****

**软件测试技术第一次实验报告**



**学 院 软件学院**

**专 业 软件工程**

**年 级 大三**

**姓 名 窦淑洁**

**2019年 3 月 15 日**

# 软件测试技术第一次实验报告

1. 需求分析（描述具体需求）

**Tasks:**

1. Install Junit(4.12), Hamcrest(1.3) with Eclipse
2. Install Eclemma with Eclipse
3. Write a java program for the triangle problem and test the program with Junit.
   1. Description of triangle problem:

There is one 50 yuan, one 20 yuan, two 5 yuan bills and three 1 yuan coins in your pocket. Write a program to find out whether you can take out a given number (x) yuan.

**Requirements for the experiment:**

1. Finish the tasks above individually.
2. Check in your java code and junit test program to github and send the URL to [tjuscsst@qq.com](mailto:tjuscsst@qq.com)
3. Please send your experiment to [tjuscsst@qq.com](mailto:tjuscsst@qq.com) , the following information should be included in your report:
   1. The brief description that you install junit, hamcrest and eclemma.
   2. The test result and coverage report (print screen) of your tests on triangle problem.
4. 概要设计（简单描述设计思路，配合UML图）

Description of triangle problem:

方法一：动态规划解决问题，dp[j] = max(dp[j-k\*c[i]]+k,dp[j])实现零钱兑换解决方案。

方法二：使用深度优先搜索暴力枚举解决方案，若存在零钱兑换解决方案，即可返回true

测试文件设置8条测试用例用来完全覆盖测试程序的全部边和节点。

1. 详细设计（详细描述具体如何实现，附代码及说明）

Description of triangle problem:

方法一：

**public** **static** **boolean** triangle(**int** x){

**int**[] cost = {50,20,5,1};

**int**[] num = {1,1,2,3};

**int**[] dp = **new** **int**[1110];

**int** len = cost.length;

**for**(**int** i=1;i< 1110;i++)

dp[i] = 0x3f3f3f3f;

**for**(**int** i = 0;i < 4;i++){

**for**(**int** j = x;j >= cost[i];j--){

**for**(**int** k = 0;k <= num[i];k++){

**if**(j >= k\*cost[i]){

//System.out.println(j);

dp[j] = Math.*min*(dp[j-k\*cost[i]]+k, dp[j]);

}

}

}

}

**if**(dp[x] == 0x3f3f3f3f){

**return** **false**;

}**else**{

System.***out***.println(dp[x]);

**return** **true**;

}

}

方法二：

**public** **static** **void** dfs(**int** floor,**int** index,**int** value,**int** x){

**if**(value == x){

*flag* = **true**;

**return** ;

}**else** **if**(index == 7||value > x){

**return** ;

}

**int** begin = index;

**if**(floor == -1){

**for**(**int** i = 0;i < 7;i++){

*dfs*(0,i,*cost\_tt*[i],x);

}

}**else**{

**for**(**int** i = index+1;i < 7;i++){

*dfs*(0,i,*cost\_tt*[i]+value,x);

}

}

}

**public** **static** **boolean** ss(**int** x){

*flag* = **false**;

*dfs*(-1,0,0,x);

//System.out.println(flag);

**if**(*flag* == **true**){

**return** **true**;

}**else**{

**return** **false**;

}

}

设计测试用例：

@Test

**public** **void** testAdd(){

*assertEquals*(**true**,**new** homework1().*triangle*(1));

*assertEquals*(**true**,**new** homework1().*triangle*(5));

*assertEquals*(**false**,**new** homework1().*triangle*(100));

*assertEquals*(**false**,**new** homework1().*triangle*(120));

*assertEquals*(**false**,**new** homework1().*triangle*(124));

*assertEquals*(**true**,**new** homework1().*triangle*(75));

