天津大学

软件测试技术第一次实验报告

学　　院 智能与计算学部

专　　业 软件工程

年　　级 2016

姓　　名 张环禹

2019 年 3 月 12 日

软件测试技术第一次实验报告

一、 需求分析

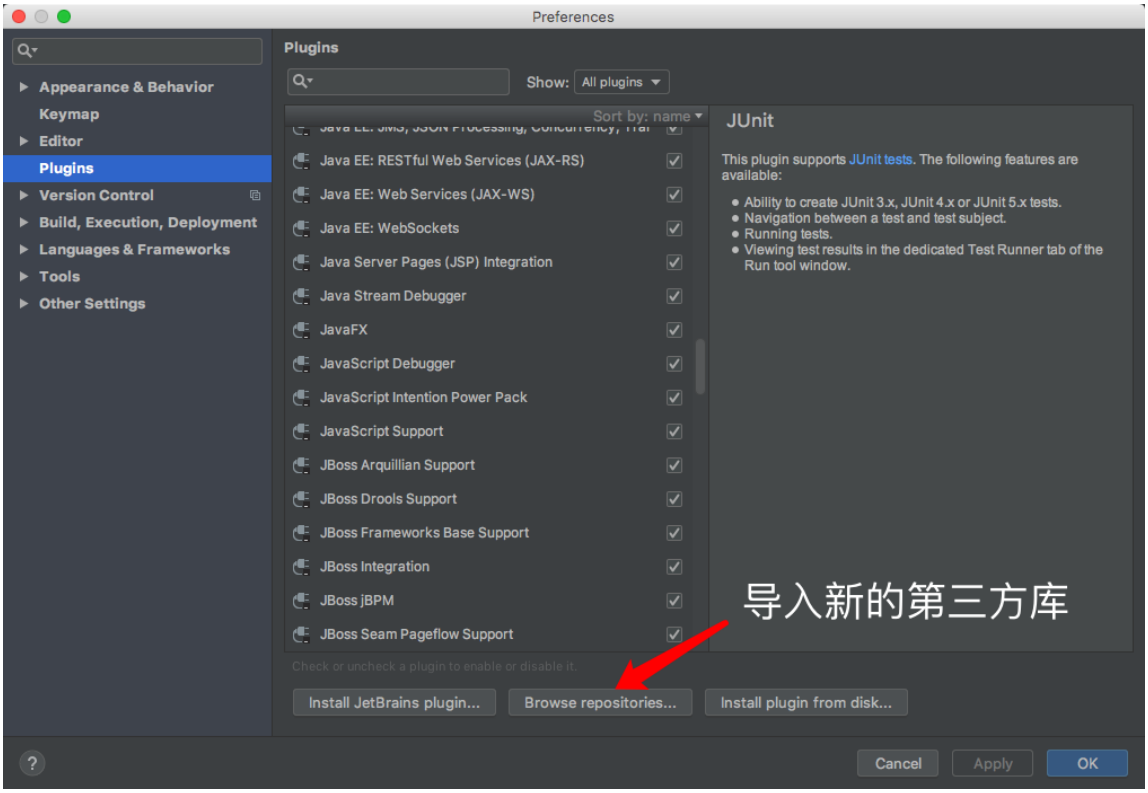
1. 安装 Junit、软件测试覆盖率工具
2. 编写代码解决如下问题并且使用 Junit 进行测试：
   a) There is one 50 yuan, one 20 yuan, two 5 yuan bills and three 1 yuan coins in your pocket. Write a program to find out whether you can take out a given number (x) yuan.

