2018

Michael Chrystal S0016497 Michael Christie S00151977

3/8/2018

**Software Quality & Testing Assignment**



https://github.com/s00151977/TestingPractice

Contents

[Flow graph & cyclomatic complexity of the program 0](#_Toc508904808)

[Test cases 100% branch coverage. 0](#_Toc508904809)

[NUnit automated white box tests 1](#_Toc508904810)

[Devise black box Tests 2](#_Toc508904811)

[Classes 2](#_Toc508904812)

[representative data 3](#_Toc508904813)

[Test Data 3](#_Toc508904814)

[Fitnesse automated acceptance testing server 5](#_Toc508904815)

[Use selenium IDE and webdriver system tests 6](#_Toc508904816)

[Katalon Class from test 6](#_Toc508904817)

[Katalon Screenshots 11](#_Toc508904818)

# Flow graph & cyclomatic complexity of the program

1

Cyclomatic Complexity= Edges-Nodes + 2

23-16+2=8

Michael Christie S00151977

Michael Chrystal S00164997

Female

Male

15

17

14

12

18

22

23

21

9

7

6

4

10

T

F

F

F

F

T

T

T

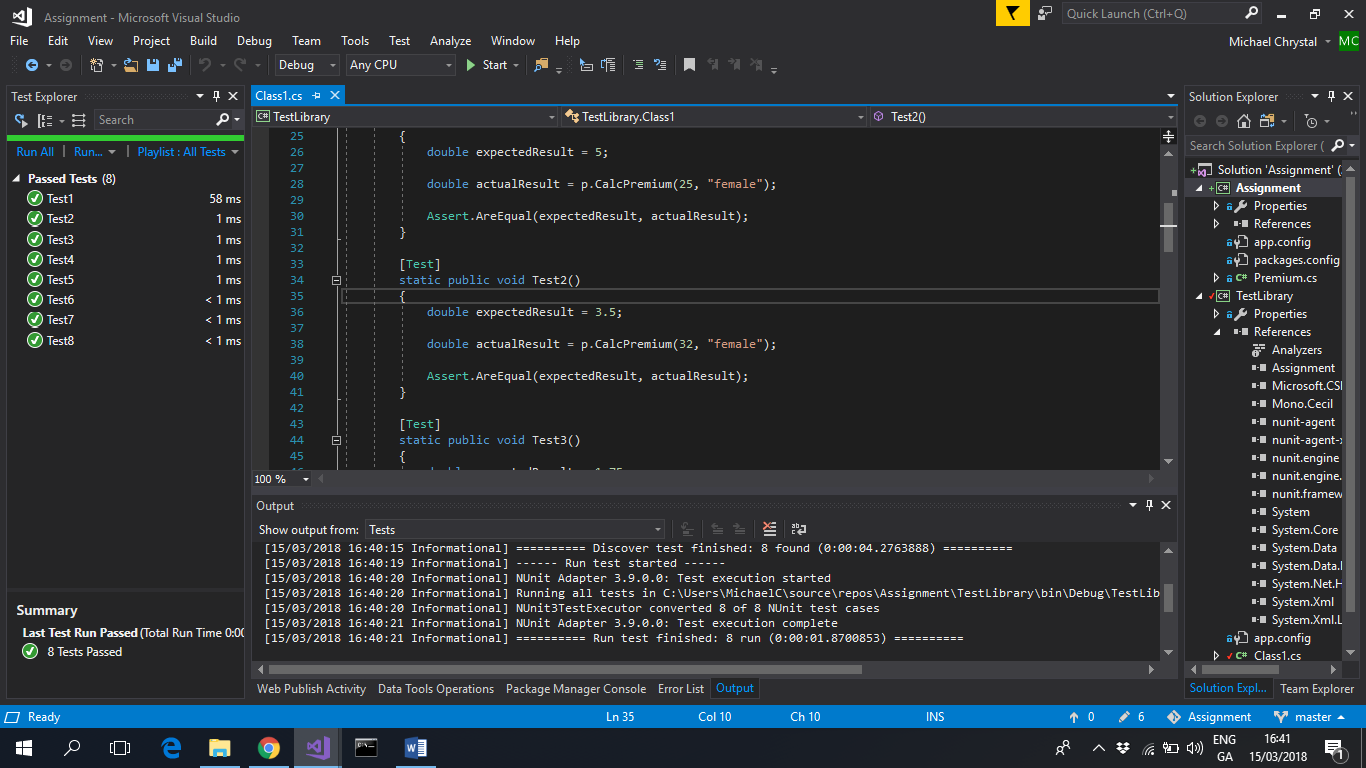
2

3

# Test cases 100% branch coverage.

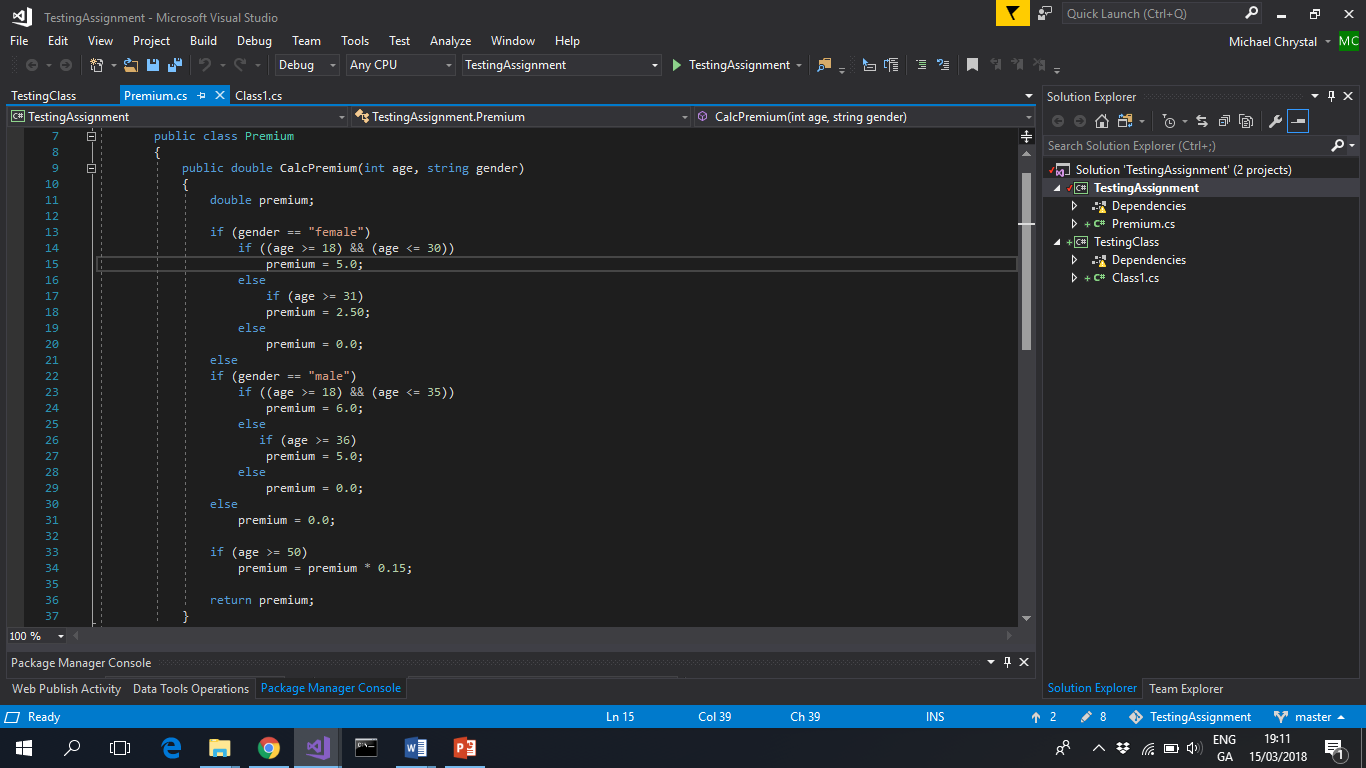
|  |  |  |  |
| --- | --- | --- | --- |
| Gender | Age | Actual Result | Expected Result |
| Female | 25 | 5 | 5 |
| Female | 32 | 3.5 | 3.5 |
| Female | 55 | 1.75 | 1.75 |
| Female | 15 | 0 | 0 |
| Male | 25 | 6 | 6 |
| Male | 37 | 5 | 5 |
| Male | 55 | 2.5 | 2.5 |
| Male | 15 | 0 | 0 |
| Pizza | 30 | 0 | 0 |

# **NUnit** automated white box tests



# Devise black box Tests

## Classes



## representative data



## Test Data

using NUnit.Framework;

using NUnit;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using SQTTAssignment;

namespace TestLibrary

{

[TestFixture]

static public class Class1

{

static Premium p;

[SetUp]

static public void init()

{

p = new Premium();

}

[Test]

static public void Test1()

