МІНІСТЕРСТВО ОСВІТИ І НАУКИ

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп’ютерної та програмної інженерії

Кафедра інженерії програмного забезпечення



Лабораторна робота №1.6

з дисципліни «Якість програмного забезпечення та тестування»

на тему:

**«Колективне інспектування програмного забезпечення»**

Виконав: студент групи ПІ – 322  
Царук С.О.  
Перевірила: Корнієнко С.П

Київ – 2021

**Тема:** Колективне інспектування програмного забезпечення

**Мета:** Перевірити вихідний код справжнього програмного забезпечення та створити звіт по виконаній роботі

**Опис завдання:**

1) Завантажити та встановити RSS Bandit та його вихідний код.

2) Відкрити вихідний код у Visual Studio 2008/2012/… (RSS Bandit.2008.sln).

3) Оглянути наступні .cs - файли згідно до основних принципів написання коду на С# (документи у додатку):

a. RssBandit/Core/CommandLineParser.cs

b. RssBandit/Core/GuiInvoker.cs

c. RssBandit/Core/RssBanditApplication.Commands.cs

d. RssBandit/Core/RssBanditApplication.cs

e. RssBandit/Core/RssBanditApplication.EventHandlers.cs

f. RssBandit/Core/ RssBanditUpdateManager.cs

4) Додаткові завдання:

a. Знайдіть повторюваний код, запропонуйте методи рефакторінгу;

b. Спробуйте знайти місця у коді, які потребують впровадження паттернів;

**Хід роботи**

1. Ролі в команді:

* модератор (Косих)
* інспектори (Павлов , Царук)

