

第十二次博客作业

一、规格化设计的大致发展历史

在计算机刚发明时，程序员仅仅设计一些小型的程序，此时程序员单打独斗，很少合作。后来程序越来越大，分成大量模块，为了规定模块的规格，便于程序员之间协作，于是产生了规格化设计。规格化设计为程序员提供了清晰的规范，约束了代码的要求，方便模块化测试。

二、好的规格

过程规格：

```
public synchronized void PassWriteFile(LinkedList<String>list)
    /*@REQUIRES:list!=null
    @
    @
    @MODIFIES:passenger.txt
    @
    @EFFECTS:将 list 中的内容输入到 passenger.txt
    @
    @THREAD_REQUIRES:none
    @
    @THREAD_EFFECTS:\locked()
    @
    */
public synchronized void TaxiWriteFile(LinkedList<String>list)
    /*@REQUIRES:list!=null
    @
    @
    @MODIFIES:taxi.txt
    @
    @EFFECTS:将 list 中的内容输入到 taxi.txt
    @
    @THREAD_REQUIRES:none
    @
    @THREAD_EFFECTS:\locked()
    @
    */
```

```

public synchronized void WriteFile(ServiceMessage se)
    /*
    @REQUIRES:s!=null
    @
    @
    @MODIFIES: Servicemessage.txt
    @
    @EFFECTS: 将 s 输入到 ServiceMessage.txt 中
    @THREAD_REQUIRES: none
    @
    @THREAD_EFFECTS: \locked()
    @
    */

```

数据规格:

```

public class Taxi extends Thread
    //表示对象:Matrix matrix;MyPoint point;int index;int
status;int credit;int count;int direction;OutPut
output;LinkedList <String> StrList=new
LinkedList<String>();TaxiGUI gui;Light light;LinkedList<MyPoint>
way1=new LinkedList<MyPoint>();LinkedList<MyPoint> way2=new
LinkedList<MyPoint>();
    //
    //抽象函数: AF(c)=(Matrix matrix,int index,OutPut
output,TaxiGUI gui,Light light)where
matrix==c.matrix,index==c.index,output==c.output,gui==c.gui,li
ght==c.light
    //
    //                                不                                变
式 :c.matrix!=null&&0<=c.index<=99&&c.output!=null&&c.gui!=null&
&c.light!=null
public class Taxi extends Thread
    //表示对象:Matrix matrix;MyPoint point;int index;int
status;int credit;int count;int direction;OutPut
output;LinkedList <String> StrList=new
LinkedList<String>();TaxiGUI gui;Light light;LinkedList<MyPoint>
way1=new LinkedList<MyPoint>();LinkedList<MyPoint> way2=new
LinkedList<MyPoint>();
    //
    //抽象函数: AF(c)=(Matrix matrix,int index,OutPut
output,TaxiGUI gui,Light light)where
matrix==c.matrix,index==c.index,output==c.output,gui==c.gui,li
ght==c.light
    //

```

```

//                                不                                变
式 :c.matrix!=null&&0<=c.index<=99&&c.output!=null&&c.gui!=null&
&c.light!=null
public class OutPut
    //表示对象:none
    //
    //抽象函数: AF(c)=()
    //
    //不变式:none

```

三、Bug 与规格的关系

有不少 bug 在写代码时没有发现，等到写方法的 JSF 时，才发现写的代码并不严谨，如果先写 JSF，可能就不会出现这样的 bug。