**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 №3**

з дисципліни

«Компоненти програмної інженерії. Якість та тестування програмного забезпечення»

на тему

«Unit тестування з використанням методів White Box Testing»

Виконав:

Перевірив:

студент групи ІП-93

Бабарикін Ігор Владиславович

Домінський Валентин Олексійович

номер залікової книжки: 9311

номер у списку: 9

Київ 2021

Зміст

[Мета: 3](#_Toc70775625)

[Завдання: 3](#_Toc70775626)

[Хід роботи: 3](#_Toc70775627)

[Початок роботи: 3](#_Toc70775628)

[Тестування: 4](#_Toc70775629)

[ReadLines 4](#_Toc70775630)

[TryCopy 4](#_Toc70775631)

[Результати тестування 4](#_Toc70775632)

[Сирцеві коди: 5](#_Toc70775633)

[TestFileWorkingUtils (тести) 5](#_Toc70775634)

[Висновки: 5](#_Toc70775635)

[Джерела: 5](#_Toc70775636)

Мета:

Написати Unit тести з використанням методів White Box Testing

Завдання:

|  |  |  |
| --- | --- | --- |
| N п/п | 9311 mod 6 | Library |
| 1 | 0, 2, 3 | IIG.BinaryFlag |
| 2 | 1, 4, 5 | IIG.PasswordHashingUtils |

Варіант = 9311 mod 6 = 5, отже провести тестування IIG.PasswordHashingUtils

Отже, мій вибір для лабораторної:

* .NET 5
* Бібліотека для тестування xUnit
* Бібліотека IIG.PasswordHashingUtils

Хід роботи:

Початок роботи:

Я створив проект “Lab3” на .NET 5, додав xUnit Test Project “TestPasswordHashingUtils” та бібліотеку IIG.PasswordHashingUtils:



Оскільки у даній лабораторній Ми використовуємо BBT, то можемо переглянути її вміст та зробити відповідні графи для подальшого використовування їх при написанні тестів.

Давайте пройдемося по коду та спробуємо його проаналізувати:



Як видно з даного шматку, Ми маємо два приватних статичних поля з певними значеннями за замовченням:

* \_modAdler представляє з себе модуль найбільшого простого числа, меншого, ніж 216 , який буде використовуватися при хешуванні паролю
* \_salt використовується для створення певного хешу Нашого паролю



Дана функція перезаписує значення приватних полів, які були вище, з певними перевірками на нові, які приходять як параметри



Цей метод повертає певний хеш для паролю, який Ми передали. Сюди ж можна вписати salt та adlerMod, якщо Нам потрібні якісь конкретні значення.

Далі вже йдуть функції, які є приватними, отже тестувати їх напряму - неможливо



У цьому шматку коду приймається певна стрічка, яка приймає участь у створенні SHA2 хешу.



І останній метод Adler32CheckSum. Як видно з XML-коментаря дана функція приймає текст, індекс та довжину Adler32CheckSum, а повертає Adler32CheckSum у вигляді стрічки.

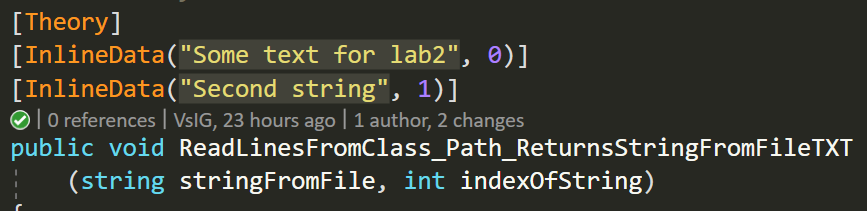
Отже Ми можемо дійти до висновку, що напряму тестувати можна лише дві функції: Init та GetHash

**Тестування**:

Тут хочу розказати деякі речі з якими я стикнувся під час тестування:

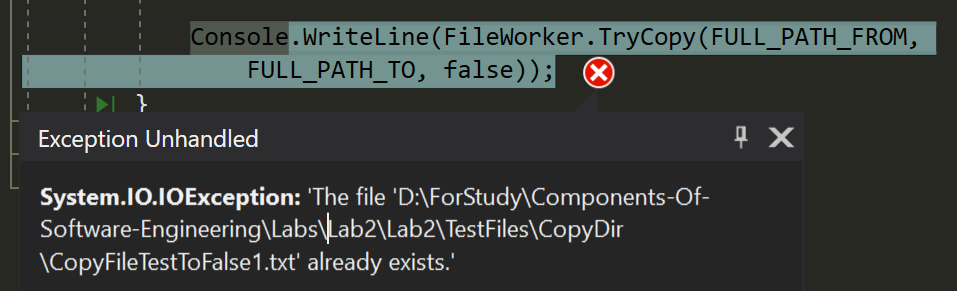
ReadLines

Даний метод повертає масив стрічок, які читає за рядками. [Офіційна документація](https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-best-practices#characteristics-of-a-good-unit-test) каже, що в тестах потрібно використовувати якомога менше логіки, саме тому я використав InlineData для вхідних параметрів та замінив [Fact] на [Theory]:



TryCopy

Я стикнувся з проблемою, коли при копіюванні файлу до директорії без перезапису, де він уже є, виникає помилка, яку вдалося виправити лише видаленням файлу після проходження тесту.



Результати тестування



Сирцеві коди:

TestFileWorkingUtils (тести)

**using** IIG**.**Core**.**FileWorkingUtils**;**

Висновки:

Виконавши цю лабораторну роботу я познайомився з White Box тестуванням у цілому, а також використав таку техніку, як Тестування потоку виконання

Джерела:

* Github - <https://github.com/VsIG-official/Components-Of-Software-Engineering>
* Program.cs - <https://github.com/VsIG-official/Components-Of-Software-Engineering/blob/master/Labs/Lab2/Lab2/Program.cs>
* TestFileWorkingUtils.cs - <https://github.com/VsIG-official/Components-Of-Software-Engineering/blob/master/Labs/Lab2/TestFileWorkingUtils/TestFileWorkingUtils.cs>
* Офіційна документація - [https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-best-practices#characteristics-of-a-good-unit-test](https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-best-practices%23characteristics-of-a-good-unit-test)
* Лекція по темі Black Box Testing - [https://docs.google.com/presentation/d/1zDBgNUV73ja\_yShIY-\_WpTGkktol7DOE/edit#slide=id.p18](https://docs.google.com/presentation/d/1zDBgNUV73ja_yShIY-_WpTGkktol7DOE/edit%23slide=id.p18%20)