Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №2

Black box testing

№ залікової книжки = 9301

9301 mod 3 = 1

Бібліотека: Binary Flag

Сирцевий код тестів

using System;

using Xunit;

using IIG.BinaryFlag;

namespace BinaryFlagTests

{

public class Tests

{

[Fact]

public void GetFlagIfTrue()

{

MultipleBinaryFlag myBinaryFlag = new MultipleBinaryFlag(3);

bool? actual = myBinaryFlag.GetFlag();

bool expected = true;

Assert.Equal(expected, actual);

}

[Fact]

public void GetFlagIfFalse()

{

MultipleBinaryFlag myBinaryFlag = new MultipleBinaryFlag(3);

myBinaryFlag.ResetFlag(2);

bool? actual = myBinaryFlag.GetFlag();

bool expected = false;

Assert.Equal(expected, actual);

}

[Fact]

public void ResetFlag()

{

MultipleBinaryFlag myBinaryFlag = new MultipleBinaryFlag(3);

myBinaryFlag.ResetFlag(2);

myBinaryFlag.SetFlag(2);

bool? actual = myBinaryFlag.GetFlag();

bool expected = true;

Assert.Equal(expected, actual);

}

[Fact]

public void DisposeNull()

{

MultipleBinaryFlag myBinaryFlag = new MultipleBinaryFlag(3);

bool? expected = null;

myBinaryFlag.Dispose();

bool? actual = myBinaryFlag.GetFlag();

Assert.Equal(expected, actual);

}

[Fact]

public void MaxValue()

{

MultipleBinaryFlag actual = new MultipleBinaryFlag(17179868704);

MultipleBinaryFlag expected = null;

Assert.NotEqual(expected, actual);

}

[Fact]

public void MaxValueDec()

{

MultipleBinaryFlag actual = new MultipleBinaryFlag(17179868704 - 1);

MultipleBinaryFlag expected = null;

Assert.NotEqual(expected, actual);

}

[Fact]

public void MaxValueInc()

{

Assert.Throws<ArgumentOutOfRangeException>(() => new MultipleBinaryFlag(17179868704 + 1));

}

[Fact]

public void MinValue()

{

MultipleBinaryFlag actual = new MultipleBinaryFlag(2);

MultipleBinaryFlag expected = null;

Assert.NotEqual(expected, actual);

}

[Fact]

public void MinValueInc()

{

MultipleBinaryFlag actual = new MultipleBinaryFlag(2 + 1);

MultipleBinaryFlag expected = null;

Assert.NotEqual(expected, actual);

}

[Fact]

public void MinValueDec()

{

Assert.Throws<ArgumentOutOfRangeException>(() => new MultipleBinaryFlag(2 - 1));

}

[Fact]

public void ResetTest()

{

MultipleBinaryFlag myBinaryFlag = new MultipleBinaryFlag(3);

myBinaryFlag.ResetFlag(1);

string expected = "TFT";

string actual = myBinaryFlag.ToString();

Assert.Equal(expected, actual);

}

[Fact]

public void SetTest()

{

MultipleBinaryFlag myBinaryFlag = new MultipleBinaryFlag(3, false);

myBinaryFlag.SetFlag(1);

string expected = "FTF";

string actual = myBinaryFlag.ToString();

Assert.Equal(expected, actual);

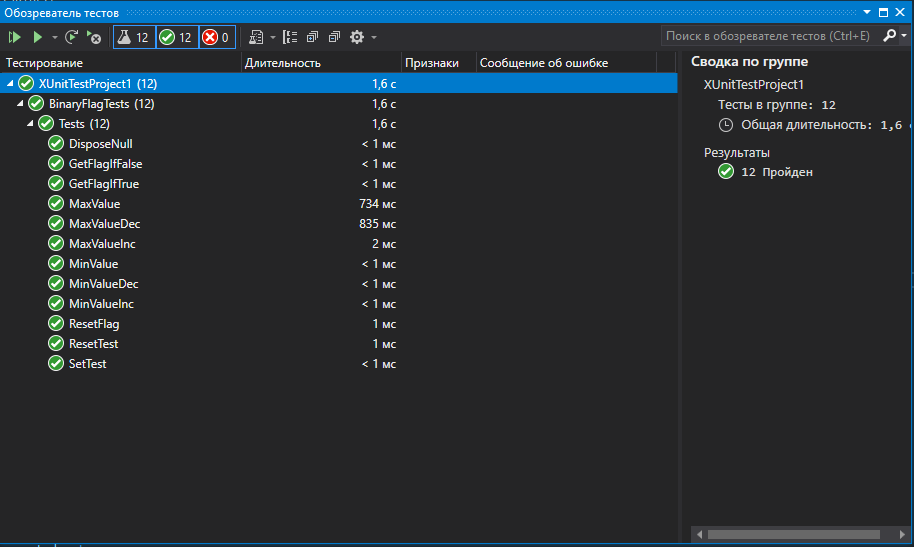
}

}

}

В лабораторній роботі були протестовані методи класу MultipleBinaryFlag. Для тестування конструктора було використано граничне тестування, тобто брались значення мінімальне – 1, мінімальне, мінімальне + 1, максимальне – 1, максимальне, максимальне + 1

Результат тестів:



[Посилання на гітхаб](https://github.com/tr1ckste/testing/blob/master/lab2/BBT/XUnitTestProject1/XUnitTestProject1/my_test.cs)