{

double expectedResult = 5;

double actualResult = p.CalcPremium(25, "female");

Assert.AreEqual(expectedResult, actualResult);

}

[Test]

static public void Test2()

{

double expectedResult = 3.5;

double actualResult = p.CalcPremium(32, "female");

Assert.AreEqual(expectedResult, actualResult);

}

[Test]

static public void Test3()

{

double expectedResult = 1.75;

double actualResult = p.CalcPremium(55, "female");

Assert.AreEqual(expectedResult, actualResult);

}

[Test]

static public void Test4()

{

double expectedResult = 0;

double actualResult = p.CalcPremium(15, "female");

Assert.AreEqual(expectedResult, actualResult);

}

[Test]

static public void Test5()

{

double expectedResult = 6;

double actualResult = p.CalcPremium(25, "male");

Assert.AreEqual(expectedResult, actualResult);

}

[Test]

static public void Test6()

{

double expectedResult = 5;

double actualResult = p.CalcPremium(37, "male");

Assert.AreEqual(expectedResult, actualResult);

}

[Test]

static public void Test7()

{

double expectedResult = 2.5;

double actualResult = p.CalcPremium(55, "male");

Assert.AreEqual(expectedResult, actualResult);

}

[Test]

static public void Test8()

{

double expectedResult = 0;

double actualResult = p.CalcPremium(15, "male");

Assert.AreEqual(expectedResult, actualResult);

