**学习总结**

姓名\_董金伟\_ 专业\_计算计科学与技术\_ 班级2班\_\_\_ 日期\_\_2022/1/8\_

1. 当天所学习主要内容（做一总结）

**一、方法存在的必要**

简化代码，有利于后期的维护修改。

**二、定义方法的语法**

访问权限 修饰符 返回类型 方法名（形参表）{ 方法体 //一段代码 }

变量名---变量名---保存数据

方法名---方法名---完成一定功能

**三、无参无返回值的方法**

void 方法名(){ ​ 方法体; }

方法名();

方法名：和变量名规则相同。见名知意。

**四、带参方法**

void 方法名(形参表){ 方法体; }

方法名(实参表);

形参表：类型名 形参名，类型名 形参名，。。。

实参表：表达式，表达式。。。

形参名本质就是变量名 值传递

**五、方法返回值**

返回类型是返回值的类型

return 表达式;

return：

1. 终止方法的执行。
2. 将表达式的值返回。

范例：

1. 打印直角三角形时传递行数为负数时的判断。
2. 计算阶乘
3. 判断素数

**六、方法总结**

1. 清晰明确的知道方法完成的功能
2. 名字：见名知意
3. 形参：是否需要提高一些信息
4. 返回类型：是否需要返回一个值
5. 方法体：口语描述算法，写出伪代码，编写程序

注意事项：

* 方法的嵌套。
* 参数是值传递
* 形参类型为：类型名... 形参名，本质：类型名[] 形参名
  + 实参：
    - 可以不传实参，可以传一个或多个实参。
  + 局限性：
    - 形参中最多只能有一个这样的形参。
    - 这个格式的形参只能作为最后一个形参。
* 形参：args

**七、递归方法**

语法：在方法体中直接或间接的调用自身方法。用于实现循环。

适用：

1. 小规模问题解决了，大规模问题自然就解决了。
2. 小规模问题的问题解决和大规模问题的解决方法相同。
3. 当规模小到一定程度，问题文件自然解决。

典型案例：汉诺塔

1. 学习中所存在的问题及解决办法

问题：1. 参数传递时，无法快速判断出结果。

解决：1. 在刚开始不了解参数传递时，可以画内存图来帮助理解。

3．学习收获及建议

学会了方法的定义与使用。

在定义方法时，命名规则与变量的命名规则一样。方法的参数传递分为形参和实参，形参可以理解为值的类型，实参可以理解为值。

使用方法时，比较特殊和难以理解的使用有递归方法，递归就时方法自己调用自己，实现循环。