二、 概要设计

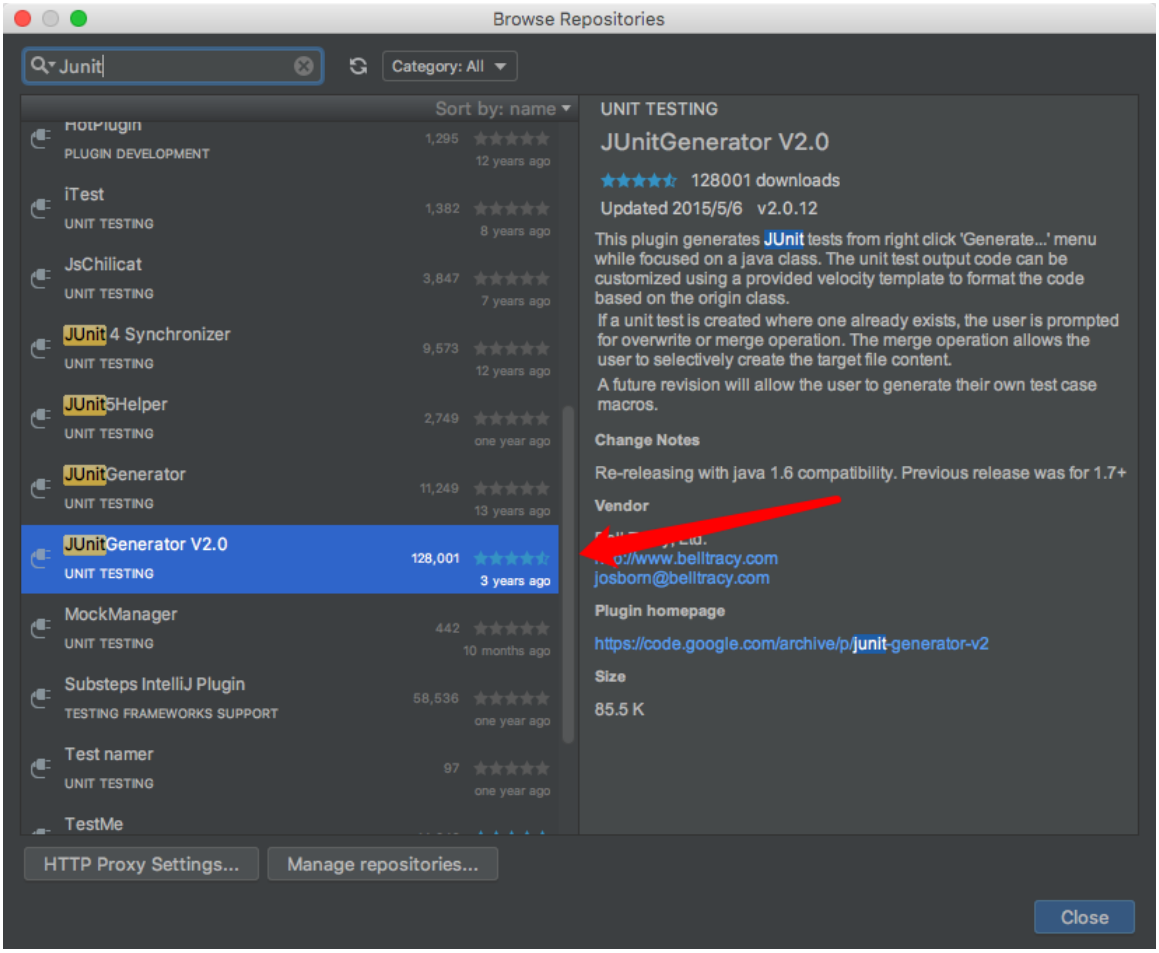
1. 本人使用 Intellij Idea 作为本次实验的 IDE，安装 Junit、软件测试覆盖率工具；
2. 编写解决 triangle problem 代码；
3. 编写测试类代码；
4. 输入测试数据，得到测试结果并分析覆盖率

三、 详细设计

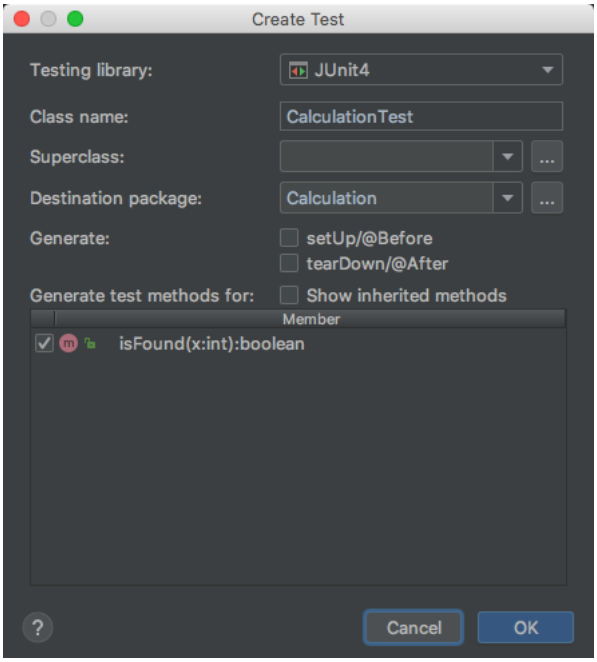
1. 安装 Junit、覆盖率工具已在
   a) 打开包管理平台



b） 由于 Idea 里已经装好 Junit 测试模块，因此我们仅仅需要安装好 Junit



Generator 来自动生成测试类的文件。之后重启 IDEA 使得新装包生效；



c） 生成测试类之后即可编写测试类代码

d） 因为 Idea 自带覆盖率测试工具，所以本次实验未使用 Eclemma；

2. 编写 triangle problem 解决代码

本人采用回溯法解决此问题，首先将硬币们按序排列，接着每次将一个不大于剩余没匹配的钱的硬币放入栈中，如果接下里的一枚硬币超过了剩余钱，则将栈顶的硬币弹出，从新的栈顶硬币遍历原栈顶位置之后的硬币，若能匹配到一个使得剩余钱数为 0 的硬币，输出 true；否则输出 false。以下为关键代码

```
for(int i=pos+1;i<coins.length;i++)
{
    if(flag)
    {
        break;
    }
    Stack <Integer>state2=(Stack<Integer>)state.clone();
    if(remains-coins[i]>=0)
    {
        state2.push(i);
        flag=dfs(state2,remains-coins[i]);
    }
}
state.pop();
```