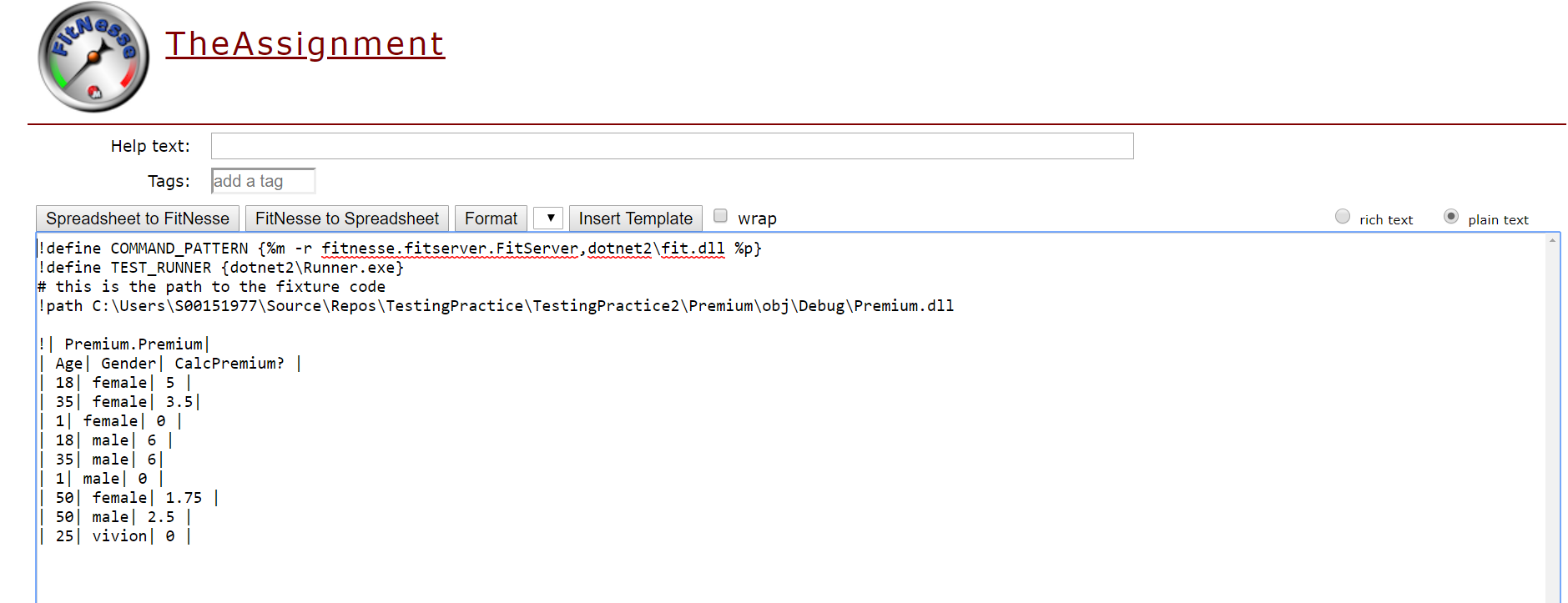
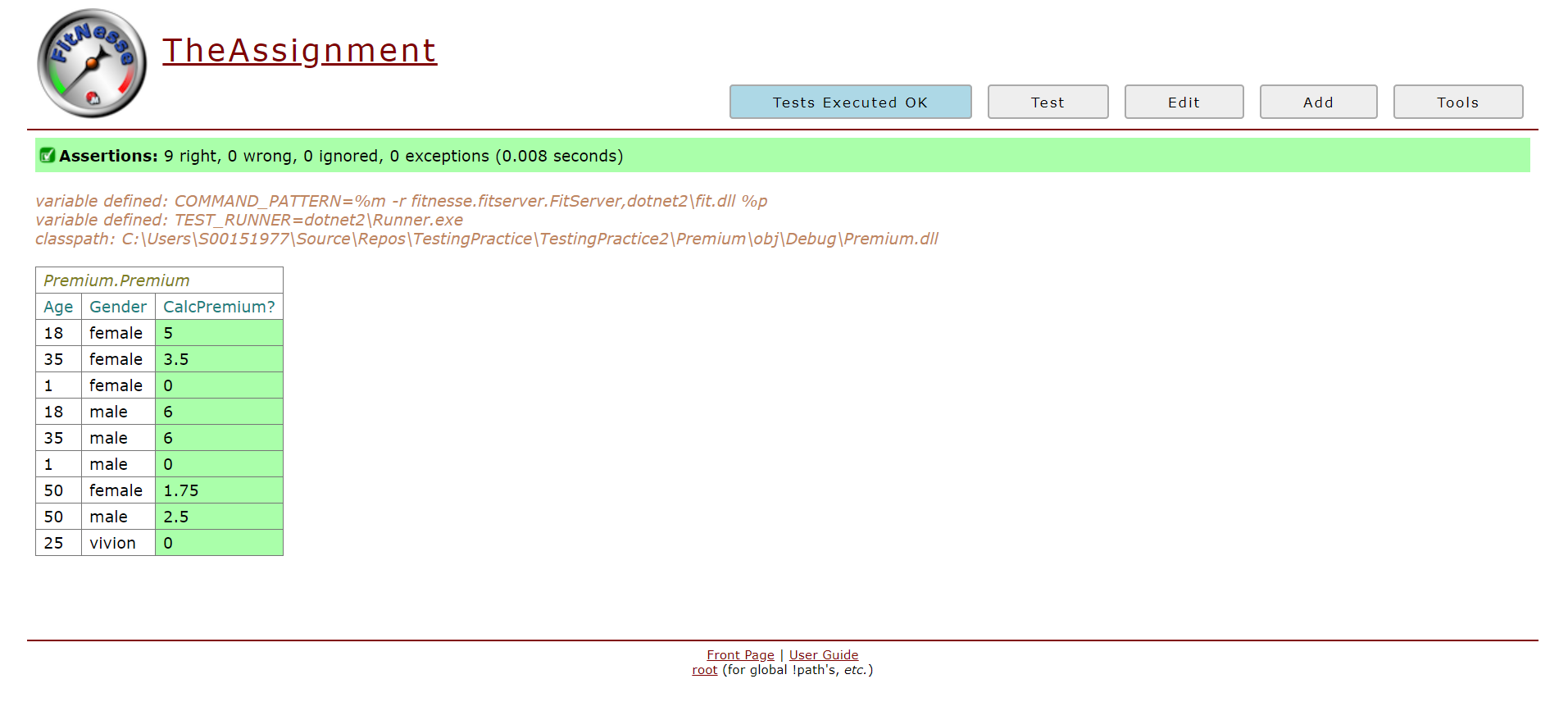
}

}

}

## 

# **Fitnesse** **automated acceptance testing server**



# **Selenium IDE and webdriver system tests**

## Katalon Class from test

using System;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading;

using NUnit.Framework;

using OpenQA.Selenium;

using OpenQA.Selenium.Firefox;

using OpenQA.Selenium.Support.UI;

namespace SeleniumTests

{

    [TestFixture]

    public class VivSTestCase

    {

        private IWebDriver driver;

        private StringBuilder verificationErrors;

        private string baseURL;

        private bool acceptNextAlert = true;

        [SetUp]

        public void SetupTest()

        {

            driver = new FirefoxDriver();

            baseURL = "<https://www.katalon.com/>";

            verificationErrors = new StringBuilder();

        }

        [TearDown]

        public void TeardownTest()

        {

            try

            {

                driver.Quit();

            }

            catch (Exception)

