

**LAB 1.1**

*Explain Junit Test with very simple example*

**Steps:**

- Create a new project in the name of MyTestApplication.
- Include one new Junit test case under the org.capgemini.testDemo package
  - RightClick→select Junit Testcase→include Junit 4 jars
- Add the following classes.

**Code:**

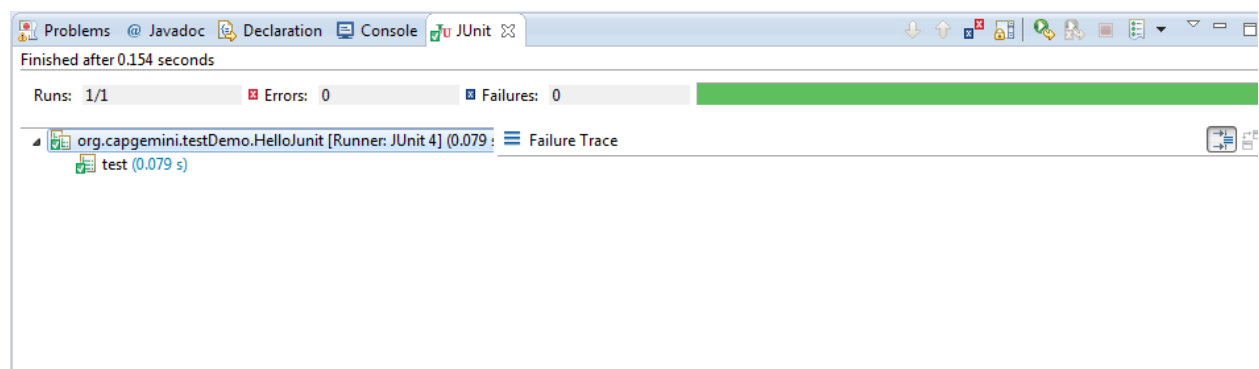
```
package org.capgemini.testDemo;

import static org.junit.Assert.*;

import org.junit.Test;
public class HelloJUnit {
    @Test
    public void test() {
        //fail("Not yet implemented");
    }
}
```

**Output:**

Right Click the class HelloJUnit→Run As→JUnit Test

**Learning:**

- From the above example, we learnt how to start the Junit in IDE.

## LAB 1.2

Explain Junit @Before and @After annotation with example

## Steps:

- Create new java project “ProteinTracker” in SpringToolSuite.
- Create new Package in the name of org.capgemini. And then add the following classes.

## Code:

HistoryItem.java

```
package org.capgemini;

public class HistoryItem {
    private final int id;
    private final int amount;
    private final String operation;
    private final int total;
    public int getId() {
        return id;
    }
    public int getAmount() {
        return amount;
    }
    public String getOperation() {
        return operation;
    }
    public int getTotal() {
        return total;
    }
    public HistoryItem(int id, int amount, String operation, int total) {

        this.id = id;
        this.amount = amount;
        this.operation = operation;
        this.total = total;
    }
}
```

TrackingService.java

```
package org.capgemini;
import java.util.ArrayList;
import java.util.List;

public class TrackingService {
```

```
private int total;
private int goal;
private List<HistoryItem> history=new ArrayList<HistoryItem>();
private int historyId=0;

public void addProtein(int amount){
    total+=amount;
    history.add(new HistoryItem(historyId++,amount,"add",total));
}

public void removeProtein(int amount){
    total-=amount;
    if(total<0)
        total=0;
    history.add(new HistoryItem(historyId++,amount,"subtract",total));
}

public int getTotal() {
    return total;
}

public void setTotal(int total) {
    goal = total;
}

public boolean isGoalMet(){
    return total>=goal;
}

public List<HistoryItem> getHistory() {
    return history;
}

public void setHistory(List<HistoryItem> history) {
    this.history = history;
}
```

```
public int getHistoryId() {  
    return historyId;  
}  
  
public void setHistoryId(int historyId) {  
    this.historyId = historyId;  
}  
  
}
```

- Create new java project “**ProteinTrackerTests**” in SpringToolSuite.
- Create new Package in the name of org.test.capgemini. And then add the following classes.

#### TrackingServiceTest.java

```
package org.test.capgemini;  
import org.capgemini.TrackingService;  
import org.junit.After;  
import org.junit.Before;  
import org.junit.Test;  
import static org.junit.Assert.*;  
  
public class TrackingServiceTest {  
  
    private TrackingService service;  
  
    @Before  
    public void setUp(){  
        System.out.println("Before");  
        service=new TrackingService();  
    }  
  
    @After  
    public void tearDown(){  
        System.out.println("After");  
    }  
  
    @Test  
    public void NewTrackingServiceIsZero(){  
  
        assertEquals("Tracking Service total was not zero",0, service.getTotal());  
    }  
}
```

```
}

@Test
public void whenAddingProteinTotalIncreasesByThatAmount(){

    service.addProtein(10);
    assertEquals(10,service.getTotal());

}

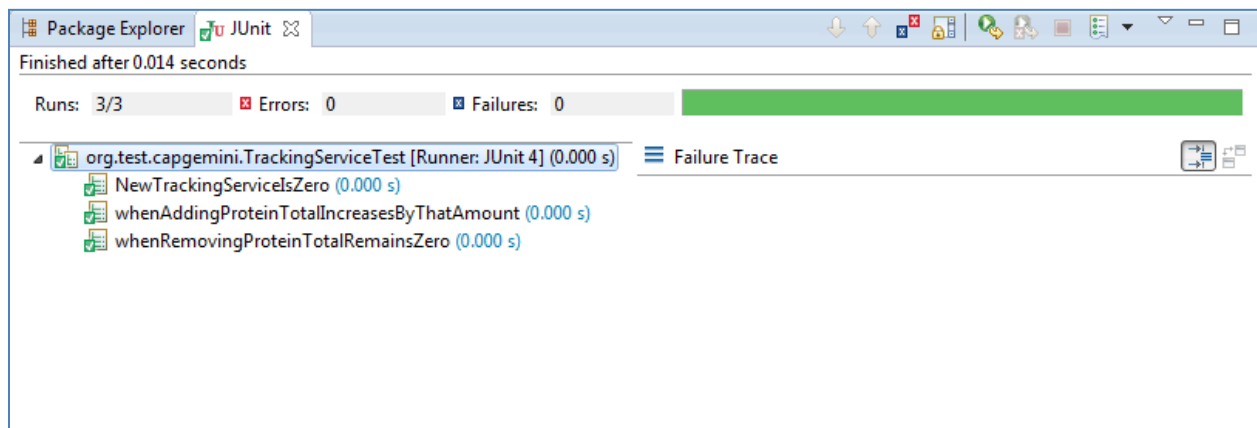
@Test
public void whenRemovingProteinTotalRemainsZero(){

    service.removeProtein(5);
    assertEquals(0 ,service.getTotal());

}

}
```

## Output



Before

After

Before

After

Before

After

## Learning:

- From the above example we learnt how to use the @Before and @After annotation in JUnit. These annotations will be invoked after and before calling each method in that class.

## LAB 1.3

Explain @BeforeClass ,@AfterClass and @Ignore with example

## Steps:

- Create new java project “ProteinTracker” in SpringToolSuite.
- Create new Package in the name of org.capgemini. And then add the following classes.

## Code:

HistoryItem.java

```
package org.capgemini;

public class HistoryItem {
    private final int id;
    private final int amount;
    private final String operation;
    private final int total;
    public int getId() {
        return id;
    }
    public int getAmount() {
        return amount;
    }
    public String getOperation() {
        return operation;
    }
    public int getTotal() {
        return total;
    }
    public HistoryItem(int id, int amount, String operation, int total) {

        this.id = id;
        this.amount = amount;
        this.operation = operation;
        this.total = total;
    }
}
```

TrackingService.java

```
package org.capgemini;
import java.util.ArrayList;
import java.util.List;

public class TrackingService {
    private int total;
    private int goal;
```

```
private List<HistoryItem> history=new ArrayList<HistoryItem>();
private int historyId=0;

public void addProtein(int amount){
    total+=amount;
    history.add(new HistoryItem(historyId++,amount,"add",total));
}

public void removeProtein(int amount){
    total-=amount;
    if(total<0)
        total=0;
    history.add(new HistoryItem(historyId++,amount,"subtract",total));
}

public int getTotal() {
    return total;
}

public void setTotal(int total) {
    goal = total;
}

public boolean isGoalMet(){
    return total>=goal;
}

public List<HistoryItem> getHistory() {
    return history;
}

public void setHistory(List<HistoryItem> history) {
    this.history = history;
}

public int getHistoryId() {
    return historyId;
}
```

```

    }

    public void setHistoryId(int historyId) {
        this.historyId = historyId;
    }
}

```

- Create new java project “**ProteinTrackerTests**” in SpringToolSuite.
- Create new Package in the name of org.test.capgemini. And then add the following classes.

#### TrackingServiceTest.java

```

package org.test.capgemini;
import org.capgemini.TrackingService;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Ignore;
import org.junit.Test;
import static org.junit.Assert.*;

public class TrackingServiceTest {

    private TrackingService service;

    @BeforeClass
    public static void before(){
        System.out.println("Before Class");
    }

    @AfterClass
    public static void after(){
        System.out.println("After Class");
    }

    @Before
    public void setUp(){
        System.out.println("Before");
        service=new TrackingService();
    }
}

```



```
}

@After
public void tearDown(){
    System.out.println("After");
}

@Test
public void NewTrackingServiceIsZero(){

    assertEquals("Tracking Service total was not zero",0, service.getTotal());
}

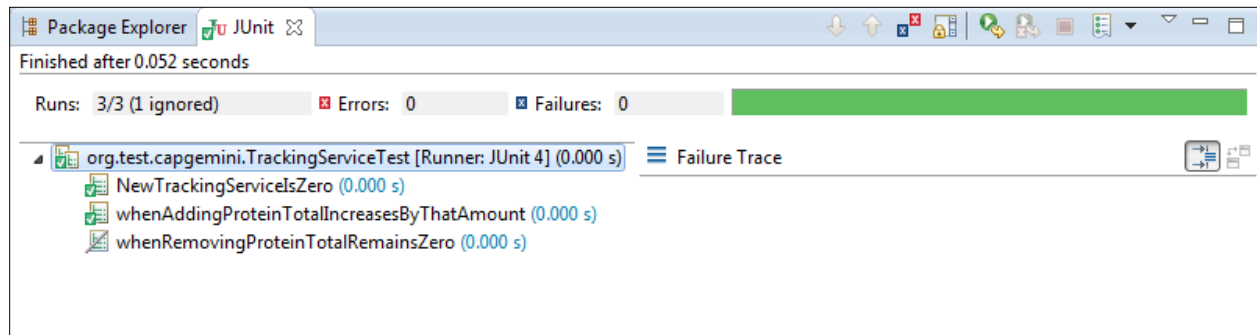
@Test
public void whenAddingProteinTotalIncreasesByThatAmount(){

    service.addProtein(10);
    assertEquals(10,service.getTotal());
}

@Test
@Ignore
public void whenRemovingProteinTotalRemainsZero(){

    service.removeProtein(5);
    assertEquals(0 ,service.getTotal());
}
}
```

## Output



Before Class

Before

After

Before

After

After Class

## Learning:

- @BeforeClass annotation should be used for static methods only. And it will be invoked once before calling any method in that class.
- @AfterClass annotation should be used for static methods only. And it will be invoked once after calling any method in that class. This method usually takes **clean up** process.
- @Ignore method will ignore a particular method from the test class.

## LAB 1.4

*Explain test case with exceptions in JUnit*

**Steps:**

- Create new java project “**ProteinTracker**” in SpringToolSuite.
- Create new Package in the name of org.capgemini. And then add the following classes.

**Code:****HistoryItem.java**

```
package org.capgemini;

import java.util.ArrayList;
import java.util.List;

public class TrackingService {
    private int total;
    private int goal;
    private List<HistoryItem> history=new ArrayList<HistoryItem>();
    private int historyId=0;

    public void addProtein(int amount){
        total+=amount;
        history.add(new HistoryItem(historyId++,amount,"add",total));
    }

    public void removeProtein(int amount){
        total-=amount;
        if(total<0)
            total=0;
        history.add(new
HistoryItem(historyId++,amount,"subtract",total));

    }

    public int getTotal() {
        return total;
    }

    public void setTotal(int total) {
        goal = total;
    }

    public boolean isGoalMet(){
```

```
        return total>=goal;
    }

    public int getGoal() {
        return goal;
    }

    public void setGoal(int goal) throws InvalidGoalException {
        if(goal<0)
            throw new InvalidGoalException();
        this.goal = goal;
    }

    public List<HistoryItem> getHistory() {
        return history;
    }

    public void setHistory(List<HistoryItem> history) {
        this.history = history;
    }

    public int getHistoryId() {
        return historyId;
    }

    public void setHistoryId(int historyId) {
        this.historyId = historyId;
    }
}
```

#### InvalidGoalException.java

```
package org.capgemini;

public class InvalidGoalException extends Exception {
}
}
```

#### TrackingService.java

```
package org.capgemini;

import java.util.ArrayList;
import java.util.List;
```

```
public class TrackingService {
    private int total;
    private int goal;
    private List<HistoryItem> history=new ArrayList<HistoryItem>();
    private int historyId=0;

    public void addProtein(int amount){
        total+=amount;
        history.add(new HistoryItem(historyId++,amount,"add",total));
    }

    public void removeProtein(int amount){
        total-=amount;
        if(total<0)
            total=0;
        history.add(new HistoryItem(historyId++,amount,"subtract",total));
    }

    public int getTotal() {
        return total;
    }

    public void setTotal(int total) {
        goal = total;
    }

    public boolean isGoalMet(){
        return total>=goal;
    }

    public int getGoal() {
        return goal;
    }
}
```

```

    public void setGoal(int goal) throws InvalidGoalException {
        if(goal<0)
            throw new InvalidGoalException();
        this.goal = goal;
    }

    public List<HistoryItem> getHistory() {
        return history;
    }

    public void setHistory(List<HistoryItem> history) {
        this.history = history;
    }

    public int getHistoryId() {
        return historyId;
    }

    public void setHistoryId(int historyId) {
        this.historyId = historyId;
    }
}

```

- Create new java project “**ProteinTrackerTests**” in SpringToolSuite.
- Create new Package in the name of org.test.capgemini. And then add the following classes.

