

后端

Java05

task1

嗯~感觉后面的题比配置环境和学习git怎么用简单

(///ω\\)

不过依旧有一些不太懂的东西，后文我会提到的！！

以下是要求的代码，（在github中也上传了，命名为java05答题代码）

```
package com.ISEKAI.tool;

public class PersonMain {
    public class Person{
        private String name;
        private int age;
        private int sex;
        private Person(String name,int age,int sex) {
            this.name=name;
            this.age=age;
            this.sex=sex;
            //说起来以前都是this->age这样写的
        }
        private void eat(){
            System.out.println(name+"学长让我吃东西(*^_^*)");
        }

        private void sleep(){
            System.out.println(name+"我不要睡觉，招新题还没做完/(T_o_T)/~~");
        }

        private void dadoudou(){
            System.out.println(name+"/"+age+"/"+sex+"管你是谁，不准打豆豆(ノ=皿=)ノ");
        }

        public void xugai(String newname,Person origin){
            //我想在输入的变量里使用&origin引用，但不行，不知道java是不需要还是需要其它语法，/(T_o_T)/~~
            origin.name=newname;
        }

        public
    }

    public void main(String[] args) {
        Person myclass=new Person("XY",18,1);
    }
}
```

下一个问题：

- **default** (默认) : 在同一包内可见, 不使用任何修饰符。使用对象: 类、接口、变量、方法。
- **private** : 在同一类内可见。使用对象: 变量、方法。 **注意: 不能修饰类 (外部类)**
- **public** : 对所有类可见。使用对象: 类、接口、变量、方法
- **protected** : 对同一包内的类和所有子类可见。使用对象: 变量、方法。 **注意: 不能修饰类 (外部类) 。**

目前我有用到的也就只有private和public了, 心得体会谈不上, 主要是现在需要实现的功能没有那么多方面的考虑, 感觉甚至可以全用public(不是)

task2

搜索的时候看到了这句话

你可以通过类名来调用静态方法, 而不需要创建类的实例

好方便...

"菜鸟"如是说道:

静态方法与实例方法的区别:

关联性: 静态方法与类本身相关, 而实例方法与类的实例相关。

调用方式: 静态方法通过类名调用, 而实例方法需要通过对象实例来调用。

访问权限: 静态方法可以访问类的静态成员, 但不能访问非静态成员 (实例成员)。实例方法可以访问类的静态和非静态成员。

内部引用: 静态方法中不能使用this关键字, 因为它没有当前对象的引用。实例方法可以使用this来引用当前对象。

生命周期: 静态方法在类加载时初始化, 而实例方法在对象创建时初始化。

简单来说:

静态方法是与类本身相关的方法, 通常用于执行通用操作或访问静态成员。

实例方法则是与类的实例相关的方法, 可以访问和修改实例的状态。

然后, 对于要求的代码:

我在这里卡了很久, 因为想要实现每次新生成一个类, 都将类中的静态变量num++, 但苦于尝试多种方法无果 (最终发现是忘记在class类声明时加上static导致无法在main中使用类的初始化函数)

然后我求助下ai, 很尴尬, 因为ai帮我解决了一些问题, 但是它给出的代码没有实现我们所需要的全部功能, ai给出的代码如下(这个我就不传github了, 学长也不用看着个代码)

```
package com.ISEKAI;
public class CESHI {
    class Person {
        private static int countOfObjects = 0;
        public Person() {
            countOfObjects++;
        }
        public static int count() {
            return countOfObjects;
        }
    }
}
```

```

    }

    public static void main(String[] args) {
        Person person1 = new Person();
        Person person2 = new Person();
        Person person3 = new Person();
        System.out.println("当前存在的 Person 对象数量: " + Person.count());
        // 模拟销毁一个对象（假设没有实际的销毁操作，只是减少计数的示意）
        // 在实际应用中，如果对象被垃圾回收，计数可能会自动调整，但这里只是简单示意不统计
        已“销毁”对象
        person2 = null;
        System.out.println("模拟销毁一个对象后，当前存在的 Person 对象数量: " +
        Person.count());
    }
}

```

这其中有两个错误，一个是没在（class Person）前加上static导致主函数中的初始化函数使用报错，以及将person2置空后计数器任然是3。

我想这个原因是，我们应该定义一个**析构函数**（搜了下java的析构是怎么写的）来删除并把计数器自减而不是直接置空，所以最后我对我的代码修改了一下。

（两task的代码为同一个，已上传至github，文件名为Java05答题代码）

好像篇幅长了，节约学长时间，就此打住。