

第一组

问题一：

```
MerchandiseV2 m = new MerchandiseV2();
```

```
public MerchandiseV2 gift;
```

```
m.gift = giftNoodle;
```

其中gift的数据类型具体是什么？

讨论：

gift确实是m的成员变量，这个gift变量的类型是MerchandiseV2类的引用，相当于在商品对象m里存了另一个商品对象的引用。

➡ 组合

- 组合就是 A 类的对象是 B 类的成员变量。相当于 A 类对象是 B 类对象的一个变量，A 类中的所有功能，B 类都可以通过 A 类对象的调用来实现。
- 组合体现的是整体与部分、拥有的关系，即 has - a 的关系

问题二：

在使用检测输入的过程中都需要对输入的类型进行限制，比如说如何实现同时监测 int 类型和字符串兼容的情况，倘若用重载，其中会不会产生因为优先级问题导致输入不可读的情况？

➡ 参见例程

问题三：

静态变量/方法是否能通过一个实例化的对象来访问/调用？

➔ 通过实例变量也可以调用静态方法，但这只是编译器自动帮我们把实例改写成类名而已。

通常情况下，通过实例变量访问静态字段和静态方法，会得到一个编译警告。

第二组

问题描述：

Java 栈内存、堆内存、方法区 对象和引用的关系。

讨论记录：

栈内存：

1. 基本数据类型（一共有八种，char, byte, short, int, long, float, double [boolean](#)）直接分配在“栈”；
2. 用于存储局部变量，当数据使用完，所占空间会自动释放。

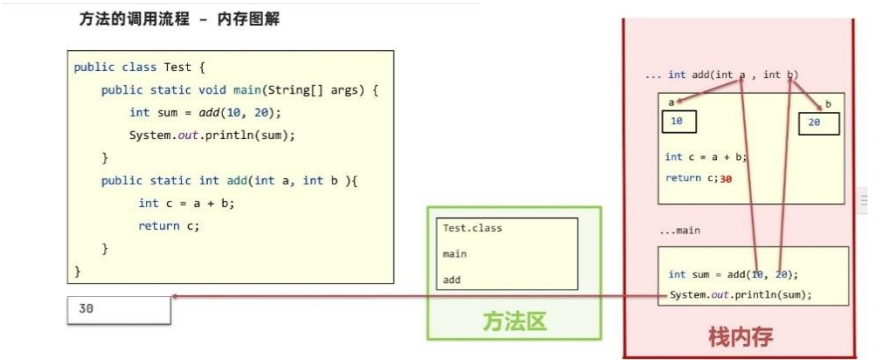
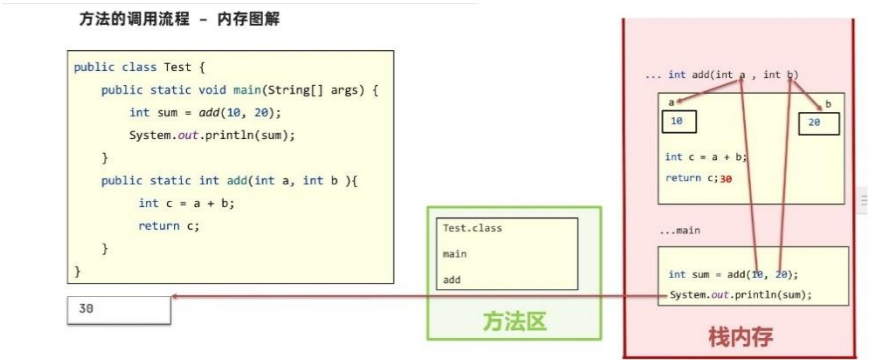
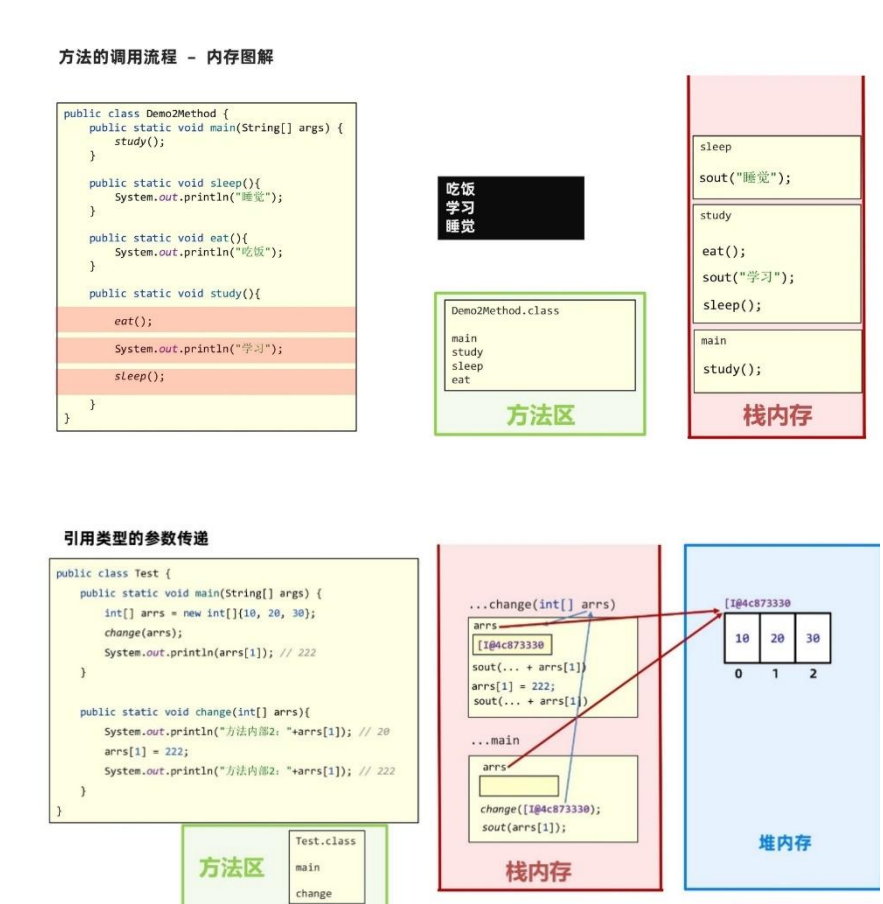
堆内存：