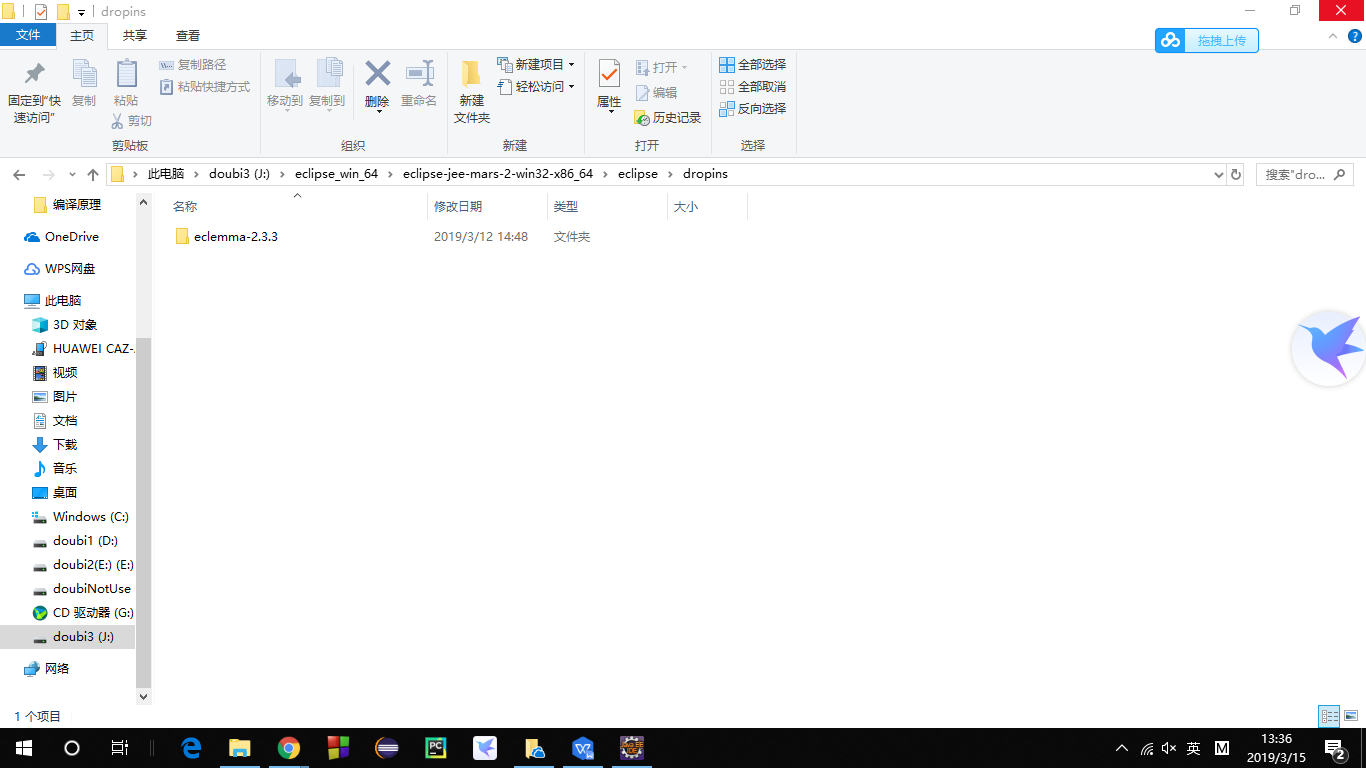
*assertEquals*(**false**,**new** homework1().*triangle*(79));

*assertEquals*(**false**,**new** homework1().*triangle*(98));

}

1. 调试分析（在实验过程中遇到的问题以及如何解决）

将eclemma压缩包下载下来后解压到Eclipse本地目录的dropins下，重启Eclipse



重启Eclipse后eclemma插件图标未出现

进入eclemma[文件夹删除](https://www.baidu.com/s?wd=%E6%96%87%E4%BB%B6%E5%A4%B9%E5%88%A0%E9%99%A4&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "https://blog.csdn.net/weixin_38946532/article/details/_blank)其中的META-INF文件夹，再次重启Eclipse

