**软件测试上机报告**

****

第一次上机作业

**学 院\_\_智能与计算学部\_\_**

**专 业\_\_软件工程 \_\_**

**姓 名\_\_马天宇\_\_\_\_\_\_\_\_\_\_**

**学 号\_\_3017218064\_\_\_\_\_\_**

**年 级\_\_2017级\_\_\_\_\_\_\_\_\_**

**班 级\_\_软工一班\_\_ \_\_\_\_\_**

# 一、实验要求

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

There is one 50 yuan, one 20 yuan, one 10 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、源代码

Money.java:

**package** ex1;

**public** **class** Money {

**private** **final** **static** **int** ***FIFTY*** = 50;

**private** **final** **static** **int** ***TWENTY*** = 20;

**private** **final** **static** **int** ***TEN*** = 10;

**private** **final** **static** **int** ***FIVE*** = 5;

**private** **final** **static** **int** ***ONE*** = 1;

**private** **final** **static** **int**[] ***NUM*** = {1, 1, 1, 2, 3};

**private** **final** **static** **int**[] ***MONEY*** = {***FIFTY***, ***TWENTY***, ***TEN***, ***FIVE***, ***ONE***};

**private** **final** **static** **int** ***MONEY\_TYPE*** = 5;

**protected** **boolean** hasMoney(**int** money) {

**return** **this**.hasMoney(money, ***NUM***);

}

**private** **boolean** hasMoney(**int** left, **int**[] used) {

**if**(left == 0) {

**return** **true**;

}

**for**(**int** i = 0; i < ***MONEY\_TYPE***; i++) {

**if**(left >= ***MONEY***[i] && used[i] > 0) {

// System.out.println(used[i]);

**int**[] use = **this**.copyArray(used);

use[i]--;

**if**(**this**.hasMoney(left - ***MONEY***[i], use) || **this**.hasMoney(left, use)) {

**return** **true**;

}

}

}

**return** **false**;

}

**private** **int**[] copyArray(**int**[] array) {

**final** **int** LENGTH = array.length;

**int**[] newArray = **new** **int**[LENGTH];

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

newArray[i] = array[i];

}

**return** newArray;

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

// m.hasMoney(50);

Money m = **new** Money();

**int**[] test = {1, 20, 70, 60, 55, 39};

// int[] test = {60};

**for**(**int** i = 0; i < test.length; i++) {

// Money m = new Money();

System.***out***.println(m.hasMoney(test[i]));

}

}

}

MoneyTest.java:

**package** ex1;

**import** **static** org.junit.Assert.\*;

**import** java.util.Arrays;

**import** java.util.Collection;

**import** org.junit.Before;

**import** org.junit.Test;

**import** org.junit.runner.RunWith;

**import** org.junit.runners.Parameterized;

**import** org.junit.runners.Parameterized.Parameters;

@RunWith(Parameterized.**class**)

**public** **class** MoneyTest {

**private** **int** input;

**private** **boolean** expected;

**private** Money money = **null**;

**public** MoneyTest(**int** input, **boolean** excepted) {

**this**.input = input;

**this**.expected = excepted;

}

@Before

**public** **void** setUp() {

money = **new** Money();

}

@Parameters

**public** **static** Collection<Object[]> getData(){

**return** Arrays.*asList*(**new** Object[][] {

{0, **true**},

{1000, **false**},

{93, **true**},

{94, **false**},

{92, **true**},

{88, **true**},

{89, **false**}

});

}

@Test

**public** **void** test() {

*assertEquals*(**this**.expected, money.hasMoney(**this**.input));

