访问。

Task 2. 类中的变量和方法

问：

4. 为你的 Person 类创建一个静态方法命名为 count，该方法返回 Person 类存在多少个对象。注意，count 方法不应该统计到已经被销毁的 Person 对象。

答：

4. 首先声明静态变量：

```
private static int count = 0;
```

在所有构造器执行代码中均加入 count++ 实现每创建一个对象统计数均会对应加一：

```
public Person() {
    count++;
}

public Person(String name, int age, int sex) {
    this.name = name;
    this.age = age;
    this.sex = sex;
    count++;
}

public Person(Person other) {
    this.name = other.name;
    this.age = other.age;
    this.sex = other.sex;
    count++;
}
```

重写 finalize 方法达到不计入已销毁对象的目的：

```
@Override
protected void finalize() throws Throwable {
    count--;
    super.finalize(); // 调用父类的finalize方法
}

// 注意：finalize方法在java9以上的版本中已被弃用。
```

声明静态方法 getCount 返回对象个数：

```
public static int getCount() {  
    return count;  
}
```

结束。

PS: 以笔者目前的能力似乎并没有在 jdk8 以后版本中实现销毁对象个数自减的功能，或许只有更加深入地学习才可以解决这一难题了。