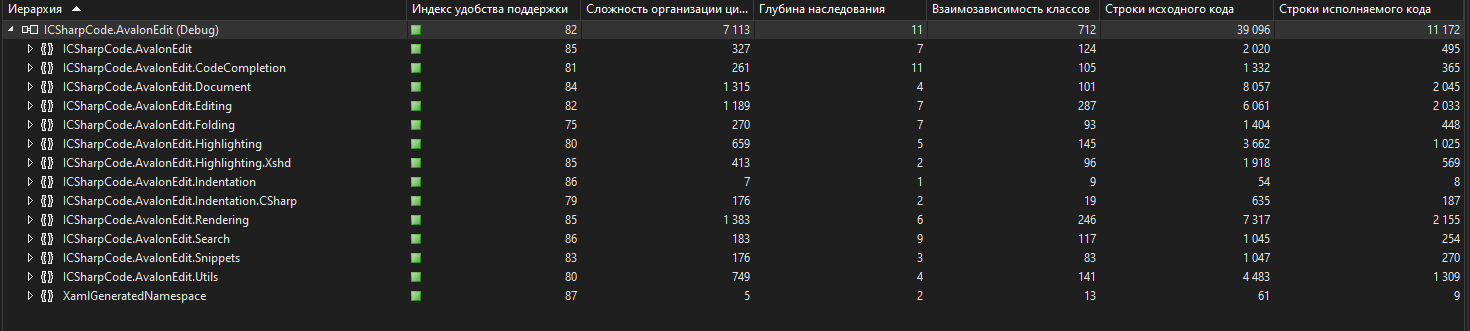
|  |  |  |  |
| --- | --- | --- | --- |
| МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка  ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  **Кафедра програмних систем і технологій**        Дисципліна  **«Емпіричні методи програмної інженерії»**      **Практична робота № 2**  **“ЗБІР ДАНИХ, ВИМІРЮВАННЯ ТА ДОСЛІДЖЕННЯ МЕТРИК В ЕМПІРИЧНІЙ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ”** | | | |
| **Виконав:** | Гоша Д. О | **Перевірила**: | Юрчук Ірина Аркадіївна |
| Група | ІПЗ-23 | Дата перевірки |  |
| Форма навчання | денна | Оцінка |  |
| Спеціальність | 121 |
| 2022 | | | |

Мета: дослідити застосування, класифікацію та призначення метрик програмного забезпечення, а також вимірювати програмне забезпечення та використовувати метрики в розробці.

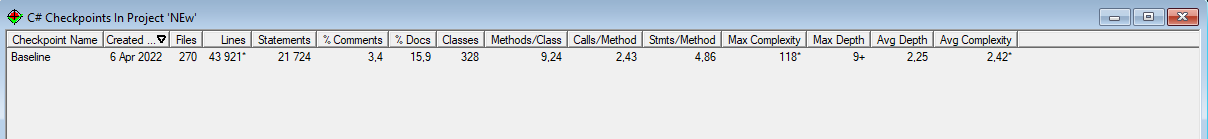
# Варіант 4

# Виконання лабораторної роботи

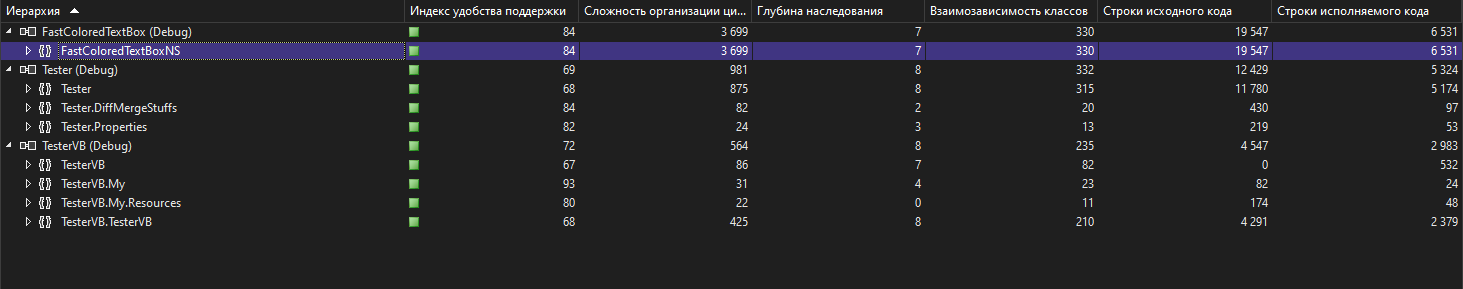
1. Порівняння засобів збору (вказані у варіанті, див. додаток 1, таблиця на не менше ніж 2 проектах по одному з кожної групи.
   1. Проект AvalonEdit (https://github.com/icsharpcode/AvalonEdit)
      1. VS studio



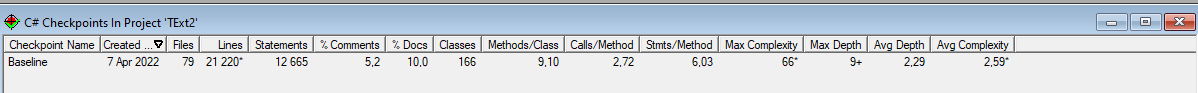
* + 1. SourceMonitor



* 1. Проект FastColoredTextBox (<https://github.com/PavelTorgashov/FastColoredTextBox>)
     1. VS studio



