```java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package lab6.solution.src.lab06;

/**
 *
 * @author vanting
 */
class AssertionDemo {

    private static int balance = 900;

    public static void main(String args[]) {
        System.out.println("Acc balance: " + withdrawMoney(500));
        System.out.println("Acc balance: " + withdrawMoney(500));
    }

    public static double withdrawMoney(double amount) {
        // precondition: the balance must be enough for the withdraw amount
        assert balance >= amount : "Insufficient Account Balance: " + (balance -
amount);

        balance -= amount;
        return balance;
    }
}

```



```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package lab6.solution.src.lab06;

/**
 *
 * @author vanting
 */
public class DimensionMismatchException extends Exception {

    private Matrix firstMatrix;
    private Matrix secondMatrix;

    public Matrix getFirstMatrix() {
        return firstMatrix;
    }

    public void setFirstMatrix(Matrix firstMatrix) {
        this.firstMatrix = firstMatrix;
    }

    public Matrix getSecondMatrix() {
        return secondMatrix;
    }

    public void setSecondMatrix(Matrix secondMatrix) {
        this.secondMatrix = secondMatrix;
    }
}

```

```java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package lab6.solution.src.lab06;

/**
 *
 * @author vanting
 */
public class Matrix {

    private int[][] elements;

    Matrix(int[][] e) {
        this.elements = e;
    }

    public int getElement(int row, int col) {
        return elements[row][col];
    }

    public void setElement(int row, int col, int value) {
        elements[row][col] = value;
    }

    public boolean add(Matrix matrix2) {

        int row1 = this.elements.length;
        int col1 = this.elements[0].length;
        int row2 = matrix2.elements.length;
        int col2 = matrix2.elements[0].length;

        if (row1 == row2 && col1 == col2) {
            for (int i = 0; i < row1; i++) {
                for (int j = 0; j < col1; j++) {
                    this.elements[i][j] += matrix2.elements[i][j];
                }
            }
            return true;
        } else {
            return false;
        }
    }

    public void print() {

        int row = this.elements.length;
        int col = this.elements[0].length;

        for (int i = 0; i < row; i++) {
            for (int j = 0; j < col; j++) {
                System.out.printf("[%02d]", this.elements[i][j]);
            }
            System.out.println("");
        }
    }

    // for Q6 in Lab 03
```



```java
    public Matrix multiply(Matrix matrix2) throws DimensionMismatchException {

        int row1 = this.elements.length;
        int col1 = this.elements[0].length;
        int row2 = matrix2.elements.length;
        int col2 = matrix2.elements[0].length;

        int sum = 0;
        int multiply[][] = new int[row1][col2];

        if (col1 == row2) {
            for (int i = 0; i < row1; i++) {
                for (int j = 0; j < col2; j++) {
                    for (int k = 0; k < col1; k++) {
                        sum = sum + elements[i][k] * matrix2.elements[k][j];
                    }
                    multiply[i][j] = sum;
                    sum = 0;
                }
            }
            return new Matrix(multiply);
        } else {
            DimensionMismatchException ex = new DimensionMismatchException();
            ex.setFirstMatrix(this);
            ex.setSecondMatrix(matrix2);
            throw ex;
        }
    }
}
```

```java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package lab6.solution.src.lab06;

/**
 *
 * @author vanting
 */
public class TestMatrix {

    public static void main(String[] args) {
        // initialize both matrices
        Matrix m1 = new Matrix(new int[][] { { 5, 15, 0 }, { 2, 22, 0 } });
        Matrix m2 = new Matrix(new int[][] { { 5, 6 }, { 7, 8 } });

        System.out.println("First matrix:");
        m1.print();

        System.out.println("Second matrix:");
        m2.print();

        System.out.println("");
        System.out.println("Result of multiplication:");

        try {
            m1.multiply(m2).print();
        } catch (DimensionMismatchException ex) {
            System.out.println("Invalid matrix size: ");
            System.out.println("First matrix:");
            ex.getFirstMatrix().print();
            System.out.println("Second matrix:");
            ex.getSecondMatrix().print();
        }
    }
}

```



```java
/*
 * This program should be placed in the Test Packages.
 */
package lab06;

import static junit.framework.Assert.*;
import org.junit.Before;
import org.junit.Test;


/**
 *
 * @author vanting
 */
public class MatrixUnitTest {

    Matrix matA;
    Matrix matB;
    Matrix matC;

    @Before
    public void setUp() {
        matA = new Matrix(new int[][]{{1,2,3},{4,5,6}});
        matB = new Matrix(new int[][]{{1,1,1},{2,2,2}});
        matC = new Matrix(new int[][]{{1},{1},{1}});
    }

    @Test
    public void getElementTest() {
        int[][] expected = new int[][]{{1,2,3},{4,5,6}};
        for(int i=0; i<expected.length; i++)
            for(int j=0; j< expected[0].length; j++)
                assertEquals(expected[i][j], matA.getElement(i, j));
    }

    @Test
    public void addTest() {
        int[][] expected = new int[][]{{2,3,4},{6,7,8}};
        matA.add(matB);
        for(int i=0; i<expected.length; i++)
            for(int j=0; j< expected[0].length; j++)
                assertEquals(expected[i][j], matA.getElement(i, j));
    }

    //--------------------------- to be done by students

    @Test
    public void multiplyOnSuccessTest() throws DimensionMismatchException {
        int[][] expected = new int[][]{{6},{15}};
        Matrix result = matA.multiply(matC);
        for(int i=0; i<expected.length; i++)
            for(int j=0; j< expected[0].length; j++)
                assertEquals(expected[i][j], result.getElement(i, j));
    }

    @Test (expected = DimensionMismatchException.class)
    public void multiplyOnFailTest() throws DimensionMismatchException {

        Matrix result = matA.multiply(matB);    // throw expection
```

```
60
61      }
62
63  }
64
```