#### TrackingServiceTest.java

```

package org.test.capgemini;
import org.capgemini.TrackingService;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Ignore;
import org.junit.Test;
import static org.junit.Assert.*;

public class TrackingServiceTest {

    private TrackingService service;

```

```
@BeforeClass
```

```
public static void before(){
```

```
    System.out.println("Before Class");
```

```
}
```

```
@AfterClass
```

```
public static void after(){
```

```
    System.out.println("After Class");
```

```
}
```

```
@Before
```

```
public void setUp(){
```

```
    System.out.println("Before");
```

```
    service=new TrackingService();
```

```
}
```

```
@After
```

```
public void tearDown(){
```

```
    System.out.println("After");
```

```
}
```

```
@Test
```

```
public void NewTrackingServiceIsZero(){
```

```
    assertEquals("Tracking Service total was not zero",0, service.getTotal());
```

```
}
```

```
@Test
```

```
public void whenAddingProteinTotalIncreasesByThatAmount(){
```

```
    service.addProtein(10);
```

```
    assertEquals(10,service.getTotal());
```

```
}
```

```
@Test
```

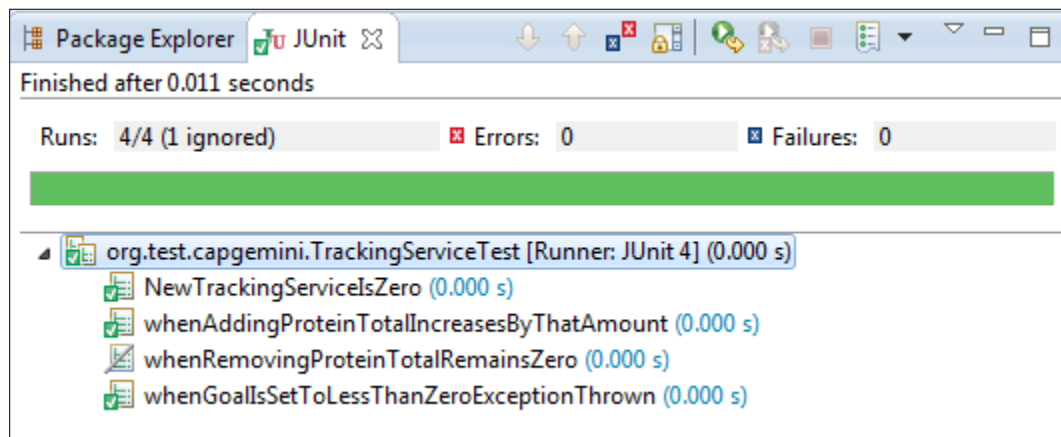
```
@Ignore
```

```
public void whenRemovingProteinTotalRemainsZero(){
```

```
        service.removeProtein(5);
        assertEquals(0 ,service.getTotal());
    }

    @Test(expected=InvalidGoalException.class)
    public void whenGoalIsSetToLessThanZeroExceptionThrown() throws
    InvalidGoalException{
        service.setGoal(-5);
    }
}
```

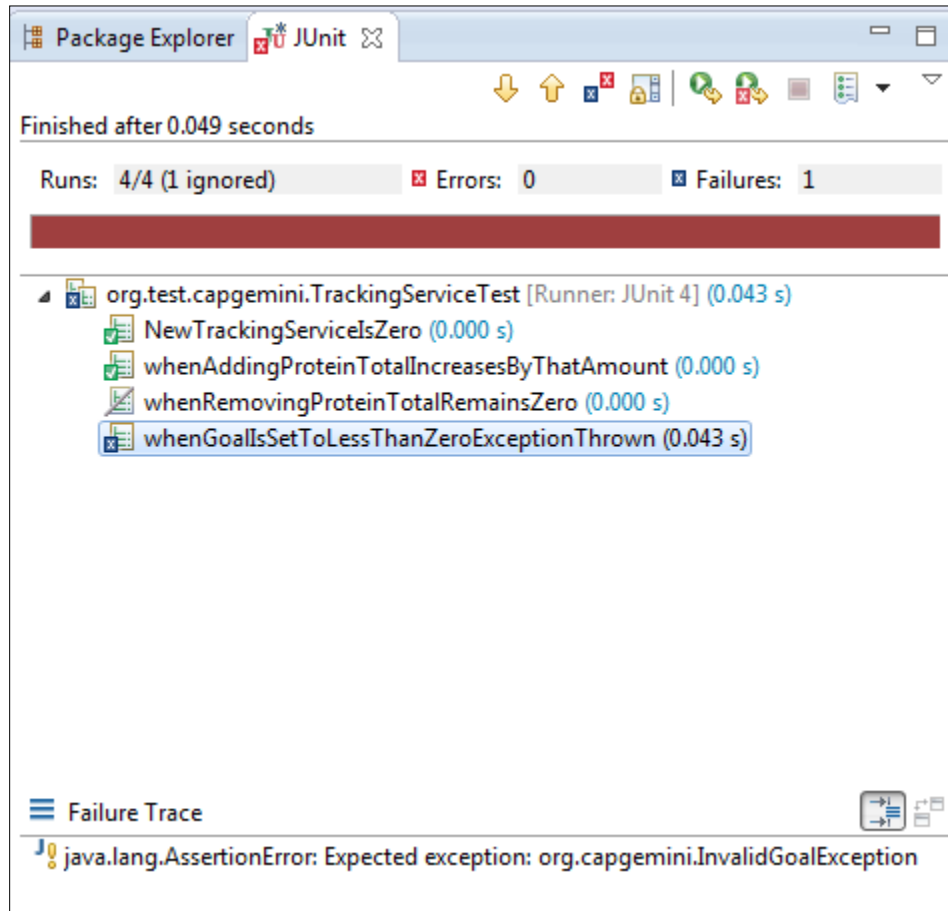
#### Output



If you comment the following 2 lines the result will be error.

```
//if(goal<0)
//throw new InvalidGoalException();
this.goal = goal;
```





Before Class  
Before  
After  
Before  
After  
After Class

#### Learning:

- From the above example we learnt how to test the exceptions in JUnit.

## LAB 1.5

*Explain test case with timeout in JUnit*

**Steps:**

- Create new java project “**ProteinTracker**” in SpringToolSuite.
- Create new Package in the name of org.capgemini. And then add the following classes.

**Code:****HistoryItem.java**

```
package org.capgemini;

import java.util.ArrayList;
import java.util.List;

public class TrackingService {
    private int total;
    private int goal;
    private List<HistoryItem> history=new ArrayList<HistoryItem>();
    private int historyId=0;

    public void addProtein(int amount){
        total+=amount;
        history.add(new HistoryItem(historyId++,amount,"add",total));
    }

    public void removeProtein(int amount){
        total-=amount;
        if(total<0)
            total=0;
        history.add(new
HistoryItem(historyId++,amount,"subtract",total));

    }

    public int getTotal() {
        return total;
    }

    public void setTotal(int total) {
        goal = total;
    }

    public boolean isGoalMet(){
```

```
        return total>=goal;
    }

    public int getGoal() {
        return goal;
    }

    public void setGoal(int goal) throws InvalidGoalException {
        if(goal<0)
            throw new InvalidGoalException();
        this.goal = goal;
    }

    public List<HistoryItem> getHistory() {
        return history;
    }

    public void setHistory(List<HistoryItem> history) {
        this.history = history;
    }

    public int getHistoryId() {
        return historyId;
    }

    public void setHistoryId(int historyId) {
        this.historyId = historyId;
    }
}
```

#### InvalidGoalException.java

```
package org.capgemini;

public class InvalidGoalException extends Exception {
}

```

#### TrackingService.java

```
package org.capgemini;

import java.util.ArrayList;
import java.util.List;
```

```
public class TrackingService {
    private int total;
    private int goal;
    private List<HistoryItem> history=new ArrayList<HistoryItem>();
    private int historyId=0;

    public void addProtein(int amount){
        total+=amount;
        history.add(new HistoryItem(historyId++,amount,"add",total));
    }

    public void removeProtein(int amount){
        total-=amount;
        if(total<0)
            total=0;
        history.add(new HistoryItem(historyId++,amount,"subtract",total));
    }

    public int getTotal() {
        return total;
    }

    public void setTotal(int total) {
        goal = total;
    }

    public boolean isGoalMet(){
        return total>=goal;
    }

    public int getGoal() {
        return goal;
    }
}
```

```

    public void setGoal(int goal) throws InvalidGoalException {
        if(goal<0)
            throw new InvalidGoalException();
        this.goal = goal;
    }

    public List<HistoryItem> getHistory() {
        return history;
    }

    public void setHistory(List<HistoryItem> history) {
        this.history = history;
    }

    public int getHistoryId() {
        return historyId;
    }

    public void setHistoryId(int historyId) {
        this.historyId = historyId;
    }
}

```

- Create new java project “**ProteinTrackerTests**” in SpringToolSuite.
- Create new Package in the name of org.test.capgemini. And then add the following classes.

#### TrackingServiceTest.java

```

package org.test.capgemini;
import org.capgemini.TrackingService;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Ignore;
import org.junit.Test;
import static org.junit.Assert.*;

public class TrackingServiceTest {

    private TrackingService service;

```

```
@BeforeClass
```

```
public static void before(){
```

```
    System.out.println("Before Class");
```

```
}
```

```
@AfterClass
```

```
public static void after(){
```

```
    System.out.println("After Class");
```

```
}
```

```
@Before
```

```
public void setUp(){
```

```
    System.out.println("Before");
```

```
    service=new TrackingService();
```

```
}
```

```
@After
```

```
public void tearDown(){
```

```
    System.out.println("After");
```

```
}
```

```
@Test
```

```
public void NewTrackingServiceIsZero(){
```

```
    assertEquals("Tracking Service total was not zero",0, service.getTotal());
```

```
}
```

```
@Test
```

```
public void whenAddingProteinTotalIncreasesByThatAmount(){
```

```
    service.addProtein(10);
```

```
    assertEquals(10,service.getTotal());
```

```
}
```

```
@Test
```

```
@Ignore
```

```
public void whenRemovingProteinTotalRemainsZero(){
```

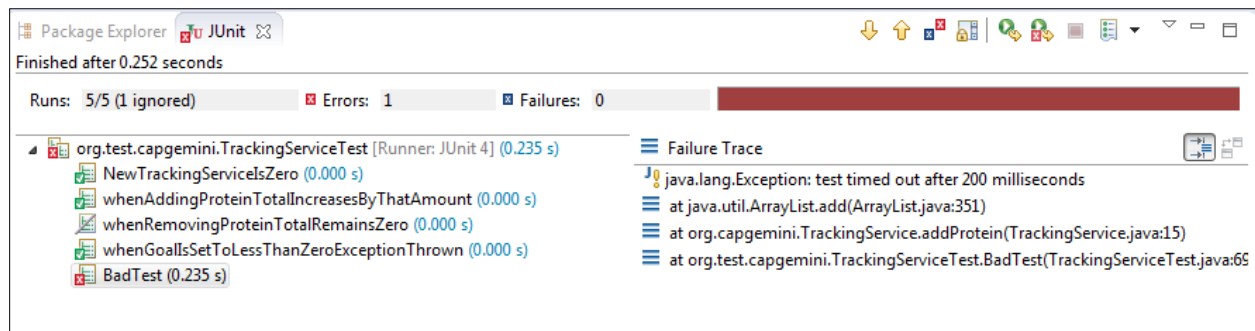
```

        service.removeProtein(5);
        assertEquals(0 ,service.getTotal());
    }

@Test(expected=InvalidGoalException.class)
    public void whenGoalIsSetToLessThanZeroExceptionThrown() throws
InvalidGoalException{
        service.setGoal(-5);
    }
@Test(timeout=200)
    public void BadTest(){
        for(int i=0;i<10000000;i++)
            service.addProtein(1);
    }
}

```

### Output



### Learning:

- From the above example we learnt how to handle the timeout for the session.

## LAB 1.6

*Explain how to create test suites in JUnit*

**Steps:**

- Create new java project “**ProteinTracker**” in SpringToolSuite.
- Create new Package in the name of org.capgemini. And then add the following classes.

**Code:****HistoryItem.java**

```
package org.capgemini;

import java.util.ArrayList;
import java.util.List;

public class TrackingService {
    private int total;
    private int goal;
    private List<HistoryItem> history=new ArrayList<HistoryItem>();
    private int historyId=0;

    public void addProtein(int amount){
        total+=amount;
        history.add(new HistoryItem(historyId++,amount,"add",total));
    }

    public void removeProtein(int amount){
        total-=amount;
        if(total<0)
            total=0;
        history.add(new
HistoryItem(historyId++,amount,"subtract",total));

    }

    public int getTotal() {
        return total;
    }

    public void setTotal(int total) {
        goal = total;
    }

    public boolean isGoalMet(){
```



```
        return total>=goal;
    }

    public int getGoal() {
        return goal;
    }

    public void setGoal(int goal) throws InvalidGoalException {
        if(goal<0)
            throw new InvalidGoalException();
        this.goal = goal;
    }

    public List<HistoryItem> getHistory() {
        return history;
    }

    public void setHistory(List<HistoryItem> history) {
        this.history = history;
    }

    public int getHistoryId() {
        return historyId;
    }

    public void setHistoryId(int historyId) {
        this.historyId = historyId;
    }
}
```

#### InvalidGoalException.java

```
package org.capgemini;

public class InvalidGoalException extends Exception {
}

```

#### TrackingService.java

```
package org.capgemini;

import java.util.ArrayList;
import java.util.List;
```

```
public class TrackingService {
    private int total;
    private int goal;
    private List<HistoryItem> history=new ArrayList<HistoryItem>();
    private int historyId=0;

    public void addProtein(int amount){
        total+=amount;
        history.add(new HistoryItem(historyId++,amount,"add",total));
    }

    public void removeProtein(int amount){
        total-=amount;
        if(total<0)
            total=0;
        history.add(new HistoryItem(historyId++,amount,"subtract",total));
    }

    public int getTotal() {
        return total;
    }

    public void setTotal(int total) {
        goal = total;
    }

    public boolean isGoalMet(){
        return total>=goal;
    }

    public int getGoal() {
        return goal;
    }
}
```

```

    public void setGoal(int goal) throws InvalidGoalException {
        if(goal<0)
            throw new InvalidGoalException();
        this.goal = goal;
    }

    public List<HistoryItem> getHistory() {
        return history;
    }

    public void setHistory(List<HistoryItem> history) {
        this.history = history;
    }

    public int getHistoryId() {
        return historyId;
    }

    public void setHistoryId(int historyId) {
        this.historyId = historyId;
    }
}

```

#### InvalidGoalException

```

package org.capgemini;
public class InvalidGoalException extends Exception {
}

```

- Create new java project “**ProteinTrackerAdvTests**” in SpringToolSuite.
- Create new Package in the name of org.test.capgemini. And then add the following classes.