            {

                // Ignore errors if unable to close the browser

            }

            Assert.AreEqual("", verificationErrors.ToString());

        }

        [Test]

        public void TheVivSTestCaseTest()

        {

            driver.Navigate().GoToUrl("<http://localhost:9156/WebForm1.aspx>");

            driver.FindElement(By.Id("TextGender")).Click();

            driver.FindElement(By.Id("TextGender")).Clear();

            driver.FindElement(By.Id("TextGender")).SendKeys("male");

            driver.FindElement(By.Id("TextAge")).Click();

            driver.FindElement(By.Id("TextAge")).Clear();

            driver.FindElement(By.Id("TextAge")).SendKeys("36");

            driver.FindElement(By.Id("Button1")).Click();

        }

        public void TheVivSTestCaseTest1()

        {

            driver.Navigate().GoToUrl("<http://localhost:9156/WebForm1.aspx>");

            driver.FindElement(By.Id("TextGender")).Click();

            driver.FindElement(By.Id("TextGender")).Clear();

            driver.FindElement(By.Id("TextGender")).SendKeys("male");

            driver.FindElement(By.Id("TextAge")).Click();

            driver.FindElement(By.Id("TextAge")).Clear();

            driver.FindElement(By.Id("TextAge")).SendKeys("25");

            driver.FindElement(By.Id("Button1")).Click();

        }

        public void TheVivSTestCaseTest2()

        {

            driver.Navigate().GoToUrl("<http://localhost:9156/WebForm1.aspx>");

            driver.FindElement(By.Id("TextGender")).Click();

            driver.FindElement(By.Id("TextGender")).Clear();

            driver.FindElement(By.Id("TextGender")).SendKeys("male");

            driver.FindElement(By.Id("TextAge")).Click();

            driver.FindElement(By.Id("TextAge")).Clear();

            driver.FindElement(By.Id("TextAge")).SendKeys("55");

            driver.FindElement(By.Id("Button1")).Click();

        }

        public void TheVivSTestCaseTest3()

        {

            driver.Navigate().GoToUrl("<http://localhost:9156/WebForm1.aspx>");

            driver.FindElement(By.Id("TextGender")).Click();

            driver.FindElement(By.Id("TextGender")).Clear();

            driver.FindElement(By.Id("TextGender")).SendKeys("female");

            driver.FindElement(By.Id("TextAge")).Click();

            driver.FindElement(By.Id("TextAge")).Clear();

            driver.FindElement(By.Id("TextAge")).SendKeys("25");

            driver.FindElement(By.Id("Button1")).Click();

        }

        public void TheVivSTestCaseTest4()

        {

            driver.Navigate().GoToUrl("<http://localhost:9156/WebForm1.aspx>");

            driver.FindElement(By.Id("TextGender")).Click();

            driver.FindElement(By.Id("TextGender")).Clear();

            driver.FindElement(By.Id("TextGender")).SendKeys("female");

            driver.FindElement(By.Id("TextAge")).Click();

            driver.FindElement(By.Id("TextAge")).Clear();

            driver.FindElement(By.Id("TextAge")).SendKeys("32");

            driver.FindElement(By.Id("Button1")).Click();

        }

        public void TheVivSTestCaseTest5()

        {

            driver.Navigate().GoToUrl("<http://localhost:9156/WebForm1.aspx>");

            driver.FindElement(By.Id("TextGender")).Click();

            driver.FindElement(By.Id("TextGender")).Clear();

            driver.FindElement(By.Id("TextGender")).SendKeys("female");

            driver.FindElement(By.Id("TextAge")).Click();

            driver.FindElement(By.Id("TextAge")).Clear();

            driver.FindElement(By.Id("TextAge")).SendKeys("55");

            driver.FindElement(By.Id("Button1")).Click();

        }

        private bool IsElementPresent(By by)

        {

            try

            {

                driver.FindElement(by);

                return true;

            }

            catch (NoSuchElementException)

            {

                return false;

            }

        }

        private bool IsAlertPresent()

        {

            try

            {

                driver.SwitchTo().Alert();

                return true;

            }

            catch (NoAlertPresentException)

            {

                return false;

            }

        }

        private string CloseAlertAndGetItsText() {

            try {

                IAlert alert = driver.SwitchTo().Alert();

                string alertText = alert.Text;

                if (acceptNextAlert) {

                    alert.Accept();

                } else {

                    alert.Dismiss();

                }

                return alertText;

            } finally {

                acceptNextAlert = true;

            }

        }

    }

}

## Katalon Screenshots

