**НПУ імені М.П.Драгоманова**

**Факультет інформатики**

***Кафедра теоретичних основ інформатики***

**ЛАБОРАТОРНА РОБОТА №1**

## ТЕМА: Метрики програмного коду

## з курсу

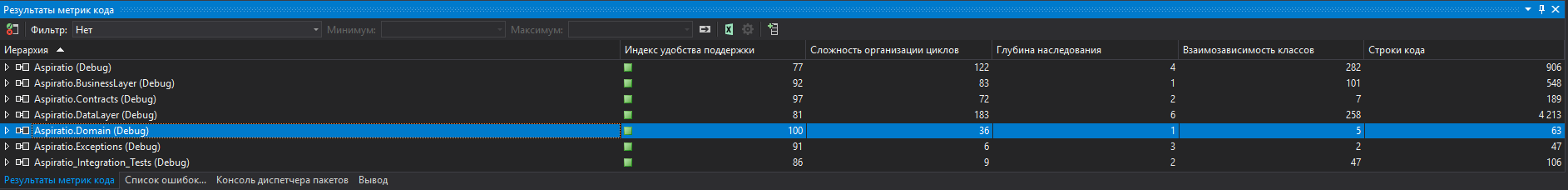
***«Емпіричні методи програмної інженерії»***

Студент Трембіцький Н.В.

Група 41-ІПЗ

Факультет Інформатики

Викладач: професор Біляй Ю.П.



1. Maintainability Index (Индекс удобства поддержки) - это комплексный показатель качества кода. Эта метрика рассчитывается по следующей формуле:

MI = MAX(0, (171 — 5.2 \* ln(HV) — 0.23 \* CC — 16.2 \* ln(LoC)) \* 100 / 171)

* HV – Halstead Volume, вычислительная сложность. Чем больше операторов, тем больше значение этой метрики;
* CC – Cyclomatic Complexity;
* LoC – количество строк кода.

2. Cyclomatic Complexity (Сложность организации циклов) – показывает структурную сложность кода. Иными словами, количество различных ветвей кода. Считается на основе операторов в Вашем коде, строя графы переходов от одного оператора к другому. К примеру, оператор if-else увеличит эту метрику, потому что здесь будут разные ветви выполнения.

3. Depth of Inheritance (Глубина наследования) - для каждого класса эта метрика показывает, насколько глубоко он в цепочке наследования.

4. Class Coupling (Взаимозависимость классов) – указывает на зависимость классов друг от друга. Проект с множеством зависимостей очень трудно и дорого поддерживать.

5. Lines of Code (Строки кода) - напрямую используется редко. В наши дни, с множеством разнообразных как подходов к программированию, так и языков, эта метрика дает нам мало полезной информации. Если брать во внимание отдельный метод, то можно разбить его на несколько методов поменьше.

Изначально стоит обращать внимание на Maintainability Index. Старайтесь придерживать его около 70-90. Это значительно облегчит сопровождения кода как Вами, так и другими программистами. Иногда стоит оставить его на уровне 50-60, так как переписать некоторые участки кода бывает очень затратным. Оценивайте здраво как код, так и Ваши возможности с затратами.

Стоит также уделить много внимания Class Coupling. Эта метрика должна быть как можно меньшей. Ведь она так же способствует поддержке кода. Для оптимизации возможно придется пересматривать дизайн проекта и некоторые архитектурные решения.

Теперь стоит уделить внимание Cyclomatic Complexity. Эта метрика показывает сложность кода, а это так же влияет на поддержку кода в будущем. Иногда приходится переписывать куски кода, которые писали до Вас другие люди, так как Вы просто не можете понять, что, как и зачем в этом методе. Конечно, этому еще способствует стиль кода и идея, но не забывайте о Cyclomatic Complexity при рефакторинге.