#### HelloJUnitTests.java

```

package org.test.capgemini;
import static org.junit.Assert.*;
import org.junit.Test;

public class HelloJUnitTests {

    @Test
    public void test() {

```

```
        //fail("Not yet implemented");  
    }  
}
```

#### TrackingServiceTest.java

```
package org.test.capgemini;  
import org.capgemini.InvalidGoalException;  
import org.capgemini.TrackingService;  
import org.junit.After;  
import org.junit.AfterClass;  
import org.junit.Before;  
import org.junit.BeforeClass;  
import org.junit.Ignore;  
import org.junit.Test;  
import static org.junit.Assert.*;  
  
public class TrackingServiceTest {  
  
    private TrackingService service;  
  
    @BeforeClass  
    public static void before(){  
        System.out.println("Before Class");  
    }  
  
    @AfterClass  
    public static void after(){  
        System.out.println("After Class");  
    }  
  
    @Before  
    public void setUp(){  
        System.out.println("Before");  
        service=new TrackingService();  
    }  
  
    @After  
    public void tearDown(){  
        System.out.println("After");  
    }  
}
```

```
@Test
public void NewTrackingServiceIsZero(){

    assertEquals("Tracking Service total was not zero",0, service.getTotal());
}

@Test
public void whenAddingProteinTotalIncreasesByThatAmount(){

    service.addProtein(10);
    assertEquals(10,service.getTotal());
}

@Test
@Ignore
public void whenRemovingProteinTotalRemainsZero(){

    service.removeProtein(5);
    assertEquals(0 ,service.getTotal());
}

@Test(expected=InvalidGoalException.class)
public void whenGoalsSetToLessThanZeroExceptionThrown() throws InvalidGoalException{
    service.setGoal(-5);
}

@Test(timeout=200)
public void BadTest(){
    for(int i=0;i<10000000;i++)
        service.addProtein(1);
}
}

ProteinTrackerSuite.java
package org.test.capgemini;

import org.junit.runner.RunWith;
```

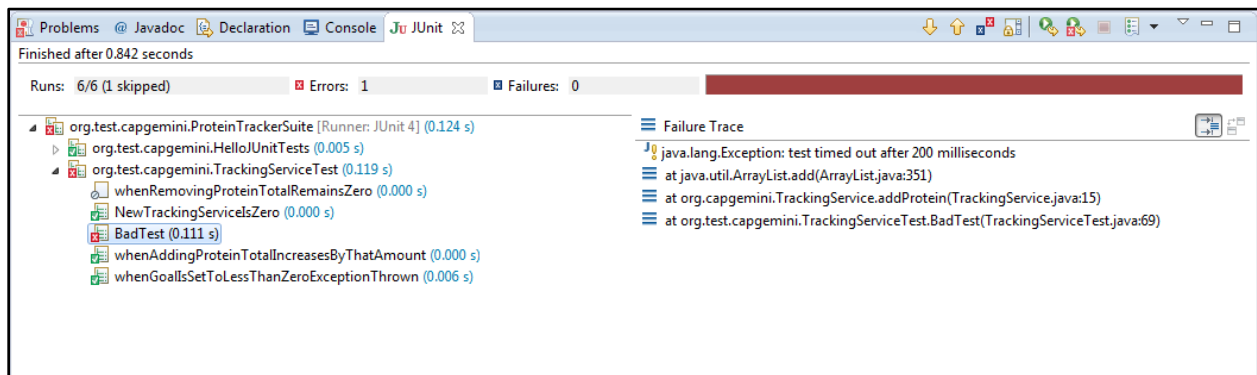
```
import org.junit.runners.Suite;

@RunWith(Suite.class)
@Suite.SuiteClasses({
    HelloJUnitTests.class,
    TrackingServiceTest.class
})
public class ProteinTrackerSuite {

}
```

### Output

Right Click the ProteinTrackerSuite → Run As → JUnit Test



### Learning:

From the above example we learnt how to create test suite.

## LAB 1.7

*Explain how to categories in JUnit*

**Steps:**

- Create new java project “**ProteinTracker**” in SpringToolSuite.
- Create new Package in the name of org.capgemini. And then add the following classes.

**Code:****HistoryItem.java**

```
package org.capgemini;

import java.util.ArrayList;
import java.util.List;

public class TrackingService {
    private int total;
    private int goal;
    private List<HistoryItem> history=new ArrayList<HistoryItem>();
    private int historyId=0;

    public void addProtein(int amount){
        total+=amount;
        history.add(new HistoryItem(historyId++,amount,"add",total));
    }

    public void removeProtein(int amount){
        total-=amount;
        if(total<0)
            total=0;
        history.add(new
HistoryItem(historyId++,amount,"subtract",total));

    }

    public int getTotal() {
        return total;
    }

    public void setTotal(int total) {
        goal = total;
    }

    public boolean isGoalMet(){
```

```

        return total>=goal;
    }

    public int getGoal() {
        return goal;
    }

    public void setGoal(int goal) throws InvalidGoalException {
        if(goal<0)
            throw new InvalidGoalException();
        this.goal = goal;
    }

    public List<HistoryItem> getHistory() {
        return history;
    }

    public void setHistory(List<HistoryItem> history) {
        this.history = history;
    }

    public int getHistoryId() {
        return historyId;
    }

    public void setHistoryId(int historyId) {
        this.historyId = historyId;
    }
}

```

#### InvalidGoalException.java

```

package org.capgemini;

public class InvalidGoalException extends Exception {

}

```

- Create new java project “**ProteinTrackerAdvTests**” in SpringToolSuite.
- Create new Package in the name of org.test.capgemini. And then add the following classes.

#### TrackingService.java

```

package org.capgemini;

```



```
import java.util.ArrayList;
import java.util.List;

public class TrackingService {
    private int total;
    private int goal;
    private List<HistoryItem> history=new ArrayList<HistoryItem>();
    private int historyId=0;

    public void addProtein(int amount){
        total+=amount;
        history.add(new HistoryItem(historyId++,amount,"add",total));
    }

    public void removeProtein(int amount){
        total-=amount;
        if(total<0)
            total=0;
        history.add(new HistoryItem(historyId++,amount,"subtract",total));
    }

    public int getTotal() {
        return total;
    }

    public void setTotal(int total) {
        goal = total;
    }

    public boolean isGoalMet(){
        return total>=goal;
    }

    public int getGoal() {
```

```

        return goal;
    }

    public void setGoal(int goal) throws InvalidGoalException {
        if(goal<0)
            throw new InvalidGoalException();
        this.goal = goal;
    }

    public List<HistoryItem> getHistory() {
        return history;
    }

    public void setHistory(List<HistoryItem> history) {
        this.history = history;
    }

    public int getHistoryId() {
        return historyId;
    }

    public void setHistoryId(int historyId) {
        this.historyId = historyId;
    }
}

```

#### InvalidGoalException

```

package org.capgemini;
public class InvalidGoalException extends Exception {
}

```

```

package org.test.capgemini;
import static org.junit.Assert.*;

import org.junit.Test;
import org.junit.experimental.categories.Category;

@Category(GoodTestCategory.class)
public class HelloJUnitTests {

```

```
@Test
public void test() {
    //fail("Not yet implemented");
}

}

package org.test.capgemini;
import org.capgemini.InvalidGoalException;
import org.capgemini.TrackingService;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Ignore;
import org.junit.Test;
import org.junit.experimental.categories.Category;

import static org.junit.Assert.*;

public class TrackingServiceTest {

    private TrackingService service;

    @BeforeClass
    public static void before(){
        System.out.println("Before Class");
    }

    @AfterClass
    public static void after(){
        System.out.println("After Class");
    }

    @Before
    public void setUp(){
        System.out.println("Before");
        service=new TrackingService();
    }

    @After
    public void tearDown(){
```

```
        System.out.println("After");
    }

    @Test
    @Category({GoodTestCategory.class,BadTestCategory.class})
    public void NewTrackingServiceIsZero(){

        assertEquals("Tracking Service total was not zero",0, service.getTotal());
    }

    @Test
    @Category(GoodTestCategory.class)
    public void whenAddingProteinTotalIncreasesByThatAmount(){

        service.addProtein(10);
        assertEquals(10,service.getTotal());
    }

    @Test
    // @Ignore
    @Category(GoodTestCategory.class)
    public void whenRemovingProteinTotalRemainsZero(){

        service.removeProtein(5);
        assertEquals(0 ,service.getTotal());
    }

    @Test(expected=InvalidGoalException.class)
    public void whenGoalsSetToLessThanZeroExceptionThrown() throws InvalidGoalException{
        service.setGoal(-5);
    }

    @Test(timeout=200)
    public void BadTest(){
        for(int i=0;i<10000000;i++)
            service.addProtein(1);
    }
```

```

}

package org.test.capgemini;

import org.junit.runner.RunWith;
import org.junit.runners.Suite;

@RunWith(Suite.class)
@Suite.SuiteClasses({
    HelloJUnitTests.class,
    TrackingServiceTest.class
})
public class ProteinTrackerSuite {
}

package org.test.capgemini;

public interface GoodTestCategory {
}

package org.test.capgemini;

public interface BadCategory {
}

package org.test.capgemini;

import org.junit.experimental.categories.Categories;
import org.junit.experimental.categories.Categories.ExcludeCategory;
import org.junit.experimental.categories.Categories.IncludeCategory;
import org.junit.runner.RunWith;
import org.junit.runners.Suite;

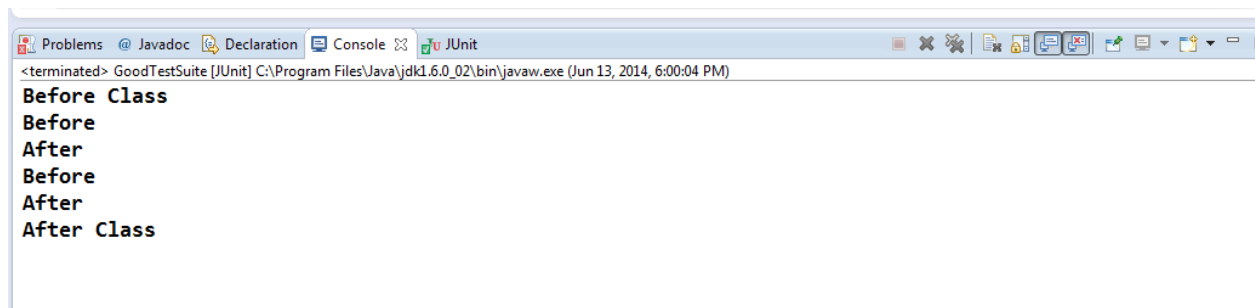
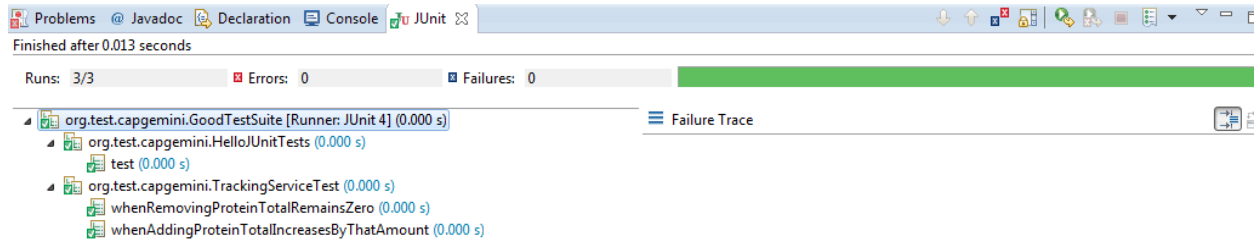
@RunWith(Categories.class)
@IncludeCategory(GoodTestCategory.class)
@ExcludeCategory(BadCategory.class)
@Suite.SuiteClasses({
    HelloJUnitTests.class,
    TrackingServiceTest.class
})
public class GoodTestSuite {

```

}

## Output

Right Click the ProteinTrackerSuite → Run As → JUnit Test



## Learning:

From the above example we learnt how to include categories.

**LAB 1.8**

*Explain how to create parameterized test cases in JUnit*

**Steps:**

- Create new java project “**ProteinTracker**” in SpringToolSuite.
- Create new Package in the name of org.capgemini. And then add the following classes.

**Code:****HistoryItem.java**

```
package org.capgemini;

import java.util.ArrayList;
import java.util.List;

public class TrackingService {
    private int total;
    private int goal;
    private List<HistoryItem> history=new ArrayList<HistoryItem>();
    private int historyId=0;

    public void addProtein(int amount){
        total+=amount;
        history.add(new HistoryItem(historyId++,amount,"add",total));
    }

    public void removeProtein(int amount){
        total-=amount;
        if(total<0)
            total=0;
        history.add(new
HistoryItem(historyId++,amount,"subtract",total));

    }

    public int getTotal() {
        return total;
    }

    public void setTotal(int total) {
        goal = total;
    }
}
```

```

    public boolean isGoalMet(){
        return total>=goal;
    }

    public int getGoal() {
        return goal;
    }

    public void setGoal(int goal) throws InvalidGoalException {
        if(goal<0)
            throw new InvalidGoalException();
        this.goal = goal;
    }

    public List<HistoryItem> getHistory() {
        return history;
    }

    public void setHistory(List<HistoryItem> history) {
        this.history = history;
    }

    public int getHistoryId() {
        return historyId;
    }

    public void setHistoryId(int historyId) {
        this.historyId = historyId;
    }
}

```

#### InvalidGoalException.java

```
package org.capgemini;
```

```
public class InvalidGoalException extends Exception {
}

```

- Create new java project “**ProteinTrackerAdvTests**” in SpringToolSuite.
- Create new Package in the name of org.test.capgemini. And then add the following classes.