1. 测试结果（描述输入和输出）

*assertEquals*(**true**,**new** homework1().*triangle*(1));

*assertEquals*(**true**,**new** homework1().*triangle*(5));

*assertEquals*(**false**,**new** homework1().*triangle*(100));

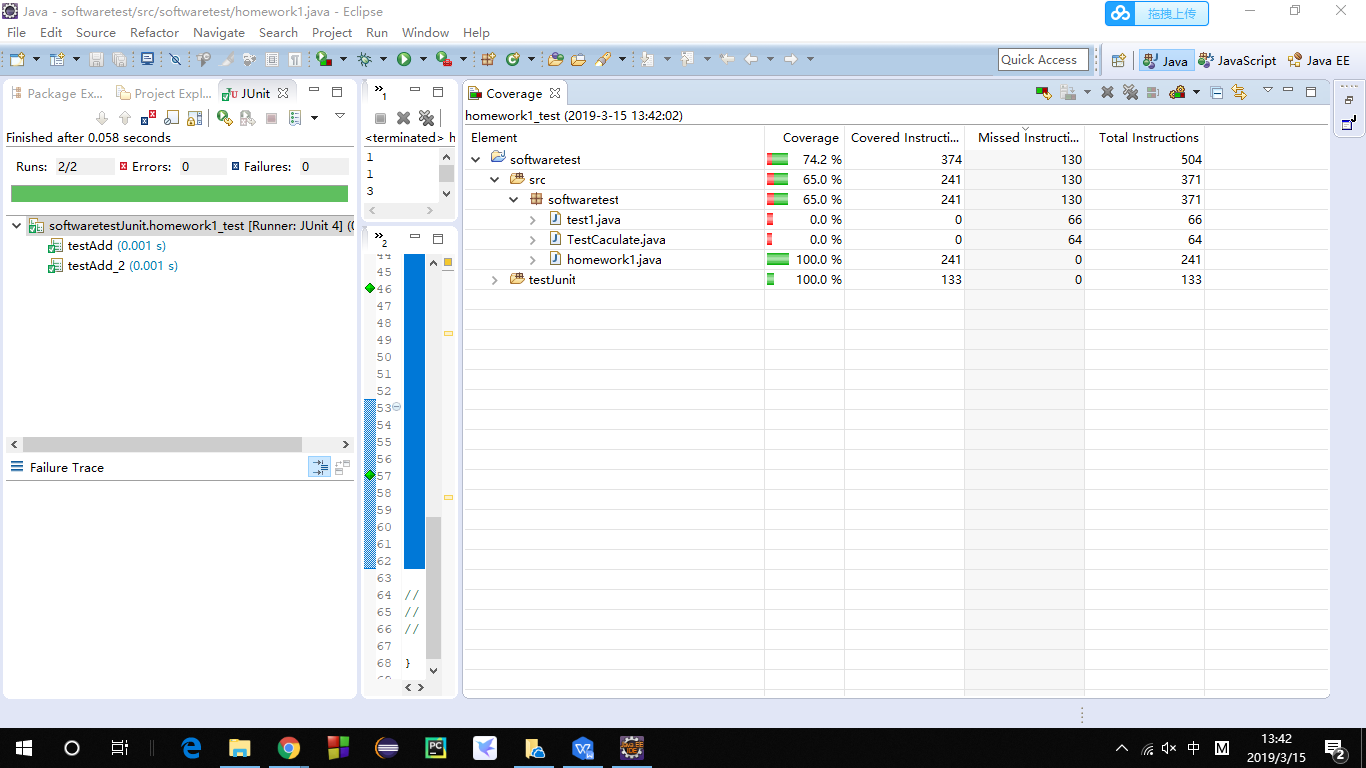
*assertEquals*(**false**,**new** homework1().*triangle*(120));