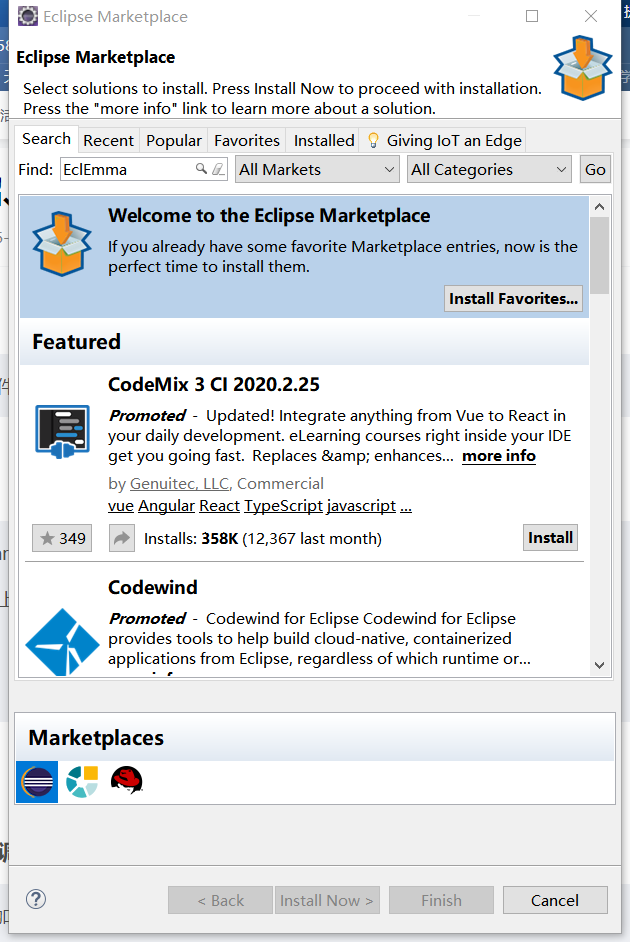
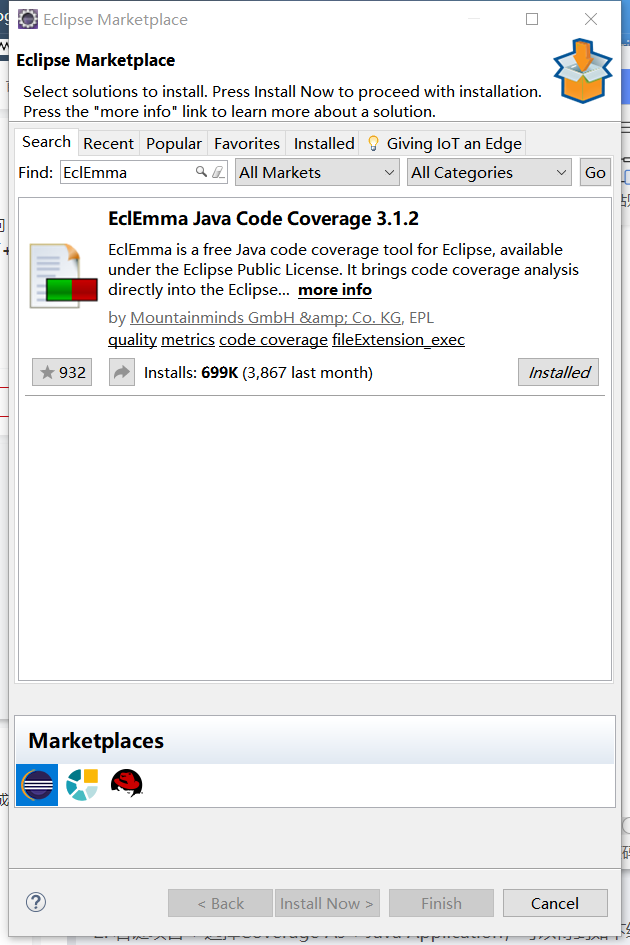
}

}

# 三、运行结果

1. Copy the required jar file to lib directory, and add them to build path, then we can import them in java file.

2. In Eclipse, choose Help->Eclipse Marketplace and search EclEmma，Install；

3. The code is shown at the last part. As for testing, I use parameterized test. I choose some critical values and random values for testing.

When I first tested, I found some failures. I printed some variables to find the fault, and I found a static constant array changed while the program ran. That confused me much. In honest, the constant variables should not change. I deleted my method and ran again, finding the variables not changed. So I thought my method changed the constant. So I checked my method, selected the operation that related to the constant, and located the fault at a sentence which copied the constant array to a local variable array, and the local array changed later. Then I realized the copy of Java. Maybe I just used shallow copy, and both the two arrays pointed to one memory, and while one changed, the other changed too. Therefore, I wrote a method to realize deep copy and ran the test cases again, with all test cases passing.