[TrackingService.java](#)



```
package org.capgemini;

import java.util.ArrayList;
import java.util.List;

public class TrackingService {
    private int total;
    private int goal;
    private List<HistoryItem> history=new ArrayList<HistoryItem>();
    private int historyId=0;

    public void addProtein(int amount){
        total+=amount;
        history.add(new HistoryItem(historyId++,amount,"add",total));
    }

    public void removeProtein(int amount){
        total-=amount;
        if(total<0)
            total=0;
        history.add(new HistoryItem(historyId++,amount,"subtract",total));
    }

    public int getTotal() {
        return total;
    }

    public void setTotal(int total) {
        goal = total;
    }

    public boolean isGoalMet(){
        return total>=goal;
    }
}
```

```
public int getGoal() {
    return goal;
}

public void setGoal(int goal) throws InvalidGoalException {
    if(goal<0)
        throw new InvalidGoalException();
    this.goal = goal;
}

public List<HistoryItem> getHistory() {
    return history;
}

public void setHistory(List<HistoryItem> history) {
    this.history = history;
}

public int getHistoryId() {
    return historyId;
}

public void setHistoryId(int historyId) {
    this.historyId = historyId;
}
}
```

#### InvalidGoalException

```
package org.capgemini;
public class InvalidGoalException extends Exception {
}
```

#### ParameterizedTest.java

```
package org.test.capgemini;
import java.util.Arrays;
import java.util.List;
import org.capgemini.TrackingService;
```

```
import org.junit.Test;
import org.junit.runner.RunWith;
import org.junit.runners.Parameterized;
import org.junit.runners.Parameterized.Parameters;
import static org.junit.Assert.*;

@RunWith(Parameterized.class)
public class ParameterizedTest {

    private static TrackingService service=new TrackingService();
    private int input;
    private int expected;
    @Parameters
    public static List<Object[]> data(){
        return Arrays.asList(new Object[][]{
            {5,5},
            {5,10},
            {-12,0},
            {50,50},
            {1,51}
        });
    }

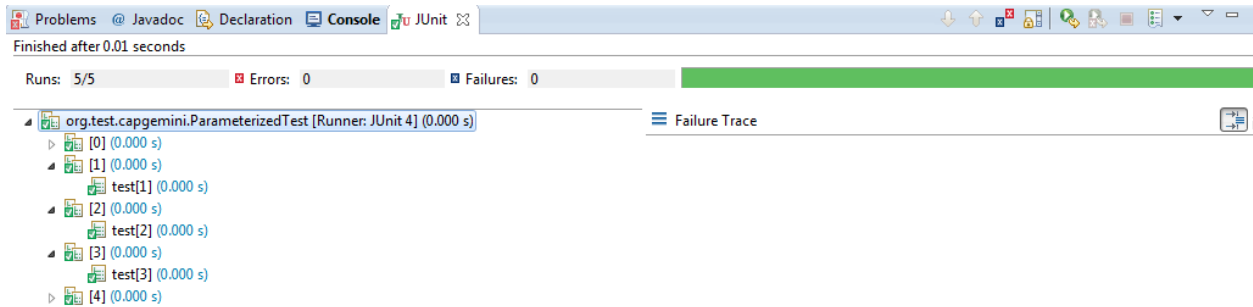
    public ParameterizedTest(int input,int expected){
        this.input=input;
        this.expected=expected;
    }

    @Test
    public void test(){
        if(input>=0)
            service.addProtein(input);
        else
            service.removeProtein(-input);

        assertEquals(expected,service.getTotal());
    }
}
```

## Output

Right Click the **ParameterizedTest** → Run As → JUnit Test



## If you change

```
{5,5},  
{5,10},  
{-12,1},  
{50,50},  
{1,51}
```

Then the output will fails

## Learning:

From the above example we have learnt the parameterized test cases.

## LAB 1.9

*Explain how to handle advanced assertions.*

**Steps:**

- Create new java project “**ProteinTracker**” in SpringToolSuite.
- Create new Package in the name of org.capgemini. And then add the following classes.

**Code:****HistoryItem.java**

```
package org.capgemini;

import java.util.ArrayList;
import java.util.List;

public class TrackingService {
    private int total;
    private int goal;
    private List<HistoryItem> history=new ArrayList<HistoryItem>();
    private int historyId=0;

    public void addProtein(int amount){
        total+=amount;
        history.add(new HistoryItem(historyId++,amount,"add",total));
    }

    public void removeProtein(int amount){
        total-=amount;
        if(total<0)
            total=0;
        history.add(new
HistoryItem(historyId++,amount,"subtract",total));

    }

    public int getTotal() {
        return total;
    }

    public void setTotal(int total) {
        goal = total;
    }
}
```

```
    public boolean isGoalMet(){
        return total>=goal;
    }

    public int getGoal() {
        return goal;
    }

    public void setGoal(int goal) throws InvalidGoalException {
        if(goal<0)
            throw new InvalidGoalException();
        this.goal = goal;
    }

    public List<HistoryItem> getHistory() {
        return history;
    }

    public void setHistory(List<HistoryItem> history) {
        this.history = history;
    }

    public int getHistoryId() {
        return historyId;
    }

    public void setHistoryId(int historyId) {
        this.historyId = historyId;
    }
}
```

#### InvalidGoalException.java

```
package org.capgemini;
```

```
public class InvalidGoalException extends Exception {
}
```

- Create new java project “ProteinTrackerAdvTests” in SpringToolSuite.
- Create new Package in the name of org.test.capgemini. And then add the following classes.

**TrackingServiceTest.java**

```
package org.test.capgemini;
import org.capgemini.InvalidGoalException;
import org.capgemini.TrackingService;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Ignore;
import org.junit.Test;
import org.junit.experimental.categories.Category;

import static org.junit.Assert.assertEquals;
import static org.hamcrest.CoreMatchers.*;
import static org.junit.matchers.JUnitMatchers.*;

import static org.junit.Assert.*;

public class TrackingServiceTest {

    private TrackingService service;

    @BeforeClass
    public static void before(){
        System.out.println("Before Class");
    }

    @AfterClass
    public static void after(){
        System.out.println("After Class");
    }

    @Before
    public void setUp(){
        System.out.println("Before");
        service=new TrackingService();
    }

    @After
    public void tearDown(){
        System.out.println("After");
    }
}
```

```

    }

    @Test
    @Category({GoodTestCategory.class, BadTestCategory.class})
    public void NewTrackingServiceIsZero(){

        assertEquals("Tracking Service total was not zero", 0,
service.getTotal());
    }

    @Test
    @Category(GoodTestCategory.class)
    public void whenAddingProteinTotalIncreasesByThatAmount(){

        service.addProtein(10);
        //assertEquals(10, service.getTotal());
        //assertThat(service.getTotal(), is(10));

        assertThat(service.getTotal(), allOf(is(10), instanceOf(Integer.class)));
    }

    @Test
    // @Ignore
    @Category(GoodTestCategory.class)
    public void whenRemovingProteinTotalRemainsZero(){

        service.removeProtein(5);
        assertEquals(0, service.getTotal());
    }

    @Test(expected=InvalidGoalException.class)
    public void whenGoalIsSetToLessThanZeroExceptionThrown() throws
InvalidGoalException{
        service.setGoal(-5);
    }

    /*
    @Test(timeout=200)
    public void BadTest(){
        for(int i=0; i<10000000; i++)
            service.addProtein(1);
    }

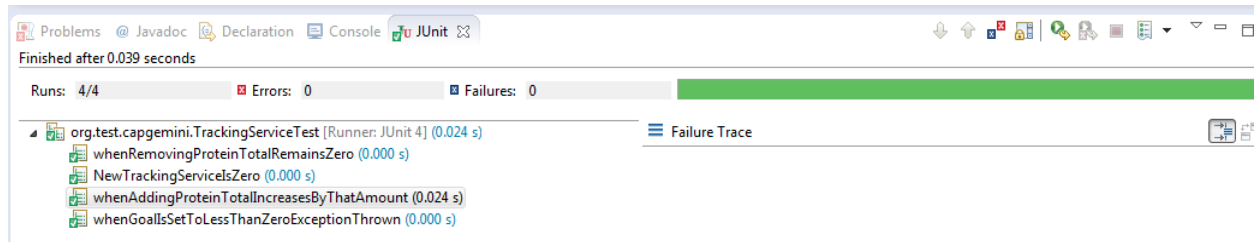
```



```
}* /
}
```

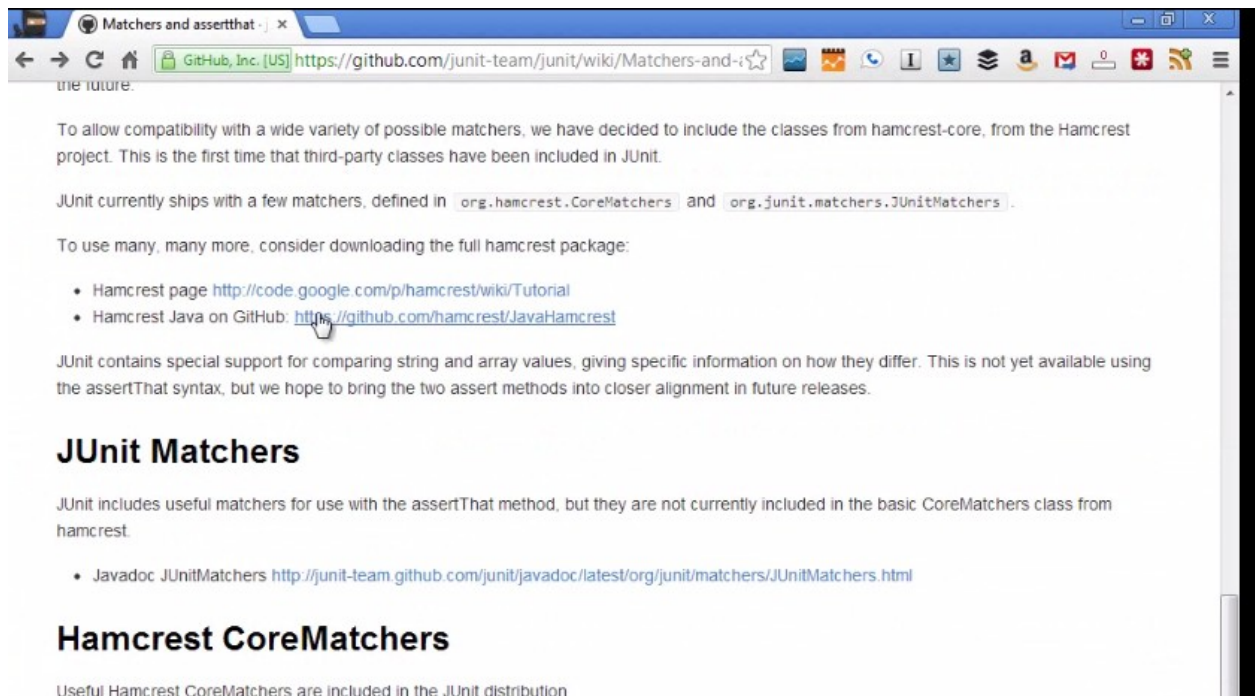
## Output

Right Click the ProteinTrackerSuite → Run As → JUnit Test



## Learning:

To learn more about this assertion please refer this site.



## LAB 1.10

*Explain advanced exception testing in JUnit.*

**Steps:**

- Create new java project “**ProteinTracker**” in SpringToolSuite.
- Create new Package in the name of org.capgemini. And then add the following classes.

**Code:****HistoryItem.java**

```
package org.capgemini;

import java.util.ArrayList;
import java.util.List;

public class TrackingService {
    private int total;
    private int goal;
    private List<HistoryItem> history=new ArrayList<HistoryItem>();
    private int historyId=0;

    public void addProtein(int amount){
        total+=amount;
        history.add(new HistoryItem(historyId++,amount,"add",total));
    }

    public void removeProtein(int amount){
        total-=amount;
        if(total<0)
            total=0;
        history.add(new
HistoryItem(historyId++,amount,"subtract",total));

    }

    public int getTotal() {
        return total;
    }

    public void setTotal(int total) {
        goal = total;
    }

    public boolean isGoalMet(){
```

```
        return total>=goal;
    }

    public int getGoal() {
        return goal;
    }

    public void setGoal(int goal) throws InvalidGoalException {
        if(goal<0)
            throw new InvalidGoalException("Goal is less than
zero");
        this.goal = goal;
    }

    public List<HistoryItem> getHistory() {
        return history;
    }

    public void setHistory(List<HistoryItem> history) {
        this.history = history;
    }

    public int getHistoryId() {
        return historyId;
    }

    public void setHistoryId(int historyId) {
        this.historyId = historyId;
    }
}
```

#### InvalidGoalException.java

```
package org.capgemini;

public class InvalidGoalException extends Exception {
    public InvalidGoalException(String msg){
        super(msg);
    }
}
```

- Create new java project “ProteinTrackerAdvTests” in SpringToolSuite.
- Create new Package in the name of org.test.capgemini. And then add the following classes.

**TrackingServiceTest.java**

```
package org.test.capgemini;
import org.capgemini.InvalidGoalException;
import org.capgemini.TrackingService;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Ignore;
import org.junit.Test;
import org.junit.experimental.categories.Category;

import static org.junit.Assert.assertEquals;
import static org.hamcrest.CoreMatchers.*;
import static org.junit.matchers.JUnitMatchers.*;

import static org.junit.Assert.*;

public class TrackingServiceTest {

    private TrackingService service;

    @BeforeClass
    public static void before(){
        System.out.println("Before Class");
    }

    @AfterClass
    public static void after(){
        System.out.println("After Class");
    }

    @Before
    public void setUp(){
        System.out.println("Before");
        service=new TrackingService();
    }

    @After
    public void tearDown(){
        System.out.println("After");
    }
}
```

```

    }

    @Test
    @Category({GoodTestCategory.class, BadTestCategory.class})
    public void NewTrackingServiceIsZero(){

        assertEquals("Tracking Service total was not zero", 0,
service.getTotal());
    }

    @Test
    @Category(GoodTestCategory.class)
    public void whenAddingProteinTotalIncreasesByThatAmount(){

        service.addProtein(10);
        //assertEquals(10, service.getTotal());
        //assertThat(service.getTotal(), is(10));

        assertThat(service.getTotal(), allOf(is(10), instanceOf(Integer.class)));
    }

    @Test
    // @Ignore
    @Category(GoodTestCategory.class)
    public void whenRemovingProteinTotalRemainsZero(){

        service.removeProtein(5);
        assertEquals(0, service.getTotal());
    }

    @Rule
    public ExpectedException thrown=ExpectedException.none();
    // @Test(expected=InvalidGoalException.class)
    @Test
    public void whenGoalIsSetToLessThanZeroExceptionThrown() throws
InvalidGoalException{
        thrown.expect(InvalidGoalException.class);
        //thrown.expectMessage("!Goal is less than zero!");
        thrown.expectMessage(containsString("Goal"));

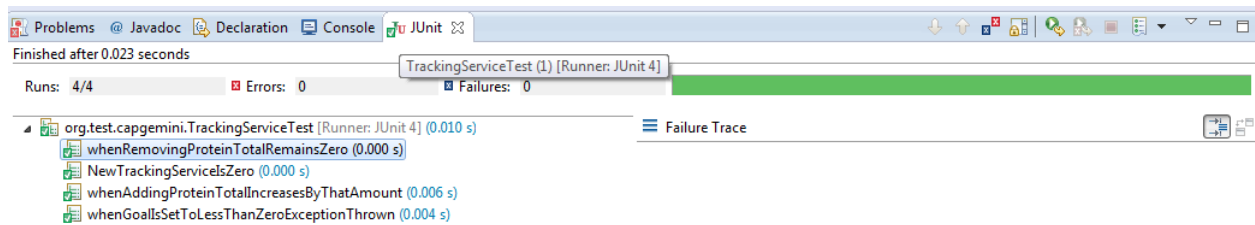
        service.setGoal(-5);
    }

```

```
/*
@Test(timeout=200)
public void BadTest(){
    for(int i=0;i<10000000;i++)
        service.addProtein(1);
}*/
}
```

## Output

Right Click the ProteinTrackerSuite → Run As → JUnit Test



## Learning:

From the above example we learnt advanced exception testing.