3. 编写测试类代码

```
@RunWith(Parameterized.class)
public class calculationTest {
```

```
    private Calculation problem=new Calculation();
    private boolean sus;
    private int com;
    public calculationTest(boolean sus,int com){
        this.com=com;
        this.sus=sus;
    }
    @Parameterized.Parameters
    public static Collection<Object[]> getData(){
        return Arrays.asList(new Object[][]{
{true,2},{true,12},{true,9},{true,29},{false,34},{false,72},{false,100}
        }); }
    @org.junit.Test
    public void isFound() {
        assertEquals(sus,problem.isFound(com));
    }
}
```

四、 调试分析

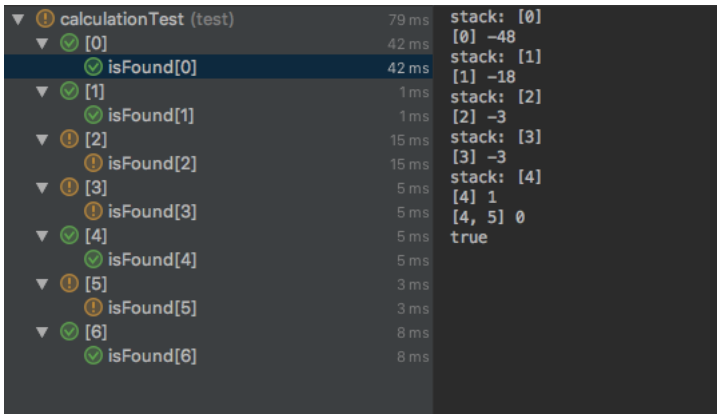
本次试验刚开始调试测试类功能时遇到了空指针错误，通过排查、Google 等方法发现问题出在于配置文件与 Junit 包不匹配的问题，将 Junit 包的顺序提到前面解决了这个问题。

五、 测试结果

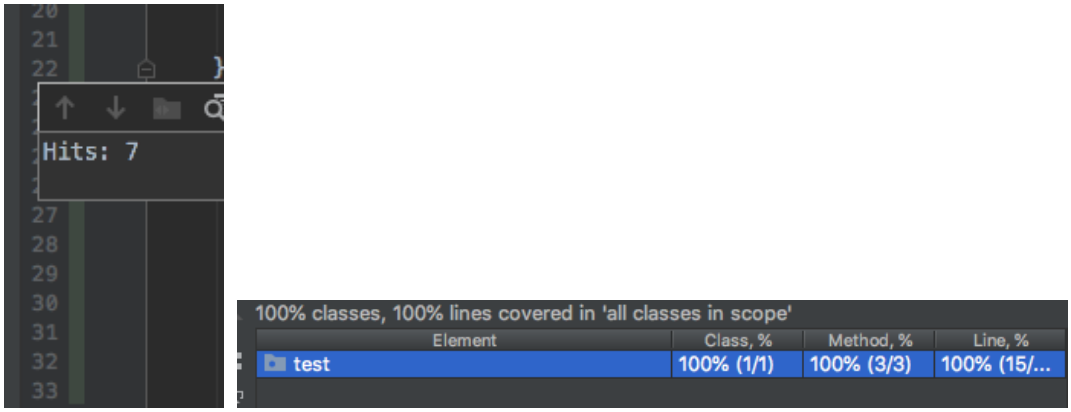
本次实验测试用例为随机选取，选取用例如下：

```
@Parameterized.Parameters
public static Collection<Object[]> getData(){
    return Arrays.asList(new Object[][]{
            {true,2},
            {true,12},
            {true,9},
            {true,29},
            {false,34},
            {false,72},
            {false,100}
    }); }
```

得到结果如下



以上输出为开始时栈中的硬币所在数组的位置；过程中栈中硬币的存入与弹出后的状态和剩余硬币的数量；最后的结果。



以上为测试覆盖率的模块，我们可以看到每句话所进行的次数和该包下的使用率。

本测试程序得到的结果和真实结果相一致。

六、总结

本次试验是第一次使用 Junit 和测试覆盖率工具，初步了解和学习到了

Junit 各个修饰器的作用和各种测试方法的使用，如：assertTrue、@Before、@Parameters 等东西的使用，会在接下来的学习中进一步掌握软件测试的各种理念、方法与工具。