**Інспектор Павлов**

**RssBandit/Core/CommandLineParser.cs**

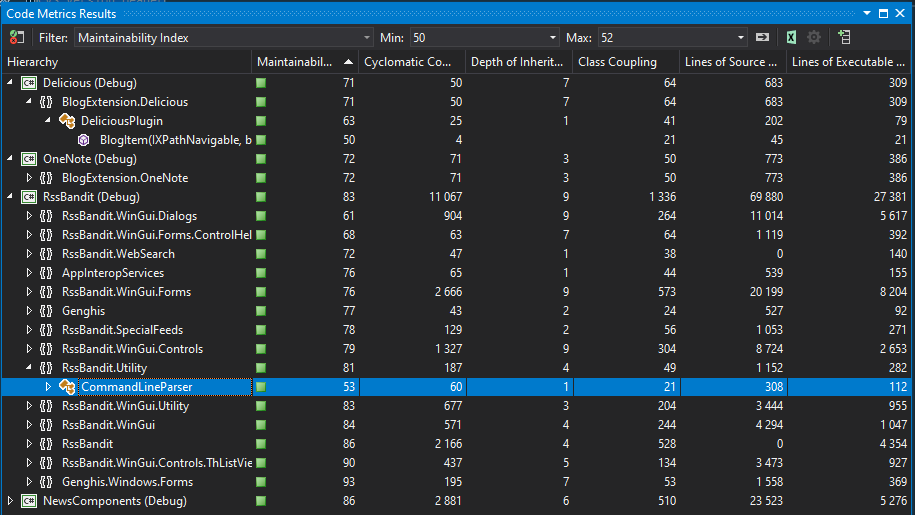
Індекс зручності підтримки – 53

Цикломатична складність - 60

Глибина наслідування – 1

Зв’язність класів – 21

Lines of Code (кількість стрічок коду) – 308



**RssBandit/Core/GuiInvoker.cs**

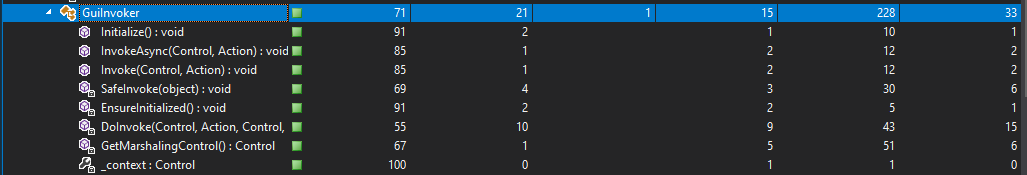
Індекс зручності підтримки – 71

Цикломатична складність - 21

Глибина наслідування – 1

Зв’язність класів – 15

Lines of Code (кількість стрічок коду) – 33



**Інспектор Царук**

**Файл RssBanditApplication.cs**

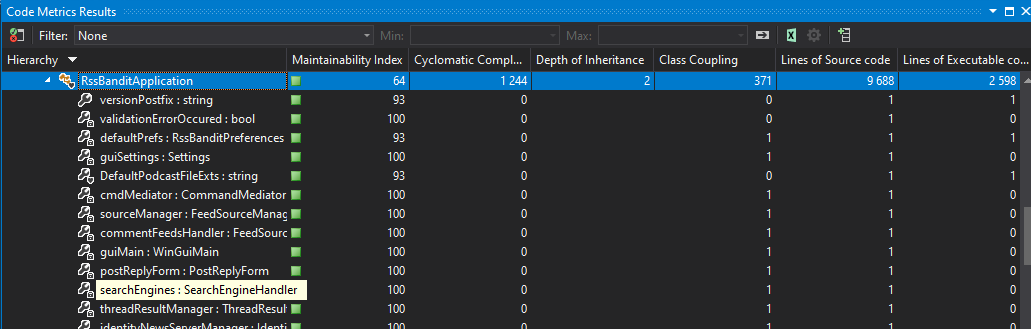
Індекс зручності підтримки – 64

Цикломатична складність - 1244

Глибина наслідування – 2

Зв’язність класів – 371

Lines of Code (кількість стрічок коду) – 2598



**RssBanditApplication.CommandLineOptions.cs:**

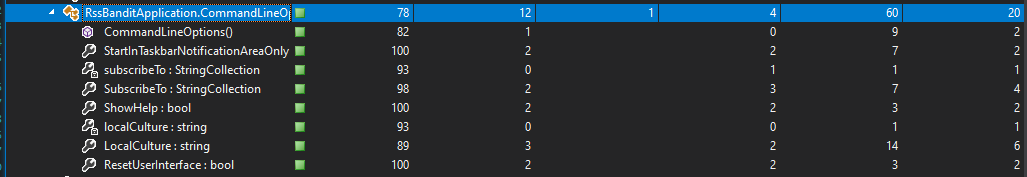
Індекс зручності підтримки – 78

Цикломатична складність - 12

Глибина наслідування – 1

Зв’язність класів – 4

Lines of Code (кількість стрічок коду) - 20



**Модератор Косих**

**RssBandit/Core/RssBanditApplication.EventArgs.cs**

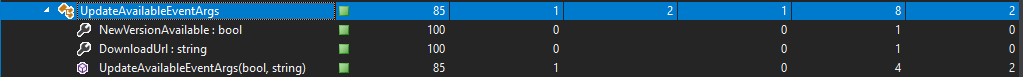
Індекс зручності підтримки – 85

Цикломатична складність - 11

Глибина наслідування – 2

Зв’язність класів – 1

Lines of Code (кількість стрічок коду) - 2



**RssBandit/Core/RssBanditUpdateManager.cs**

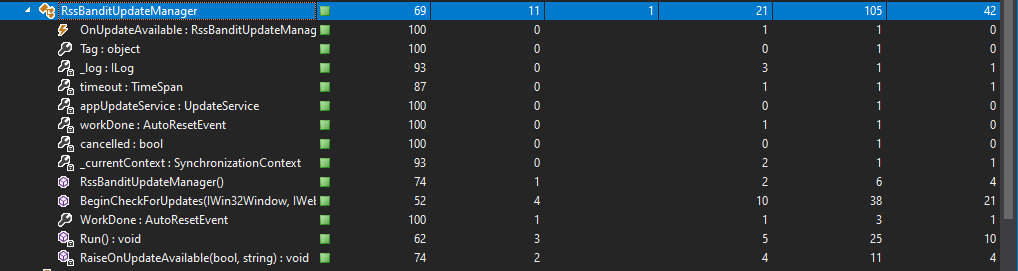
Індекс зручності підтримки – 69

Цикломатична складність - 11

Глибина наслідування – 1

Зв’язність класів – 21

Lines of Code (кількість стрічок коду) - 42



* **Iндекс експлуатаційної надійності** (Maintainability Index, MI) - комплексний показник якості коду (від 0 до 100 - чим вище, тим краще); методика його визначення розроблена фахівцями Carnegie Mellon Software Engineering Institute;
* **Циклічна складність** (Cyclomatic Complexity, CC) - показник, що характеризує число гілок в програмному коді та обчислюється шляхом підрахунку операторів циклу, умовного переходу;
* **Глибина успадкування** (Depth of Inheritance) - характеризує довжину ланцюжків спадкування в програмному коді;
* **Зчеплення класів** (Class Coupling) – відображає ступінь залежності класів між собою (в тому числі наявність спільних даних, об'єктів тощо);
* **Число рядків коду**: чим більше рядків, тим складніше програма.

**Висновок:** в ході лабораторної роботи було перевірено файли згідно завдання до програми RSS Bandit. Під час інспекції дефектів не виявлено, програма компілюється стабільно. Ми перевірили такі файли: a. RssBandit/Core/CommandLineParser.cs

b. RssBandit/Core/GuiInvoker.cs

c. RssBandit/Core/RssBanditApplication.Commands.cs

d. RssBandit/Core/RssBanditApplication.cs

e. RssBandit/Core/RssBanditApplication.EventHandlers.cs

f. RssBandit/Core/ RssBanditUpdateManager.cs

Під час інспектування даних файлів було виявлена досить велика цикломатична складність в кожному файлів. Щоб зменшити індекс цикломатичної складності ми пропонуємо цикли умовними виразами. Також було виявлено достатньо великий індекс зв’язності класів. Для зменшення індексу зв’язності класів, ми пропонуємо використовувати основні принципи ООП, а саме, поліморфізм.

За цим принципом потрібно виділити загальну функціональність класів в абстрактний клас або інтерфейс. Після цього індекс зв’язності класів повинен зменшитись