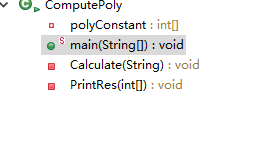
第四次博客作业

1. 三次作业程序结构分析

（1）多项式加减



类：1个。

属性：1个。

方法：3个。

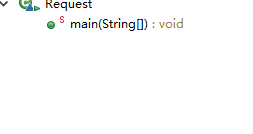
CopuutePoly类：

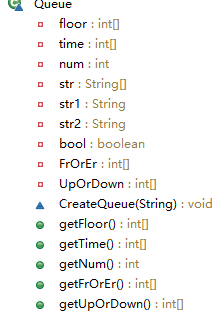
main方法：36行

Calculate方法：46行

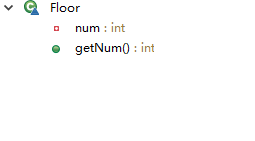
PrintRes方法：15行

（2）傻瓜电梯调度

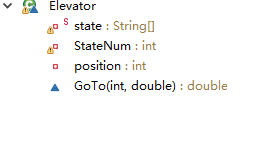




**- str**

**- num**

**- floor[]**

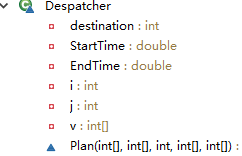
**- time[]**

**- num**

**- FrOrEr[]**

**- UpOrDown[] - destination**

**- StartTime**



**- EndTime**

类：5个

Request类（50行）：

属性：1个

方法：1个（main 42行）：

Queue类（90行）：

属性：9个

方法：6个（CreateQueue 38行，还有5个get、set方法）

Despatcher类（40行）：

属性：6个

方法：1个（Plan 30行）

Elevator类（36行）：

属性：3个

方法：1个（GoTo 30行）

Floor类（11行）：

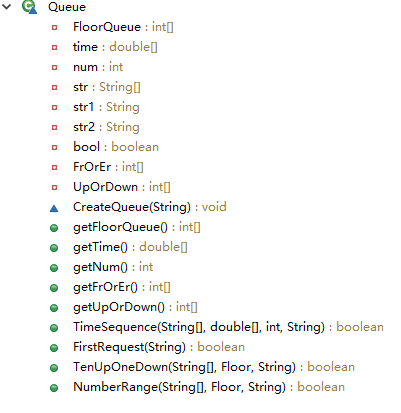
属性：1个

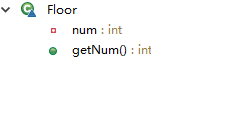
方法：1个（一个get方法）

（3）ALS单电梯调度

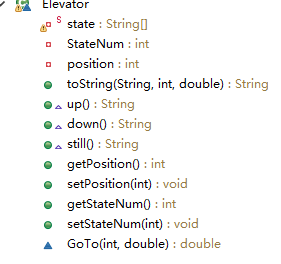


**- str**





**- num**

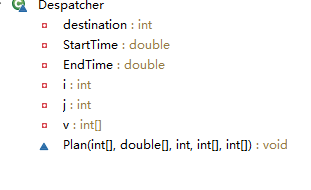


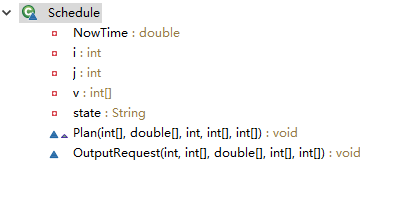
**- floor[]**

**- time[] - s**

**- num**  **- destination**

**- FrOrEr[] - EndTime**

**- UpOrDown[]**

 **Schedule继承自spatcher**

类：6个

Request类（50行）：

属性：1个

方法：1个（main 42行）：

Queue类（134行）：

属性：9个

方法：10个（CreateQueue 38行，TimeSequeuce 11行，FirstRequest 9行，TenUpOneDown 10行，NumberRange 22行，还有5个get、set方法）CreateQueue调用了TimeSequeuce、FirstRequest、TenUpOneDown、NumberRange。

Despatcher类（40行）：

属性：6个

方法：1个（Plan 30行）

Elevator类（36行）：

属性：3个

方法：8个（GoTo 30行，up 3行，down 3行，still 3行，还有四个get、set方法）

Floor类（11行）：

属性：1个

方法：1个（一个get方法）

Schedule类（229行）：

属性：5个

方法：2个（Plan 200行，OutPutRequest 10行）

总评：第一次作业我只写了一个类多个方法，这可能导致方法之间的联系过于紧密，将来对类进行修改时可能会比较麻烦。

第三次作业中，我在重写Schedule类中父类Plan方法时，没有规划好而将许多功能合并到一起，导致Plan方法竟然有200行，而且其中有大量重复使用的代码，而我也没有将其封装为一个单独的方法。

三次作业中，我没有通过第二次的三个对于int类型范围限制的测试点，因为我在第二次作业中限定输入最大为8位数，这确实是有偷懒的嫌疑，主要是害怕被人测试时超过int类型。除此之外我还因为粗心而错了两个公测。

三次作业我的优点应该在勤写注释和命名规范。只写代码不写注释是非常坑人的行为，这一点我深有体会，如果没有注释，哪怕对于自己很久之前写的代码也需要很长的时间才能看懂，更何况是别人来看。除此之外，掌握一种命名规范可以很好的帮助阅读者理清代码逻辑，而不用四处找一个变量的出现位置。

在发现他人bug方面，主要是阅读代码，根据代码来针对性地测试，尤其是对其中的数组是否越界、变量是否超过大小，以及自己在编写时曾经犯过的错误，进行重点分析。

最后讲讲我测别人程序时发现的几个问题，首先有同学写了上千行代码，结果只有一个.java文件，还有些同学只写代码不写注释。这些都是小事，但是会给他人和自己的阅读造成较大的困难，希望有这些习惯的同学能改正。