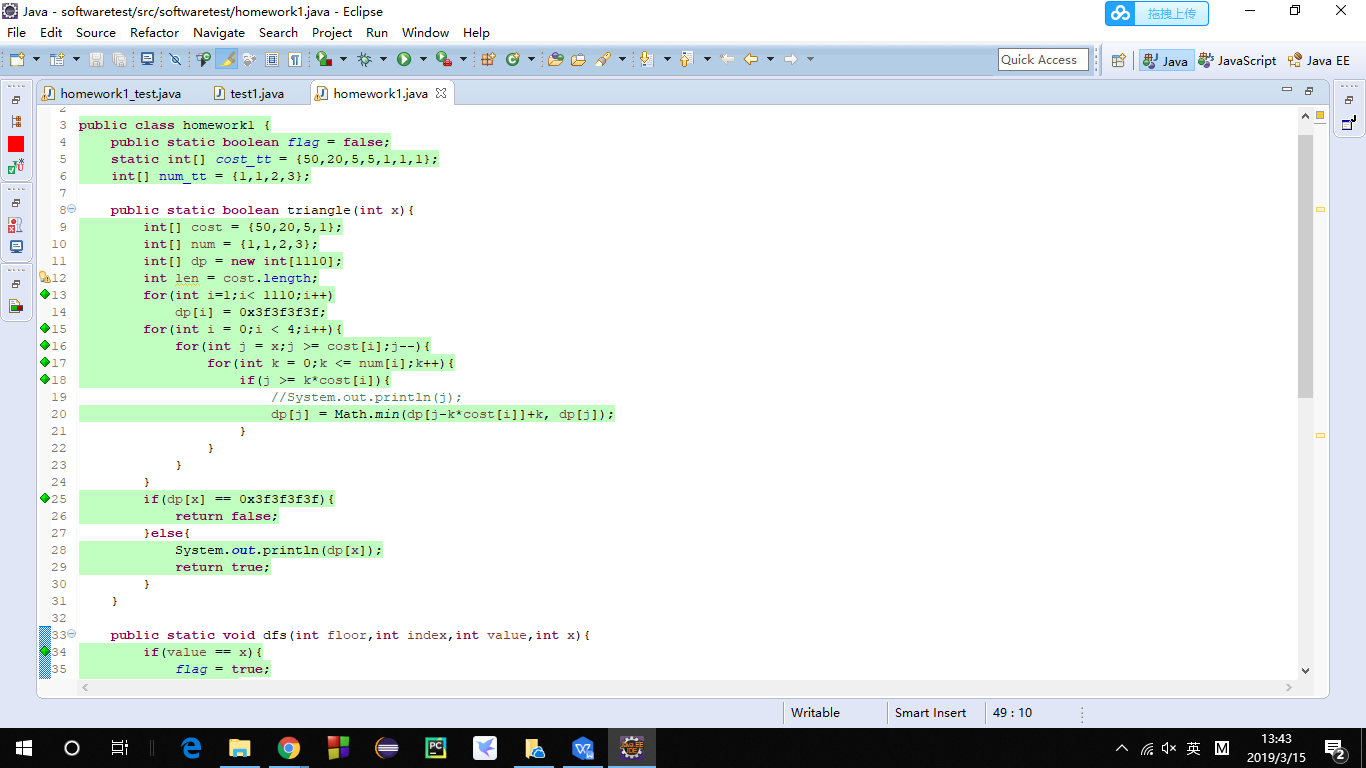
*assertEquals*(**false**,**new** homework1().*triangle*(124));

*assertEquals*(**true**,**new** homework1().*triangle*(75));

*assertEquals*(**false**,**new** homework1().*triangle*(79));

*assertEquals*(**false**,**new** homework1().*triangle*(98));





1. 总结

通过本次实验了解了junit和eclemma的自动化测试方法，对软件测试有了初步的了解，对测试用例的设计有了一定的了解。