

سپهر مقیسه

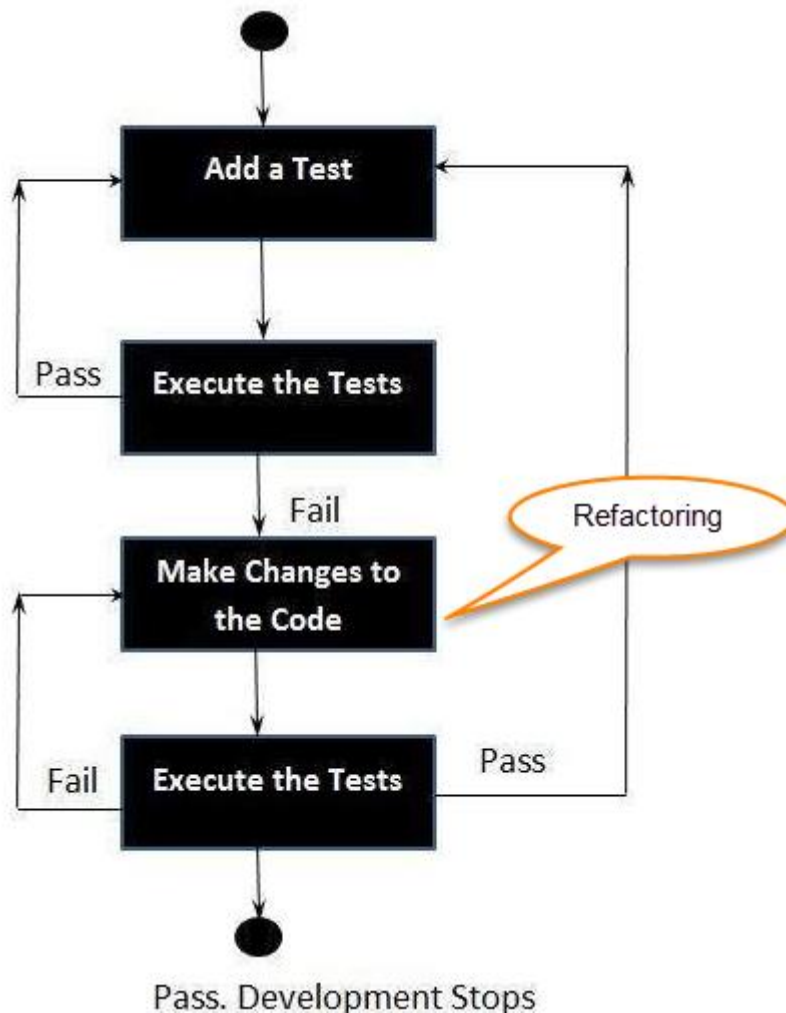
۹۸۳۱۱۰۳

تکلیف سه

درس مهندسی نرم افزار ۲

پاییز ۱۴۰۱

۱. در این نوع کدنویسی به این گونه عمل میکنیم که اول یک تست مینویسیم و سپس تا زمانی که آن تست موفق نبوده است به برطرف کردن و اضافه کردن خط کد دست میزنیم. به طور مثال در مرحله اول یک تست مینویسیم و منطقاً چون کدی برای پاس کردن تست ننوشتیم این تست با خطا رو به رو میشود و سپس کدی برای برطرف کردن تست مینویسیم و سپس بعد از قبولی آن تست بعدی را مینویسیم. این روش بسیار زمان برنامه نویسی را بهینه میکند چرا که تعداد خطا را کاهش میدهد و عمل خطایابی را آسان تر میکند و همچنین میتوان تا حد زیادی مطمئن بود با خروجی نا درست چه عکس العملی سیستم نشان میدهد و میتوان تمام جوانب را پیشبینی کرد.



به طور کلی چرخه این نوع برنامه نویسی به این صورت است که :

۱. یک تست مینویسیم

۲. آن را اجرا میکنیم

۳. تا زمانی که تست درست اجرا نشده کد را تغییر میدهیم.

۴. این عمل را تکرار میکنیم.

مثال کد ساده :

اول کلاس کد را نوشته و در آن به طوری شرط میزاریم که پسورد باید بین ۵ تا ۱۰ کاراکتر باشد

```
package Prac;

import org.testng.Assert;
import org.testng.annotations.Test;

public class TestPassword {
    @Test
    public void TestPasswordLength() {
        PasswordValidator pv = new PasswordValidator();
        Assert.assertEquals(true, pv.isValid("Abc123"));
    }
}
```

Needed for TestNG

We can not run test because this class is not created yet

This is main validation test

سپس برای اجراس تست تاییدکننده را مینویسیم:

```
package Prac;

public class PasswordValidator {
    public boolean isValid(String Password)
    {
        if (Password.length() >= 5 && Password.length() <= 10)
        {
            return true;
        }
        else
            return false;
    }
}
```

This is main condition checking length of password. If meets return true otherwise false.

خروجی به شکل زیر است

```
<terminated> TestPassword [TestNG] C:\Program Files\Java\jre1.8.0_77\bin\javaw.exe (Jul 25, 2016, 2:10:22 PM)
[TestNG] Running:
  C:\Users\kanchan\AppData\Local\Temp\testng-eclipse--571370159\testng-customsuite.xml
```

PASSED: TestPasswordLength

Result of test as Passed

```
=====
      Default test
      Tests run: 1, Failures: 0, Skips: 0
=====
```

```
=====
Default suite
Total tests run: 1, Failures: 0, Skips: 0
=====
```

```
[TestNG] Time taken by org.testng.reporters.EmailableReporter2@1b40d5f0: 202 ms
[TestNG] Time taken by org.testng.reporters.XMLReporter@28f67ac7: 63 ms
[TestNG] Time taken by org.testng.reporters.jq.Main@546a03af: 78 ms
[TestNG] Time taken by org.testng.reporters.JUnitReportReporter@5a01ccaa: 2 ms
[TestNG] Time taken by [FailedReporter passed=0 failed=0 skipped=0]: 1 ms
[TestNG] Time taken by org.testng.reporters.SuiteHTMLReporter@2b80d80f: 10 ms
```

۲.

بله معمولاً به صورت اتومات است

ابزار های زیادی برای این کار است که معمول ترین آن ها عبارتند از :

Dev mate-diffblue-ponicode-symflower

۳.

در devmate به کمک هوش مصنوعی این تست هارا اتومات میکند و به زبان C# و جاوا میتوان از ان استفاده کرد

در symflower این اتومات سازی با تمرکز بر امنیت به کار میرود. در ۶۰٪ از هزینه ها به کمک آن میتوان صرفه جویی کرد. و به عنوان یک برنامه مجزا میتوان از آن استفاده کرد. از زبان های جاوا و گولنگ پشتیبانی میکند.

Parasoft این اتومات سازی را به کمک پشتیبانی چند کامپایلر انجام میدهد و در زبان های C و C++ موجود است

یک نمونه دیگر برنامه ranorex است که برای تیم هایی که تجربه حرفه ای ندارند و به صورت تکنیکال کار را جلو نمیبرند مناسب است از ابزار مناسب در حالت ui پشتیبانی میکند به عنوان یک برنامه تحت ویندوز عمل میکند.

Smartbear دارای موتور تحت هوش مصنوعی است و از اپلیکیشن های تحت وب، موبایل و ویندوز پشتیبانی میکند. در زبان های اسکریپتی، جاوا اسکریپت و پایتون موجود است و تحت ویندوز عمل میکند.