## LAB 1.11

*Explain the rules in JUnit*

**Steps:**

- Create new java project “**ProteinTracker**” in SpringToolSuite.
- Create new Package in the name of org.capgemini. And then add the following classes.

**Code:****HistoryItem.java**

```
package org.capgemini;

import java.util.ArrayList;
import java.util.List;

public class TrackingService {
    private int total;
    private int goal;
    private List<HistoryItem> history=new ArrayList<HistoryItem>();
    private int historyId=0;

    public void addProtein(int amount){
        total+=amount;
        history.add(new HistoryItem(historyId++,amount,"add",total));
    }

    public void removeProtein(int amount){
        total-=amount;
        if(total<0)
            total=0;
        history.add(new
HistoryItem(historyId++,amount,"subtract",total));

    }

    public int getTotal() {
        return total;
    }

    public void setTotal(int total) {
        goal = total;
    }

    public boolean isGoalMet(){
```

```
        return total>=goal;
    }

    public int getGoal() {
        return goal;
    }

    public void setGoal(int goal) throws InvalidGoalException {
        if(goal<0)
            throw new InvalidGoalException("Goal is less than
zero");
        this.goal = goal;
    }

    public List<HistoryItem> getHistory() {
        return history;
    }

    public void setHistory(List<HistoryItem> history) {
        this.history = history;
    }

    public int getHistoryId() {
        return historyId;
    }

    public void setHistoryId(int historyId) {
        this.historyId = historyId;
    }
}
```

#### InvalidGoalException.java

```
package org.capgemini;

public class InvalidGoalException extends Exception {
    public InvalidGoalException(String msg){
        super(msg);
    }
}
```



- Create new java project “ProteinTrackerAdvTests” in SpringToolSuite.
- Create new Package in the name of org.test.capgemini. And then add the following classes.

**TrackingServiceTest.java**

```
package org.test.capgemini;
import org.capgemini.InvalidGoalException;
import org.capgemini.TrackingService;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Ignore;
import org.junit.Test;
import org.junit.experimental.categories.Category;

import static org.junit.Assert.assertEquals;
import static org.hamcrest.CoreMatchers.*;
import static org.junit.matchers.JUnitMatchers.*;

import static org.junit.Assert.*;

public class TrackingServiceTest {

    private TrackingService service;

    @BeforeClass
    public static void before(){
        System.out.println("Before Class");
    }

    @AfterClass
    public static void after(){
        System.out.println("After Class");
    }

    @Before
    public void setUp(){
        System.out.println("Before");
        service=new TrackingService();
    }

    @After
    public void tearDown(){
        System.out.println("After");
    }
}
```

```

    }

    @Test
    @Category({GoodTestCategory.class, BadTestCategory.class})
    public void NewTrackingServiceIsZero(){

        assertEquals("Tracking Service total was not zero", 0,
service.getTotal());
    }

    @Test
    @Category(GoodTestCategory.class)
    public void whenAddingProteinTotalIncreasesByThatAmount(){

        service.addProtein(10);
        //assertEquals(10, service.getTotal());
        //assertThat(service.getTotal(), is(10));

        assertThat(service.getTotal(), allOf(is(10), instanceOf(Integer.class)));
    }

    @Test
    // @Ignore
    @Category(GoodTestCategory.class)
    public void whenRemovingProteinTotalRemainsZero(){

        service.removeProtein(5);
        assertEquals(0, service.getTotal());
    }

    @Rule
    public ExpectedException thrown=ExpectedException.none();
    // @Test(expected=InvalidGoalException.class)
    @Test
    public void whenGoalIsSetToLessThanZeroExceptionThrown() throws
InvalidGoalException{
        thrown.expect(InvalidGoalException.class);
        //thrown.expectMessage("!Goal was less than zero!");
        thrown.expectMessage(containsString("Goal"));

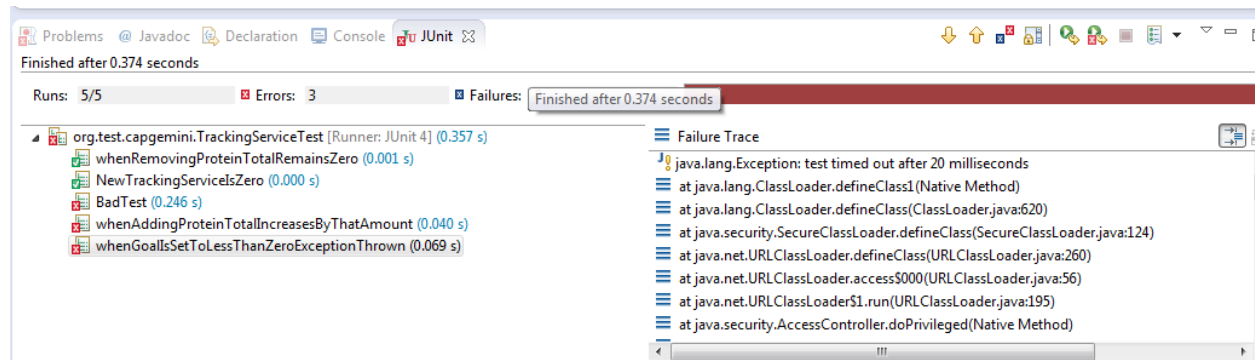
        service.setGoal(-5);
    }

```

```
@Rule
public Timeout timeout=new Timeout(20);
//@Test(timeout=200)
@Test
public void BadTest(){
    for(int i=0;i<10000000;i++)
        service.addProtein(1);
}
```

### Output

Right Click the ProteinTrackerSuite → Run As → JUnit Test



### Learning:

From the above example we learnt how to define the rules.

## LAB 1.12

*Explain ConsoleRunner*

**Steps:**

- Create new java project “**ProteinTracker**” in SpringToolSuite.
- Create new Package in the name of org.capgemini. And then add the following classes.

**Code:****HistoryItem.java**

```
package org.capgemini;

import java.util.ArrayList;
import java.util.List;

public class TrackingService {
    private int total;
    private int goal;
    private List<HistoryItem> history=new ArrayList<HistoryItem>();
    private int historyId=0;

    public void addProtein(int amount){
        total+=amount;
        history.add(new HistoryItem(historyId++,amount,"add",total));
    }

    public void removeProtein(int amount){
        total-=amount;
        if(total<0)
            total=0;
        history.add(new
HistoryItem(historyId++,amount,"subtract",total));

    }

    public int getTotal() {
        return total;
    }

    public void setTotal(int total) {
        goal = total;
    }

    public boolean isGoalMet(){
```

```
        return total>=goal;
    }

    public int getGoal() {
        return goal;
    }

    public void setGoal(int goal) throws InvalidGoalException {
        if(goal<0)
            throw new InvalidGoalException("Goal is less than
zero");
        this.goal = goal;
    }

    public List<HistoryItem> getHistory() {
        return history;
    }

    public void setHistory(List<HistoryItem> history) {
        this.history = history;
    }

    public int getHistoryId() {
        return historyId;
    }

    public void setHistoryId(int historyId) {
        this.historyId = historyId;
    }
}
```

#### InvalidGoalException.java

```
package org.capgemini;

public class InvalidGoalException extends Exception {
    public InvalidGoalException(String msg){
        super(msg);
    }
}
```

- Create new java project “ProteinTrackerAdvTests” in SpringToolSuite.
- Create new Package in the name of org.test.capgemini. And then add the following classes.

**TrackingServiceTest.java**

```
package org.test.capgemini;
import org.capgemini.InvalidGoalException;
import org.capgemini.TrackingService;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Ignore;
import org.junit.Test;
import org.junit.experimental.categories.Category;

import static org.junit.Assert.assertEquals;
import static org.hamcrest.CoreMatchers.*;
import static org.junit.matchers.JUnitMatchers.*;

import static org.junit.Assert.*;

public class TrackingServiceTest {

    private TrackingService service;

    @BeforeClass
    public static void before(){
        System.out.println("Before Class");
    }

    @AfterClass
    public static void after(){
        System.out.println("After Class");
    }

    @Before
    public void setUp(){
        System.out.println("Before");
        service=new TrackingService();
    }

    @After
    public void tearDown(){
        System.out.println("After");
    }
}
```

```

    }

    @Test
    @Category({GoodTestCategory.class, BadTestCategory.class})
    public void NewTrackingServiceIsZero(){

        assertEquals("Tracking Service total was not zero", 0,
service.getTotal());
    }

    @Test
    @Category(GoodTestCategory.class)
    public void whenAddingProteinTotalIncreasesByThatAmount(){

        service.addProtein(10);
        //assertEquals(10, service.getTotal());
        //assertThat(service.getTotal(), is(10));

        assertThat(service.getTotal(), allOf(is(10), instanceOf(Integer.class)));
    }

    @Test
    // @Ignore
    @Category(GoodTestCategory.class)
    public void whenRemovingProteinTotalRemainsZero(){

        service.removeProtein(5);
        assertEquals(0, service.getTotal());
    }

    @Rule
    public ExpectedException thrown=ExpectedException.none();
    // @Test(expected=InvalidGoalException.class)
    @Test
    public void whenGoalIsSetToLessThanZeroExceptionThrown() throws
InvalidGoalException{
        thrown.expect(InvalidGoalException.class);
        //thrown.expectMessage("!Goal was less than zero!");
        thrown.expectMessage(containsString("Goal"));

        service.setGoal(-5);
    }

```

```

@Rule
public Timeout timeout=new Timeout(20);
//@Test(timeout=200)
@Test
public void BadTest(){
    for(int i=0;i<10000000;i++)
        service.addProtein(1);
}
}

```

Create new class called ConsoleRunner under the package org.test.capgemini. And then run the class  
Run As → Java Application

```

package org.test.capgemini;

import org.junit.internal.TextListener;
import org.junit.runner.JUnitCore;

public class ConsoleRunner {

    public static void main(String[] args) {
        JUnitCore junit=new JUnitCore();

        junit.addListener(new TextListener(System.out));

        junit.run(TrackingServiceTest.class);
    }
}

```

### Output

Right Click the ConsoleRunner →Run As → Java Application

```

Before Class
.Before
After
.Before
After
.Before
After
E.Before
After
.Before
After
After Class

```



```
Time: 1.06
There was 1 failure:
1) BadTest(org.test.capgemini.TrackingServiceTest)
java.lang.OutOfMemoryError: Java heap space
    at java.util.Arrays.copyOf(Arrays.java:2760)
    at java.util.Arrays.copyOf(Arrays.java:2734)
    at java.util.ArrayList.ensureCapacity(ArrayList.java:167)
    at java.util.ArrayList.add(ArrayList.java:351)
    at
org.test.capgemini.TrackingService.addProtein(TrackingService.java:15)
    at
org.test.capgemini.TrackingServiceTest.BadTest(TrackingServiceTest.java:89)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
    at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
    at java.lang.reflect.Method.invoke(Method.java:597)
    at
org.junit.runners.model.FrameworkMethod$1.runReflectiveCall(FrameworkMethod.java:47)
    at
org.junit.internal.runners.model.ReflectiveCallable.run(ReflectiveCallable.java:12)
    at
org.junit.runners.model.FrameworkMethod.invokeExplosively(FrameworkMethod.java:44)
    at
org.junit.internal.runners.statements.InvokeMethod.evaluate(InvokeMethod.java:17)
    at
org.junit.internal.runners.statements.RunBefores.evaluate(RunBefores.java:26)
    at
org.junit.internal.runners.statements.RunAfters.evaluate(RunAfters.java:27)
    at
org.junit.rules.ExpectedException$ExpectedExceptionStatement.evaluate(ExpectedException.java:168)
    at org.junit.rules.RunRules.evaluate(RunRules.java:20)
```

```
    at org.junit.runners.ParentRunner.runLeaf(ParentRunner.java:271)
    at
org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:70)
    at
org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:50)
    at org.junit.runners.ParentRunner$3.run(ParentRunner.java:238)
    at
org.junit.runners.ParentRunner$1.schedule(ParentRunner.java:63)
    at
org.junit.runners.ParentRunner.runChildren(ParentRunner.java:236)
    at
org.junit.runners.ParentRunner.access$000(ParentRunner.java:53)
    at
org.junit.runners.ParentRunner$2.evaluate(ParentRunner.java:229)
    at
org.junit.internal.runners.statements.RunBefores.evaluate(RunBefores.java:26)
    at
org.junit.internal.runners.statements.RunAfters.evaluate(RunAfters.java:27)
    at org.junit.runners.ParentRunner.run(ParentRunner.java:309)
    at org.junit.runners.Suite.runChild(Suite.java:127)
    at org.junit.runners.Suite.runChild(Suite.java:26)
    at org.junit.runners.ParentRunner$3.run(ParentRunner.java:238)

FAILURES!!!
Tests run: 5,  Failures: 1
```

#### Learning:

From the above example we learnt how to show the test report in the console screen.