* + 1. SourceMonitor



1. За допомогою метрик програмного забезпечення визначається якість даного ПЗ. Наприклад якість коду може визначатись різними критеріями. Загалом метрики коду впливають на такі факотори:
   1. Прочитність коду.
   2. Легкість підтримки, тестування, відлагодження, виправляння помилок, рефакторингу та портування.
   3. Низька складність коду.
   4. Коректність обробки винятків.  
      В свою чергу ці фактори впливають на подальшу підтримку ПЗ, час впровадження нового компоненту в ПЗ, його коректну роботу, та час, який знадобиться новому співробітнику на розуміння коду ПЗ.
2. Основні групи метрик Метрики можуть бути розділені на такі категорії:
   1. Метрики розміру – міри сутності що аналізується.
   2. Метрики складності – міри складності сутності що аналізується.
   3. Метрики зв’язаності – міри зв’язаності даних між сутностями, що аналізується.
   4. Метрики зчеплення – міри зчеплення класів. Прямі метрики - NOM, NOC, NOP, NDD, CALL. Непрямі метрики - WMC, TCC, PNAS.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **NOM** | **NOC** | **NOP** | **NDD** | **CALL** | **WMC** | **TCC** | **PNAS?** |
| **I-1** | 3.2 | 199 | 1.62 | 3 | 1.75 | 3 | 2,16 | 2 |
| **I-2** | 9.2 | 43921 | 4.86 | 9 | 2.43 | 118 | 3,24 | 9 |
| **I-3** | 5.7 | 20098 | 3,84 | 7 | 2.15 | 96 | 3,40 | 9 |
| **I-4** | 9.1 | 21220 | 6.03 | 9 | 2.72 | 66 | 2,13 | 5 |
| **I-5** | 8.3 | 611 | 2,46 | 7 | 2.32 | 68 | 4,17 | 3 |
| **I-6** | 7.0 | 5541 | 7,88 | 4 | 2.48 | 86 | 1,49 | 4 |
| **I-7** | 4.2 | 5043 | 2.95 | 7 | 1.74 | 15 | 2,33 | 8 |
| **I-8** | 3.6 | 7064 | 10,24 | 6 | 2.94 | 13 | 2,63 | 7 |
| **I-9** | 7.1 | 1408 | 2,5 | 9 | 2.09 | 53 | 1,6 | 3 |
| **I-10** | 6.9 | 67923 | 10,5 | 2 | 2.33 | 114 | 1,45 | 4 |
| **II-1** | 4.3 | 62274 | 6,07 | 5 | 2.12 | 15 | 1,93 | 6 |
| **II-2** | 6.8 | 2475 | 9,05 | 9 | 2.89 | 141 | 1,5 | 3 |
| **II-3** | 9.2 | 983 | 19 | 8 | 2.56 | 112 | 2,3 | 4 |
| **II-4** | 5.9 | 3160 | 4,69 | 1 | 1.73 | 116 | 2,45 | 9 |
| **II-5** | 5.2 | 34563 | 7 | 6 | 2.38 | 77 | 2,21 | 7 |
| **II-6** | 8.0 | 67345 | 4,54 | 8 | 2.21 | 43 | 1,7 | 8 |
| **II-7** | 7.8 | 23453 | 4,42 | 7 | 1.94 | 89 | 3,1 | 7 |
| **II-8** | 2.9 | 341 | 6,51 | 1 | 1.51 | 118 | 2,14 | 7 |
| **II-9** | 3.6 | 67700 | 4,35 | 5 | 1.98 | 43 | 2,28 | 9 |
| **II-10** | 2.6 | 234 | 5,12 | 7 | 2.05 | 51 | 1,95 | 7 |

# Аналіз результів:

Бачимо деякі закономірності у прямих мтериках. Таких як розмір коду впливає на максимальне заглибленя(Кількість потомків). Але не завжди. Є деякі виключення , що свідчать про іншу архітектуру ПЗ , як не передбачає наслідування як концепцію ООП.

Здебільшого деякі похибки знаходяться у проектах, що написані на декількох мовах програмування. Це впливає на результат аналізу , тому що логічна частина не аналізується. Це можна спостерігати у проекті під номером один.

Глобальний висновок свідчить, що чим більше значення у метриці виклику методів (CALL) , при великому обсязі коду , та відносно не велике значення метрик наслідування(NDD) та числа пакeтів(NOP) , тип якісніше програмний застосунок.

# Висновки

В цій лабораторній роботі було розглянуто 20 проектів. Кожен з проектів був проаналізован програмним застосунком SourceMonitor, результати роботи внесені у таблицю та порівняні між собою. У першій частині завдання порівняно роботу різних програм – анілізаторів коду , а саме - Visual Studio , SourceMonitor. Прикрілені скриншоти виконання і підведені підсимки роботи застосунків.