- 1、数组和对象，通过 new 建立的实例都存放在堆内存中；
- 2、每一个实体都有内存地址；
- 3、实体中的变量都有默认初始化值；
- 4、实体不在被使用，会在不确定的时间内被垃圾回收器回收。

方法区：

1. 保存着被加载过的每一个类的信息，这些信息由[类加载](#)器在加载类的时候，从类的源文件中抽取出来；
2. static 变量信息也保存在方法区中；

如下 3 张图为搜到的可能有帮助的图，



问题描述:

创建对象的步骤

讨论结果:

1. 判断对象对应的类是否加载、链接、初始化。
2. 为对象分配内存
首先计算对象所需的内存的大小，对象所需内存的大小在类加载完成后便可完全确定。
3. 处理并发安全问题
4. 默认初始化分配到的空间。也就是对对象的字段赋 0 值，或者 null 值。
5. 设置对象的对象头。
6. 执行 init 方法进行初始化。
执行显示初始化、代码块中的初始化、构造器中的初始化。

问题描述:

在创建新的文件夹以及挪动部分 java 类后出现了“无法解析符号”的错误。但是程序仍然能正常编译执行。试图重启 IntelliJ 仍然报错。

解决方案:

这可能是由于 IntelliJ 代码重构的 bug，选择 IntelliJ 左上角文件-清除缓存，重启 IntelliJ 后错误消失。

目前待解决的问题:

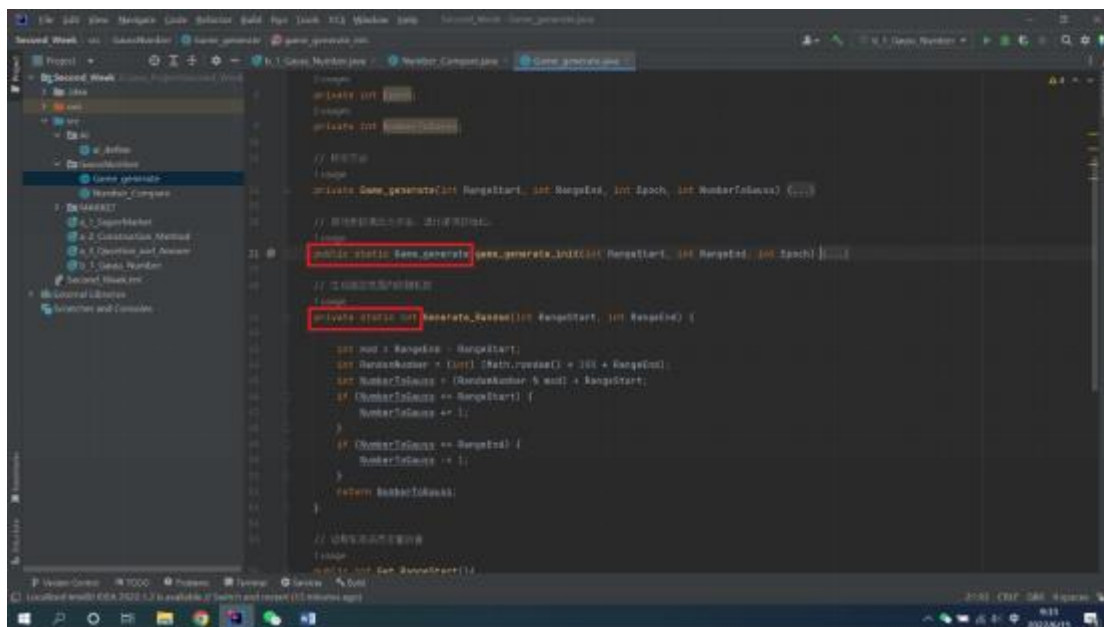
想了解下创建方法和对象以及对象的方法，在栈内存和堆内存如何进行调用。

第三组

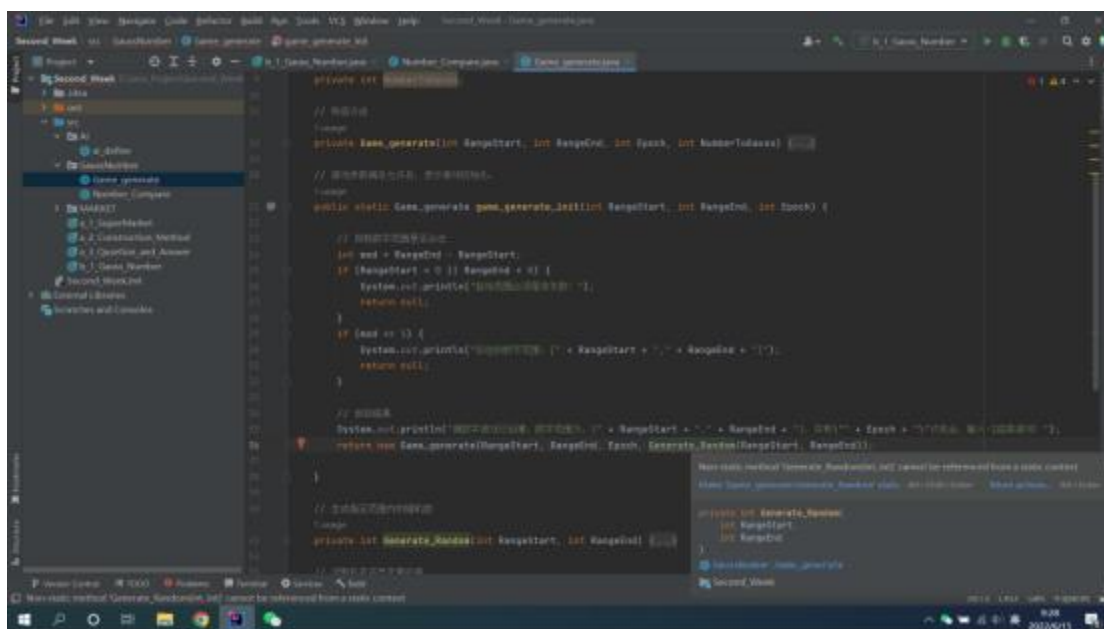
我们组的问题为

- 1、在方法前面必须使用 static 关键字才能正确运行，未添加 static 关键字时会抛出错误。

图一为添加 static 关键字运行



图二为未添加 static 关键字时运行:



2、IDEA 和 github 的连接问题：

①：使用 SSH 地址时，前面步骤都正确，最后一步连接远程仓库时报错，连接失败，一直提示没有权限。

②：用 https 的 github 链接，一直 push 不上去，弹出一个网页需要验证，但是打不开。

第四组

1、 权限修饰符，

课程中对权限修饰符提到了 public、private、和缺省三种情况，实际上还有 protected 如表所示：

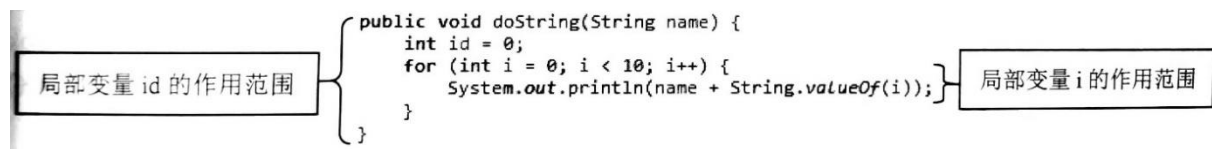
访问包位置	类 修 饰 符		
	private	protected	public
本类	可见	可见	可见
同包其他类或子类	不可见	可见	可见
其他包的类或子类	不可见	不可见	可见

缺省时，权限和 protected 一样，只能同包里可见。

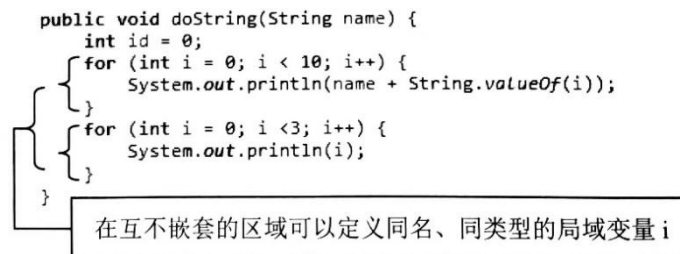
2、 局部变量的有效范围

在编程中对于局部变量的作用范围容易混淆，尤其是当变量繁多时，需要理清思路。

局部变量的有效范围为变量的作用域，即从声明开始到变量结束。



在互不嵌套的区域可以重复定义同名同类型的局部变量，而在嵌套中不可以重复定义。



```

public void doString(String name) {
    int id = 0;
    for (int i = 0; i < 10; i++) {
        System.out.println(name + String.valueOf(i));
    }
    for (int i = 0; i < 3; i++) {
        System.out.println(i);
        int id = 7;
    }
}

```

在嵌套区域内重复定义局部变量

不可以直接在作用范围外直接使用局部变量，没有作用范围外声明局部变量的代码。

3、 类、对象和引用的关系



真正的对象是“new People()”，为了方便开发者保存和调用对象，创建了 people 类型，名叫 tom 的引用。实际上 tom 只是内存地址，用于标记对象在内存中位置。需要注意的是，如果两个引用指向同一个对象，那么修改其中一个引用，另一个引用也会变化，因为引用指向的是同一个内存地址

4、 构成重载的方法

- 1、参数类型不同
- 2、参数顺序不同
- 3、参数个数不同