Run Junit TestCases in CommandPrompt:

```
D:\vidavid\JUnit\jars>java -cp C:\Utilities\*;D:\vidavid\JUnit\jars\ProteinTracker.jar
org.junit.runner.JUnitCore org.test.capgemini.TrackingServiceTest
```

Ant Build Tool JUNIT test

1. RightClick ProteinTrackerAdvTests→Export→Ant BuildFiles
2. You will get build.xml file
3. Right click → run As→Ant Build  
(OR)
4. Right click → run As→External Tool Configurations
5. In the target tab→Select the class file and run

JUNIT Report Bug

1. RightClick ProteinTrackerAdvTests→Export→Ant BuildFiles
2. You will get build.xml file
3. Right click → run As→Ant Build  
(OR)
4. Right click → run As→External Tool Configurations
5. In the target tab→Select the junitreport and run

Creating Report

1. Go to the following URL <http://ant.apache.org/bindownload.cgi>
2. Download apache-ant-1.9.4-bin.zip folder
3. Unzip and put it under c:\Utilities\
4. Open cmd prompt
5. Go to the c:\Utilities\ apache-ant-1.9.4-bin\bin

```
C:\Utilities\apache-ant-1.9.4-bin\apache-ant-1.9.4\bin>ant
Buildfile: build.xml does not exist!
Build failed

C:\Utilities\apache-ant-1.9.4-bin\apache-ant-1.9.4\bin>
```

- 6.
7. Type the following:  
ant -buildfile D:\vidavid\MyWork\JUnit\ProteinTrackerAdvTests\build.xml

```
C:\Utilities\apache-ant-1.9.4-bin\apache-ant-1.9.4\bin>ant -buildfile D:\vidavid\MyWork\JUnit\ProteinTrackerAdvTests\build.xml
Buildfile: D:\vidavid\MyWork\JUnit\ProteinTrackerAdvTests\build.xml

build-subprojects:
init:
build-project:
[echo] ProteinTracker: D:\vidavid\MyWork\JUnit\ProteinTracker\build.xml
init:
build-project:
[echo] ProteinTrackerAdvTests: D:\vidavid\MyWork\JUnit\ProteinTrackerAdvTests\build.xml
build:
BUILD SUCCESSFUL
Total time: 0 seconds
C:\Utilities\apache-ant-1.9.4-bin\apache-ant-1.9.4\bin>
```

8.

9. Type the following:

ant -buildfile D:\vidavid\MyWork\JUnit\ProteinTrackerAdvTests\build.xml junitreport

```
C:\Utilities\apache-ant-1.9.4-bin\apache-ant-1.9.4\bin>ant -buildfile D:\vidavid\MyWork\JUnit\ProteinTrackerAdvTests\build.xml junitreport
Buildfile: D:\vidavid\MyWork\JUnit\ProteinTrackerAdvTests\build.xml

junitreport:
[junitreport] Processing D:\vidavid\MyWork\JUnit\ProteinTrackerAdvTests\junit\TESTS-TestSuites.xml to C:\Users\vidavid\AppData\Local\Temp\null1839441025
[junitreport] Loading stylesheet jar:file:/C:/Utilities/apache-ant-1.9.4-bin/apache-ant-1.9.4/lib/ant-junit.jar!/org/apache/tools/ant/taskdefs/optional/junit/xsl/junit-frames.xsl
[junitreport] Transform time: 344ms
[junitreport] Deleting: C:\Users\vidavid\AppData\Local\Temp\null1839441025

BUILD SUCCESSFUL
Total time: 0 seconds
C:\Utilities\apache-ant-1.9.4-bin\apache-ant-1.9.4\bin>
```

10. Now Just open the following file, this is generated by junitreport

<file:///D:/vidavid/MyWork/JUnit/ProteinTrackerAdvTests/junit/index.html>

### JUnit And Maven

1. RightClick the project →Run As→Maven Install
2. Converting to Maven project
3. Right Click Protein Tracker → Configure→Maven Project
4. Right Click Protein TrackerAdvTests → Configure→Maven Project
5. Click the pom.xml file under

[Cobertura.sourceforge.net/introduction.html](http://cobertura.sourceforge.net/introduction.html)

## LAB 1.13

*Explain how to test dependencies in JUnit*

**Steps:**

- Create new java project “**ProteinTracker**” in SpringToolSuite.
- Create new Package in the name of org.capgemini. And then add the following classes.

**Code:**

```
package org.capgemini;

public class HistoryItem {
    private final int id;
    private final int amount;
    private final String operation;
    private final int total;
    public int getId() {
        return id;
    }
    public int getAmount() {
        return amount;
    }
    public String getOperation() {
        return operation;
    }
    public int getTotal() {
        return total;
    }
    public HistoryItem(int id, int amount, String operation, int
total) {

        this.id = id;
        this.amount = amount;
        this.operation = operation;
        this.total = total;
    }
}

package org.capgemini;

public class InvalidGoalException extends Exception {

    public InvalidGoalException(String msg){
```

```
        super(msg);
    }
}
package org.capgemini;

public interface Notifier {

    boolean send(String msg);
}
package org.capgemini;

public class NotifierStub implements Notifier {

    @Override
    public boolean send(String msg) {

        return true;
    }
}

package org.capgemini;

import java.util.ArrayList;
import java.util.List;

public class TrackingService {
    private int total;
    private int goal;
    private List<HistoryItem> history=new ArrayList<HistoryItem>();
    private int historyId=0;
    private Notifier notifier;

    public TrackingService(Notifier notifier){
        this.notifier=notifier;
    }

    public void addProtein(int amount){
        total+=amount;
        history.add(new HistoryItem(historyId++,amount,"add",total));
    }
}
```

```
        if(total>goal){
            boolean sendResult=notifier.send("goal met");
            String historyMsg="sent:goal met";
            if(!sendResult)
                historyMsg="sent_error:goal met";
            history.add(new HistoryItem(historyId++, amount, historyMsg, total));
        }
    }

    public void removeProtein(int amount){
        total-=amount;
        if(total<0)
            total=0;
        history.add(new HistoryItem(historyId++,amount,"subtract",total));
    }

    public int getTotal() {
        return total;
    }

    public void setTotal(int total) {
        goal = total;
    }

    public boolean isGoalMet(){
        return total>=goal;
    }

    public int getGoal() {
        return goal;
    }

    public void setGoal(int goal) throws InvalidGoalException {
        if(goal<0)
            throw new InvalidGoalException("Goal is less than zero");
    }
}
```



```

        this.goal = goal;
    }

    public List<HistoryItem> getHistory() {
        return history;
    }

    public void setHistory(List<HistoryItem> history) {
        this.history = history;
    }

    public int getHistoryId() {
        return historyId;
    }

    public void setHistoryId(int historyId) {
        this.historyId = historyId;
    }
}

```

- Create new java project “**ProteinTrackerAdvTests**” in SpringToolSuite.
- Create new Package in the name of org.test.capgemini. And then add the following classes.

```

package org.test.capgemini;

public interface BadCategory {
}

package org.test.capgemini;

import org.junit.internal.TextListener;
import org.junit.runner.JUnitCore;

public class ConsoleRunner {

    public static void main(String[] args) {
        JUnitCore junit=new JUnitCore();

        junit.addListener(new TextListener(System.out));
    }
}

```

```
        junit.run(TrackingServiceTest.class);
    }

} package org.test.capgemini;

public interface GoodTestCategory {
}
package org.test.capgemini;

import org.junit.experimental.categories.Categories;
import org.junit.experimental.categories.Categories.ExcludeCategory;
import org.junit.experimental.categories.Categories.IncludeCategory;
import org.junit.runner.RunWith;
import org.junit.runners.Suite;

@RunWith(Categories.class)
@IncludeCategory(GoodTestCategory.class)
@ExcludeCategory(BadCategory.class)
@Suite.SuiteClasses({
    HelloJUnitTests.class,
    TrackingServiceTest.class
})
public class GoodTestSuite {

} package org.test.capgemini;
import static org.junit.Assert.*;

import org.junit.Test;
import org.junit.experimental.categories.Category;

@Category(GoodTestCategory.class)
public class HelloJUnitTests {

    @Test
    public void test() {
        //fail("Not yet implemented");
    }

} package org.test.capgemini;
```

```
import java.util.Arrays;
import java.util.List;

import org.capgemini.NotifierStub;
import org.capgemini.TrackingService;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.junit.runners.Parameterized;
import org.junit.runners.Parameterized.Parameters;

import static org.junit.Assert.*;

@RunWith(Parameterized.class)
public class ParameterizedTest {

    private static TrackingService service=new TrackingService(new NotifierStub());
    private int input;
    private int expected;
    @Parameters
    public static List<Object[]> data(){
        return Arrays.asList(new Object[][]{
            {5,5},
            {5,10},
            {-12,0},
            {50,50},
            {1,51}
        });
    }

    public ParameterizedTest(int input,int expected){
        this.input=input;
        this.expected=expected;
    }

    @Test
    public void test(){
```

```
        if(input>=0)
            service.addProtein(input);
        else
            service.removeProtein(-input);

        assertEquals(expected,service.getTotal());
    }

} package org.test.capgemini;

import org.junit.runner.RunWith;
import org.junit.runners.Suite;

@RunWith(Suite.class)
@Suite.SuiteClasses({
    HelloJUnitTests.class,
    TrackingServiceTest.class
})
public class ProteinTrackerSuite {

}

package org.test.capgemini;
import org.capgemini.HistoryItem;
import org.capgemini.InvalidGoalException;
import org.capgemini.NotifierStub;
import org.capgemini.TrackingService;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Ignore;
import org.junit.Rule;
import org.junit.Test;
import org.junit.experimental.categories.Category;
import org.junit.rules.ExpectedException;
import org.junit.rules.Timeout;

import static org.junit.Assert.assertEquals;
import static org.hamcrest.CoreMatchers.*;
import static org.junit.matchers.JUnitMatchers.containsString;
```

```
import static org.junit.Assert.*;
public class TrackingServiceTest {

    private TrackingService service;

    @BeforeClass
    public static void before(){
        System.out.println("Before Class");
    }

    @AfterClass
    public static void after(){
        System.out.println("After Class");
    }

    @Before
    public void setUp(){
        System.out.println("Before");
        service=new TrackingService(new NotifierStub());
    }

    @After
    public void tearDown(){
        System.out.println("After");
    }

    @Test
    @Category({GoodTestCategory.class,BadCategory.class})
    public void NewTrackingServiceIsZero(){

        assertEquals("Tracking Service total was not zero",0, service.getTotal());
    }

    @Test
    public void whenGoalsMetUpdateHistory() throws InvalidGoalException{
        service.setGoal(5);
        service.addProtein(6);

        HistoryItem result=service.getHistory().get(1);
    }
}
```

```
assertEquals("sent:goal met",result.getOperation() );
}

@Test
@Category(GoodTestCategory.class)
public void whenAddingProteinTotalIncreasesByThatAmount(){

    service.addProtein(10);
    //assertEquals(10 ,service.getTotal());
    //assertThat(service.getTotal(),is(10));
    assertThat(service.getTotal(),allOf(is(10),instanceOf(Integer.class)));
}

@Test
//@Ignore
@Category(GoodTestCategory.class)
public void whenRemovingProteinTotalRemainsZero(){

    service.removeProtein(5);
    assertEquals(0 ,service.getTotal());
}

@Rule
public ExpectedException thrown=ExpectedException.none();
//@Test(expected=InvalidGoalException.class)
@Test
public void whenGoallsSetToLessThanZeroExceptionThrown() throws InvalidGoalException{
    thrown.expect(InvalidGoalException.class);
    //thrown.expectMessage("!Goal was less than zero!");
    thrown.expectMessage(containsString("Goal"));

    service.setGoal(-5);
}

/*@Rule
public Timeout timeout=new Timeout(20);
//@Test(timeout=200)
@Test
```

```

        public void BadTest(){
            for(int i=0;i<10000000;i++)
                service.addProtein(1);
        }*/
    }
package org.test.capgemini;

import org.capgemini.NotifierStub;
import org.capgemini.TrackingService;
import org.junit.Assume;
import org.junit.experimental.theories.DataPoints;
import org.junit.experimental.theories.Theories;
import org.junit.experimental.theories.Theory;
import org.junit.runner.RunWith;

import static org.junit.Assert.*;

@RunWith(Theories.class)
public class TrackingServiceTheories {

    @DataPoints
    public static int[] data(){
        return new int[]{
            1,5,10,15,20,50,-4
        };
    }

    @Theory
    public void positiveValuesShouldAlwaysHavePositiveTotals(int value){
        TrackingService service=new TrackingService(new NotifierStub());
        service.addProtein(value);

        Assume.assumeTrue(value>0);
        assertTrue(service.getTotal() > 0);
    }
}

```

**Output**

Right Click the ProteinTrackerSuite → Run As → JUnit Test

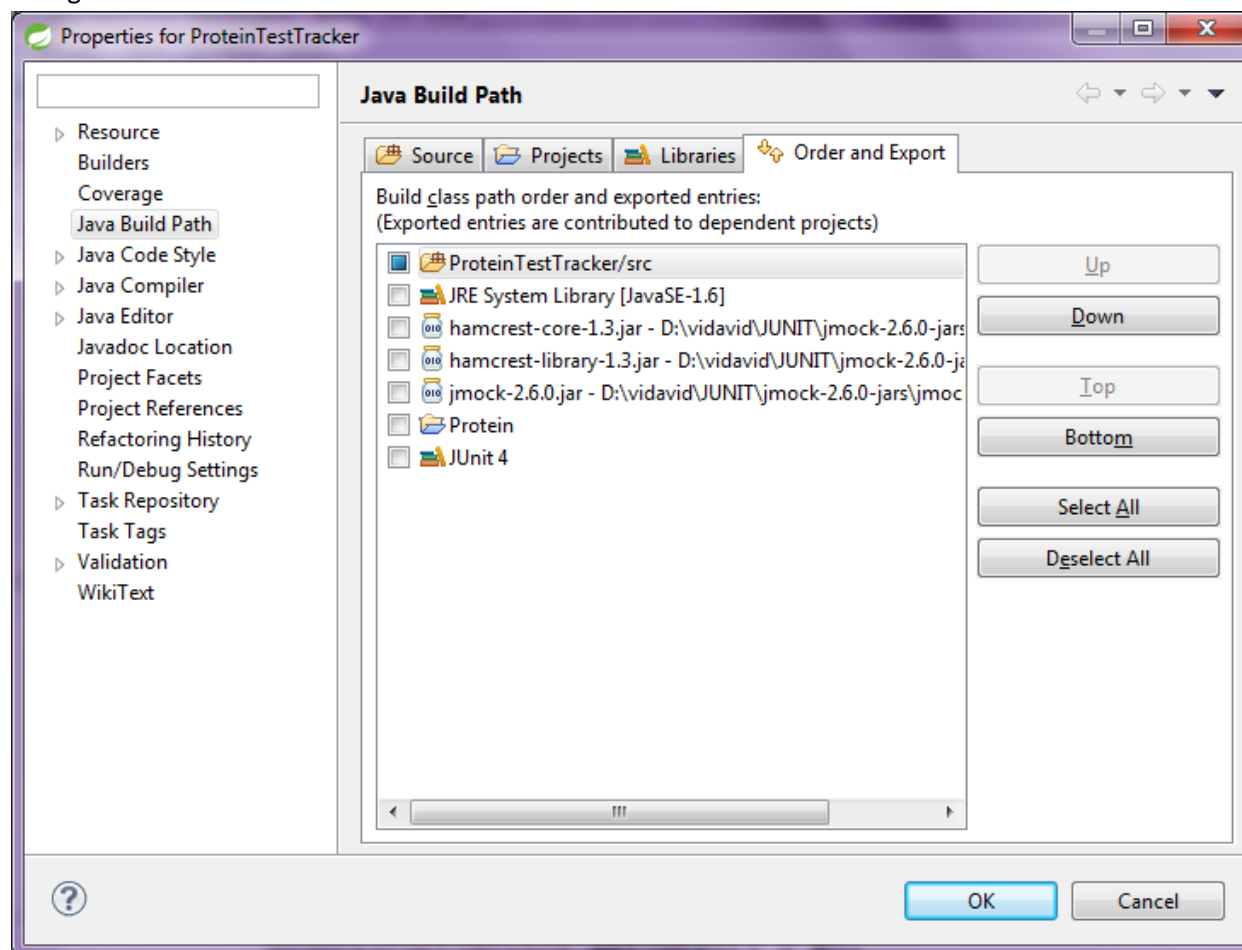
Learning:

## LAB 1.14

Test the dependencies with JMock in JUnit

## Steps:

- Create new java project “**ProteinTracker**” in SpringToolSuite.
- Create new Package in the name of org.capgemini. And then add the following classes.
- Go to Jmock.org → Get Jmock → Under 2.6.0 version → select Binary Jars
- Extract the zip folder
- Copy the following jars, and add it with the class path
- And then Configure → Configure Build path → Order and Export
  - Change order as follows





Code:

```
package org.capgemini;

public class HistoryItem {
    private final int id;
    private final int amount;
    private final String operation;
    private final int total;
    public int getId() {
        return id;
    }
    public int getAmount() {
        return amount;
    }
    public String getOperation() {
        return operation;
    }
    public int getTotal() {
        return total;
    }
    public HistoryItem(int id, int amount, String operation, int
total) {
        this.id = id;
        this.amount = amount;
        this.operation = operation;
        this.total = total;
    }
}

package org.capgemini;

public class InvalidGoalException extends Exception {

    public InvalidGoalException(String msg){
        super(msg);
    }
}

package org.capgemini;

public interface Notifier {
```

```
        boolean send(String msg);
    }
    package org.capgemini;

    public class NotifierStub implements Notifier {

        @Override
        public boolean send(String msg) {

            return true;
        }
    }

    package org.capgemini;

    import java.util.ArrayList;
    import java.util.List;

    public class TrackingService {
        private int total;
        private int goal;
        private List<HistoryItem> history=new ArrayList<HistoryItem>();
        private int historyId=0;
        private Notifier notifier;

        public TrackingService(Notifier notifier){
            this.notifier=notifier;
        }

        public void addProtein(int amount){
            total+=amount;
            history.add(new HistoryItem(historyId++,amount,"add",total));

            if(total>goal){
                boolean sendResult=notifier.send("goal met");
                String historyMsg="sent:goal met";
                if(!sendResult)
                    historyMsg="sent_error:goal met";

                history.add(new HistoryItem(historyId++, amount, historyMsg, total));
            }
        }
    }
}
```

```
    }  
  
    }  
  
    public void removeProtein(int amount){  
        total-=amount;  
        if(total<0)  
            total=0;  
        history.add(new HistoryItem(historyId++,amount,"subtract",total));  
  
    }  
  
    public int getTotal() {  
        return total;  
    }  
  
    public void setTotal(int total) {  
        goal = total;  
    }  
  
    public boolean isGoalMet(){  
        return total>=goal;  
    }  
  
    public int getGoal() {  
        return goal;  
    }  
  
    public void setGoal(int goal) throws InvalidGoalException {  
        if(goal<0)  
            throw new InvalidGoalException("Goal is less than zero");  
        this.goal = goal;  
    }  
  
    public List<HistoryItem> getHistory() {  
        return history;  
    }  
}
```

```

    public void setHistory(List<HistoryItem> history) {
        this.history = history;
    }

    public int getHistoryId() {
        return historyId;
    }

    public void setHistoryId(int historyId) {
        this.historyId = historyId;
    }
}

```

- Create new java project “**ProteinTrackerAdvTests**” in SpringToolSuite.
- Create new Package in the name of org.test.capgemini. And then add the following classes.

```

package org.test.capgemini;

public interface BadCategory {
}

package org.test.capgemini;

import org.junit.internal.TextListener;
import org.junit.runner.JUnitCore;

public class ConsoleRunner {

    public static void main(String[] args) {
        JUnitCore junit=new JUnitCore();

        junit.addListener(new TextListener(System.out));

        junit.run(TrackingServiceTest.class);
    }
}

package org.test.capgemini;

public interface GoodTestCategory {
}

```

```
}
package org.test.capgemini;

import org.junit.experimental.categories.Categories;
import org.junit.experimental.categories.Categories.ExcludeCategory;
import org.junit.experimental.categories.Categories.IncludeCategory;
import org.junit.runner.RunWith;
import org.junit.runners.Suite;

@RunWith(Categories.class)
@IncludeCategory(GoodTestCategory.class)
@ExcludeCategory(BadCategory.class)
@Suite.SuiteClasses({
    HelloJUnitTests.class,
    TrackingServiceTest.class
})
public class GoodTestSuite {

} package org.test.capgemini;
import static org.junit.Assert.*;

import org.junit.Test;
import org.junit.experimental.categories.Category;

@Category(GoodTestCategory.class)
public class HelloJUnitTests {

    @Test
    public void test() {
        //fail("Not yet implemented");
    }

} package org.test.capgemini;

import java.util.Arrays;
import java.util.List;

import org.capgemini.NotifierStub;
import org.capgemini.TrackingService;
```

```
import org.junit.Test;
import org.junit.runner.RunWith;
import org.junit.runners.Parameterized;
import org.junit.runners.Parameterized.Parameters;

import static org.junit.Assert.*;

@RunWith(Parameterized.class)
public class ParameterizedTest {

    private static TrackingService service=new TrackingService(new NotifierStub());
    private int input;
    private int expected;
    @Parameters
    public static List<Object[]> data(){
        return Arrays.asList(new Object[][]{
            {5,5},
            {5,10},
            {-12,0},
            {50,50},
            {1,51}
        });
    }

    public ParameterizedTest(int input,int expected){
        this.input=input;
        this.expected=expected;
    }

    @Test
    public void test(){
        if(input>=0)
            service.addProtein(input);
        else
            service.removeProtein(-input);

        assertEquals(expected,service.getTotal());
    }
}
```

```
    }

} package org.test.capgemini;

import org.junit.runner.RunWith;
import org.junit.runners.Suite;

@RunWith(Suite.class)
@Suite.SuiteClasses({
    HelloJUnitTests.class,
    TrackingServiceTest.class
})
public class ProteinTrackerSuite {

}

package org.test.capgemini;
import org.capgemini.HistoryItem;
import org.capgemini.InvalidGoalException;
import org.capgemini.NotifierStub;
import org.capgemini.TrackingService;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Ignore;
import org.junit.Rule;
import org.junit.Test;
import org.junit.experimental.categories.Category;
import org.junit.rules.ExpectedException;
import org.junit.rules.Timeout;

import static org.junit.Assert.assertEquals;
import static org.hamcrest.CoreMatchers.*;
import static org.junit.matchers.JUnitMatchers.containsString;
import static org.junit.Assert.*;
public class TrackingServiceTest {

    private TrackingService service;

    @BeforeClass
```

```
public static void before(){
    System.out.println("Before Class");
}

@AfterClass
public static void after(){
    System.out.println("After Class");
}

@Before
public void setUp(){
    System.out.println("Before");
    service=new TrackingService(new NotifierStub());
}

@After
public void tearDown(){
    System.out.println("After");
}

@Test
@Category({GoodTestCategory.class,BadCategory.class})
public void NewTrackingServiceIsZero(){

    assertEquals("Tracking Service total was not zero",0, service.getTotal());
}

@Test
public void whenGoalsMetUpdateHistory() throws InvalidGoalException{
    Mockery context = new Mockery();
    final Notifier mockNotifier = context.mock(Notifier.class);
    service = new TrackingService(mockNotifier);

    context.checking(new Expectations() {{
        oneOf(mockNotifier).send("goal met");
        will(returnValue(true));
    }});

    service.setGoal(5);
    service.addProtein(6);
}
```



```

HistoryItem result = service.getHistory().get(1);
assertEquals("sent:goal met", result.getOperation());

context.assertIsSatisfied();
}

```

```

@Test
@Category(GoodTestCategory.class)
public void whenAddingProteinTotalIncreasesByThatAmount(){

    service.addProtein(10);
    //assertEquals(10 ,service.getTotal());
    //assertThat(service.getTotal(),is(10));
    assertThat(service.getTotal(),allOf(is(10),instanceOf(Integer.class)));
}

@Test
//@Ignore
@Category(GoodTestCategory.class)
public void whenRemovingProteinTotalRemainsZero(){

    service.removeProtein(5);
    assertEquals(0 ,service.getTotal());
}

@Rule
public ExpectedException thrown=ExpectedException.none();
//@Test(expected=InvalidGoalException.class)
@Test
public void whenGoalsSetToLessThanZeroExceptionThrown() throws InvalidGoalException{
    thrown.expect(InvalidGoalException.class);
    //thrown.expectMessage("!Goal was less than zero!");
    thrown.expectMessage(containsString("Goal"));

    service.setGoal(-5);
}

/*@Rule

```

```
    public Timeout timeout=new Timeout(20);
    // @Test(timeout=200)
    @Test
    public void BadTest(){
        for(int i=0;i<10000000;i++)
            service.addProtein(1);
    }
}

package org.test.capgemini;

import org.capgemini.NotifierStub;
import org.capgemini.TrackingService;
import org.junit.Assume;
import org.junit.experimental.theories.DataPoints;
import org.junit.experimental.theories.Theories;
import org.junit.experimental.theories.Theory;
import org.junit.runner.RunWith;

import static org.junit.Assert.*;

@RunWith(Theories.class)
public class TrackingServiceTheories {

    @DataPoints
    public static int[] data(){
        return new int[]{
            1,5,10,15,20,50,-4
        };
    }

    @Theory
    public void positiveValuesShouldAlwaysHavePositiveTotals(int value){
        TrackingService service=new TrackingService(new NotifierStub());
        service.addProtein(value);

        Assume.assumeTrue(value>0);
        assertTrue(service.getTotal() > 0);
    }
}
```

**Output**

Right Click the ProteinTrackerSuite → Run As → JUnit Test

**Learning:**

We learnt the how to work with Mockery objects.

## LAB 1.15

*Explain how to test Theories in JUnit*

**Steps:**

- Create new java project “**ProteinTracker**” in SpringToolSuite.
- Create new Package in the name of org.capgemini. And then add the following classes.

**Code:**

```
package org.capgemini.testDemo;

import static org.junit.Assert.*;

import org.junit.Assume;
import org.junit.Test;
import org.junit.experimental.theories.DataPoints;
import org.junit.experimental.theories.Theories;
import org.junit.experimental.theories.Theory;
import org.junit.runner.RunWith;

@RunWith(Theories.class)
public class TestTheories {

    @DataPoints
    public static int[] mydatas(){
        return new int[]{
            1,2,4,-5,6
        };
    }

    @Theory
    public void theoryStatements(Integer a, Integer b){

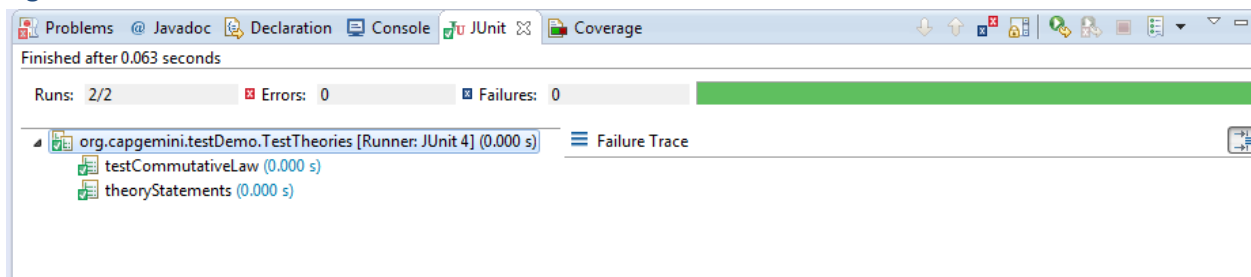
        System.out.println(a+"---->" +b);
        Assume.assumeTrue(a>0 && b>0);

        assertTrue((a+b)>0);
    }
}
```

```
@Theory
public void testCommutativeLaw(Integer a, Integer b){
    Assume.assumeTrue(a>0 && b>0);
    assertTrue(a+b == b+a);
}
}
```

### Output

Right Click the TestTheories → Run As → JUnit Test



### Learning:

We learnt How to test theories.

## LAB 1.16

*Explain how to test with Selenium products*

**Steps:**

- Create new java project “**MyApplicationTest**” in SpringToolSuite.
- Create new Package in the name of org.capgemini.testDemo And then add the following classes.
- Include **selenium-server-standalone-2.42.2.jar** under the class path.
- Then write the following classes

**Code:**

```
package org.capgemini.testDemo;

import static org.junit.Assert.*;
import org.junit.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;

public class SeleniumTest {

    @Test
    public void CanOpenGoogle() {
        WebDriver driver = new FirefoxDriver();
        driver.get("http://google.com");
        WebElement searchBox = driver.findElement(By.name("q"));
        searchBox.sendKeys("Capgemini");
        searchBox.submit();
    }
}
```

**Output**

Right Click the SeleniumTest → Run As → JUnit Test

1. It will open Firefox browser
2. It should change the url as <http://google.com>
3. And then it will open search box, in that searches for the “Capgemini” in the server.

**Learning:**

We learnt How to test Firefox Driver.

## LAB 1.17

*Explain how to do the integration Testing*

Steps:

- Create new java project “**Protein**” in SpringToolSuite.
- Create new Package in the name of org.capgemini And then add the following classes.
- Include the following jars under class path
  - **google-voice-java-1.6.jar**
  - **dom4j-1.6.1.jar**
  - **json.jar**
  - **jtidy.jar**
- Then write the following classes

Code:HistoryItem.java

```
package org.capgemini;
```

```
public class HistoryItem {  
    private final int id;  
    private final int amount;  
    private final String operation;  
    private final int total;  
    public int getId() {  
        return id;  
    }  
    public int getAmount() {  
        return amount;  
    }  
    public String getOperation() {  
        return operation;  
    }  
    public int getTotal() {  
        return total;  
    }  
    public HistoryItem(int id, int amount, String operation, int total) {  
  
        this.id = id;  
        this.amount = amount;  
        this.operation = operation;  
    }  
}
```

```
        this.total = total;
    }

}

TrackingService.java
package org.capgemini;

import java.util.ArrayList;
import java.util.List;

public class TrackingService {
    private int total;
    private int goal;
    private List<HistoryItem> history=new ArrayList<HistoryItem>();
    private int historyId=0;
    private Notifier notifier;

    public TrackingService(Notifier notifier){
        this.notifier=notifier;
    }

    public void addProtein(int amount){
        total+=amount;
        history.add(new HistoryItem(historyId++,amount,"add",total));

        if(total>goal){
            boolean sendResult=notifier.send("goal met");
            String historyMsg="sent:goal met";
            if(!sendResult)
                historyMsg="sent_error:goal met";

            history.add(new HistoryItem(historyId++, amount, historyMsg, total));
        }
    }

    public void removeProtein(int amount){
        total-=amount;
        if(total<0)
            total=0;
        history.add(new HistoryItem(historyId++,amount,"subtract",total));
    }
}
```



```
}

public int getTotal() {
    return total;
}

public void setTotal(int total) {
    goal = total;
}

public boolean isGoalMet(){
    return total>=goal;
}

public int getGoal() {
    return goal;
}

public void setGoal(int goal) throws InvalidGoalException {
    if(goal<0)
        throw new InvalidGoalException("Goal is less than zero");
    this.goal = goal;
}

public List<HistoryItem> getHistory() {
    return history;
}

public void setHistory(List<HistoryItem> history) {
    this.history = history;
}

public int getHistoryId() {
    return historyId;
}
```

```
        public void setHistoryId(int historyId) {
            this.historyId = historyId;
        }
    }

    InvalidGoalException.java
    package org.capgemini;

    public class InvalidGoalException extends Exception {

        public InvalidGoalException(String msg){
            super(msg);
        }
    }

    Notifier.java
    package org.capgemini;

    public interface Notifier {

        boolean send(String msg);

    }

    NotifierStub.java
    package org.capgemini;

    public class NotifierStub implements Notifier {

        @Override
        public boolean send(String msg) {

            return true;
        }

    }

    SMSNotifier.java
    package org.capgemini;

    import java.io.IOException;

    import com.techventus.server.voice.Voice;
```

```
public class SMSNotifier implements Notifier {

    private String userName;
    private String password;
    private String numberToMessage;

    public SMSNotifier(String userName, String password, String numberToMessage) {
        this.userName = userName;
        this.password = password;
        this.numberToMessage = numberToMessage;
    }

    @Override
    public boolean send(String message) {

        try {
            Voice voice = new Voice(userName, password);
            voice.sendSMS(numberToMessage, message);
        } catch (IOException e) {
            return false;
        }

        return true;
    }
}
```

#### IntegrationTest.java

```
package org.test.capgemini;
import static org.junit.Assert.assertTrue;
import java.io.IOException;
import org.capgemini.InvalidGoalException;
import org.capgemini.SMSNotifier;
import org.capgemini.TrackingService;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;
import com.techventus.server.voice.Voice;
import com.techventus.server.voice.datatypes.records.SMSThread;
```

```
public class IntegrationTest {

    private Voice voice;

    @Before
    public void setUp() throws IOException {
        voice = new Voice("pluralsightdemo@gmail.com", "pluralsight123", "2083522168");
    }

    @Test
    public void GoalMetShouldSendNotification() throws IOException, InvalidGoalException
    {
        TrackingService service = new TrackingService(new
        SMSNotifier("pluralsightdemo@gmail.com", "pluralsight123", "2083522168"));
        service.setGoal(50);
        service.addProtein(51);

        assertTrue(voice.getSMS().contains("goal met"));
    }

    @After
    public void tearDown() throws IOException {
        for(SMSThread thread : voice.getSMSThreads())
        {
            voice.deleteMessage(thread.getId());
        }
    }
}
```

**Output:**

Right Click the IntegrationTest → Run As → JUnit Test

**Learning:**

From the above example we learnt how to do the integration testing in JUnit.

## LAB 1.18

*Explain Mockito*

Steps:

- Create new java project “BankApp” in SpringToolSuite.
- Create new Package in the name of org.cap And then add the following classes.
- Then write the following classes

**Pom.xml**

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-
4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>org.cap.app</groupId>
    <artifactId>Day1-BankApp</artifactId>
    <version>0.0.1-SNAPSHOT</version>

    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.12</version>
            <scope>compile</scope>
        </dependency>
        <dependency>
            <groupId>org.mockito</groupId>
            <artifactId>mockito-all</artifactId>
            <version>1.10.19</version>
        </dependency>
    </dependencies>

    <build>
```

```
<plugins>
  <plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>3.5.1</version>
    <configuration>
      <!-- or whatever version you use -->
      <source>1.8</source>
      <target>1.8</target>
    </configuration>
  </plugin>
</plugins>

</build>

</project>

package org.cap.dto;

public class Account {

    private int accountNo;
    private Customer customer;
    private double amount;

    public Account(){}

    public Account(int accountNo, Customer customer, double amount) {
        super();
        this.accountNo = accountNo;
        this.customer = customer;
        this.amount = amount;
    }
    public int getAccountNo() {
        return accountNo;
    }
    public void setAccountNo(int accountNo) {
        this.accountNo = accountNo;
    }
}
```

```
public Customer getCustomer() {
    return customer;
}

public void setCustomer(Customer customer) {
    this.customer = customer;
}

public double getAmount() {
    return amount;
}

public void setAmount(double amount) {
    this.amount = amount;
}

@Override
public int hashCode() {
    final int prime = 31;
    int result = 1;
    result = prime * result + accountNo;
    long temp;
    temp = Double.doubleToLongBits(amount);
    result = prime * result + (int) (temp ^ (temp >>> 32));
    result = prime * result + ((customer == null) ? 0 : customer.hashCode());
    return result;
}

@Override
public boolean equals(Object obj) {
    if (this == obj)
        return true;
    if (obj == null)
        return false;
    if (getClass() != obj.getClass())
        return false;
    Account other = (Account) obj;
    if (accountNo != other.accountNo)
        return false;
    if (Double.doubleToLongBits(amount) !=
Double.doubleToLongBits(other.amount))
        return false;
```

```
        if (customer == null) {
            if (other.customer != null)
                return false;
        } else if (!customer.equals(other.customer))
            return false;
        return true;
    }

    @Override
    public String toString() {
        return "Account [accountNo=" + accountNo + ", customer=" + customer + ",
amount=" + amount + "]\n";
    }

}

package org.cap.dto;

public class Address {

    private String addressLine;

    public String getAddressLine() {
        return addressLine;
    }

    public void setAddressLine(String addressLine) {
        this.addressLine = addressLine;
    }

}

package org.cap.dto;

public class Customer {

    private String custName;
    private Address custAddress;
    public String getCustName() {
        return custName;
    }
}
```



```

    }
    public void setCustName(String custName) {
        this.custName = custName;
    }
    public Address getCustAddress() {
        return custAddress;
    }
    public void setCustAddress(Address custAddress) {
        this.custAddress = custAddress;
    }
}

}

package org.cap.service;

import org.cap.dto.Account;
import org.cap.dto.Customer;
import org.cap.exception.InsufficientBalanceException;
import org.cap.exception.InvalidInitialAmountException;

public interface AccountService {

    public Account addAccount(Customer customer, double amount) throws
InvalidInitialAmountException;

    public Account findAccountById(int accountNo);

    public Account withdraw(int accountNo, double amount) throws
InsufficientBalanceException;
    public Account deposit(int accountNo, double amount);
    public int addNumbers(int num1, int num2);
}

package org.cap.service;

import org.cap.dao.AccountDao;
import org.cap.dto.Account;
import org.cap.dto.Customer;
import org.cap.exception.InsufficientBalanceException;
import org.cap.exception.InvalidInitialAmountException;
import org.cap.util.AccountUtil;

```

```
public class AccountServiceImpl implements AccountService{

    private AccountDao accountDao;

    public AccountServiceImpl(){

    }

    public AccountServiceImpl(AccountDao accountDao) {
        super();
        this.accountDao = accountDao;
    }

    @Override
    public Account addAccount(Customer customer, double amount) throws
InvalidInitialAmountException {

        if(customer==null)
            throw new IllegalArgumentException("Customer Should not be
NULL.");

        if(amount<500)
            throw new InvalidInitialAmountException();

        Account account=new Account();
        account.setAccountNo(AccountUtil.generateAccountNumber());
        account.setCustomer(customer);
        account.setAmount(amount);

        if(accountDao.createAccount(account))
            return account;

        return null;
    }

    @Override
    public Account findAccountById(int accountNo) {

        return accountDao.findAccountById(accountNo);
    }
}
```

```
@Override
public Account withdraw(int accountNo, double amount) throws
InsufficientBalanceException {

    Account account=accountDao.findAccountById(accountNo);

    if(account.getAmount()<amount)
        throw new InsufficientBalanceException();

    account.setAmount(account.getAmount()-amount);

    return account;
}

@Override
public Account deposit(int accountNo, double amount) {
    Account account=accountDao.findAccountById(accountNo);

    account.setAmount(account.getAmount()+amount);

    return account;
}

@Override
public int addNumbers(int num1,int num2){
    return num1+num2;
}
```

```
}  
package org.cap.bankapp.test;  
  
import static org.junit.Assert.*;  
  
import org.cap.dao.AccountDao;  
import org.cap.dto.Account;  
import org.cap.dto.Address;  
import org.cap.dto.Customer;  
import org.cap.exception.InsufficientBalanceException;  
import org.cap.exception.InvalidInitialAmountException;  
import org.cap.service.AccountService;  
import org.cap.service.AccountServiceImpl;  
import org.junit.After;  
import org.junit.Before;  
import org.junit.Test;  
import org.mockito.Mock;  
import org.mockito.Mockito;  
import org.mockito.MockitoAnnotations;  
  
public class BankAppTestCase {  
  
    @Mock  
    private AccountDao accountDao;  
  
    private AccountService accountService;  
  
    @Before  
    public void init_method(){  
        //System.out.println("Before Method");  
        //accountService=new AccountServiceImpl();  
  
        MockitoAnnotations.initMocks(this);  
        accountService=new AccountServiceImpl(accountDao);  
    }  
}
```

```
@After
public void destroy_method(){
    //System.out.println("After Method");
}

@Test
public void test_addNumbers(){

    assertEquals(100, accountService.addNumbers(50, 50));
}

@Test(expected=IllegalArgumentException.class)
public void when_addAccount_with_null_customer_exception() throws
InvalidInitialAmountException{

    Customer customer=null;
    accountService.addAccount(customer, 900);

}

@Test(expected=InvalidInitialAmountException.class)
public void when_invalid_initial_amount_with_addAccount() throws
InvalidInitialAmountException{
    Customer customer=new Customer();
    customer.setCustName("Bryan");
    customer.setCustAddress(new Address());

    accountService.addAccount(customer, 300);
}

@Test
public void when_addAccount_with_valid_Data() throws
InvalidInitialAmountException{
    Account account=new Account();
    Customer customer=new Customer();
    customer.setCustName("Jerry");
    customer.setCustAddress(new Address());
```

```

        account.setCustomer(customer);
        account.setAccountNo(1);
        account.setAmount(1000);

        //declaration

Mockito.when(accountDao.createAccount(account)).thenReturn(true);

        //Actual Logic
        Account newAccount=accountService.addAccount(customer,
1000);

        //Verification
        Mockito.verify(accountDao).createAccount(account);

        assertEquals(newAccount.getAccountNo(),
account.getAccountNo());

    }

@Test
public void find_Account_By_Id_withDao(){

        Account account=new Account();
        Customer customer=new Customer();
        customer.setCustName("Tom");
        customer.setCustAddress(new Address());
        account.setAccountNo(2);
        account.setAmount(1000);

        //declaration
        Mockito.when(accountDao.findById(2)).thenReturn(account);

        //Actual Logic
        Account fAccount=accountService.findById(2);

        //Verification

```

```

Mockito.verify(accountDao).findAccountById(2);

//assertEquals(100, fAccount.getAmount(),0.00);
assertEquals(account.getAmount(), fAccount.getAmount(),0.00);

}

@Test
public void test_withdrawal_method() throws InsufficientBalanceException{
    Account account=new Account();
    Customer customer=new Customer();
    customer.setCustName("Tom");
    customer.setCustAddress(new Address());
    account.setAccountNo(2);
    account.setAmount(1000);

    //declaration
    Mockito.when(accountDao.findAccountById(2)).thenReturn(account);

    //Actual Logic
    //Account fAccount=accountService.findAccountById(2);
    Account modifiedAccount=accountService.withdraw(2, 300);

    //Verification
    Mockito.verify(accountDao).findAccountById(2);

    assertEquals(1700, modifiedAccount.getAmount(),0.0);

}

@Test
public void tes_depositMethod() throws InsufficientBalanceException{
    Account account=new Account();
    Customer customer=new Customer();
    customer.setCustName("Breyan");
    customer.setCustAddress(new Address());
    account.setAccountNo(201);
    account.setAmount(13000);

```

```
        //declaration
        Mockito.when(accountDao.findById(201)).thenReturn(account);

        //Actual Logic
        Account fAccount=accountService.findById(2);
        Account modifiedAccount=accountService.deposit(201, 3000);

        //Verification
        Mockito.verify(accountDao).findById(201);

        assertEquals(16000, modifiedAccount.getAmount(),0.0);
    }
}
```

**Output:**

Right Click the IntegrationTest → Run As → JUnit Test

**Learning:**

From the above example we learnt how to test dependencies with Mockito.