Задание: Провести измерение известных метрик кода на тестовом примере.

Метрики кода, вычисляемые Visual Studio:

- **Индекс удобства обслуживания** — вычисляет значение индекса от 0 до 100, которое представляет относительную простоту обслуживания кода. Высокое значение означает лучшую поддержку. Цветовые оценки можно использовать для быстрого выявления проблемных мест в коде. Зеленая оценка составляет от 20 до 100 и указывает, что код имеет хорошую поддержку. Желтая оценка составляет от 10 до 19 лет и указывает, что код умеренно поддерживается. Красная оценка — это оценка от 0 до 9 и указывает на низкую поддержку.
- **Цикломатическая сложность** — измеряет структурную сложность кода. Он создается путем вычисления количества различных путей кода в потоке программы. Программа, которая имеет сложный поток управления, требует больше тестов для достижения хорошего объема протестированного кода и менее поддерживается.
- **Глубина наследования** — указывает количество различных классов, наследуемых друг от друга, вплоть до базового класса. Глубина наследования аналогична взаимозависимости классов, что изменение базового класса может повлиять на любой из унаследованных классов. Чем выше это число, тем глубже наследование и чем выше потенциал для изменений базового класса, что приведет к критическому изменению. Для глубины наследования низкое значение хорошо, и высокое значение плохо.
- **Объединение классов** — измеряет связь с уникальными классами с помощью параметров, локальных переменных, возвращаемых типов, вызовов методов, универсальных или шаблонных экземпляров, базовых классов, реализаций интерфейсов, полей, определенных во внешних типах и оформлении атрибутов. Хорошая конструкция программного обеспечения определяет, что типы и методы должны иметь высокую сплоченность и низкую взаимозависимость. Высокая взаимозависимость указывает на структуру, которую трудно использовать и поддерживать из-за многих взаимозависимостей на других типах.
- **Строки исходного кода** — указывает точное количество строк исходного кода, присутствующих в исходном файле, включая пустые строки. Эта метрика доступна начиная с Visual Studio 2019 версии 16.4 и Microsoft.CodeAnalysis.Metrics (2.9.5).
- **Строки исполняемого кода** — указывает приблизительное количество строк или операций исполняемого кода. Это число операций в исполняемом коде. Эта метрика доступна начиная с Visual Studio 2019 версии 16.4 и Microsoft.CodeAnalysis.Metrics (2.9.5). Значение обычно является близким совпадением с предыдущей метрикой, **строками кода**, которая является метрикой на основе инструкций MSIL, используемой в устаревшем режиме.

Отображение результатов метрик кода

Окно **"Результаты метрик кода**" отображается автоматически при создании результатов метрик кода. Окно также можно отобразить в любое время.

Окно результатов метрик кода можно отобразить с помощью одной из следующих последовательностей меню:

- В меню **"Анализ"** выберите**"Результаты метрики кодаWindows**>".
- В меню **"Вид**" выберите **"Другиеметрики кода** Windows>".

Откроется окно **"Результаты метрик кода** ", даже если оно не содержит результатов.

Просмотр сведений о метриках кода

Если были созданы результаты метрик кода, разверните дерево в столбце **Hierarchy**.

Фильтрация результатов метрик кода

Вы можете отфильтровать результаты, отображаемые в окне **"Результаты метрик кода**", с помощью панели инструментов в верхней части экрана. Например, может потребоваться просмотреть только результаты с индексом удобства обслуживания ниже 65.

Раскрывающийся список **"Фильтр**" содержит имена столбцов результатов. При определении фильтра он добавляется в нижнюю часть списка вместе с отступом. Список может содержать последние 10 фильтров, которые были определены.

Фильтрация результатов метрик кода

1. В списке **фильтров** выберите имя столбца.
2. В **поле Min** введите минимальное значение, которое будет отображаться.
3. В поле **Max** введите максимальное значение, которое будет отображаться.
4. Нажмите кнопку **"Применить фильтр**".
5. Чтобы просмотреть сведения о результатах, разверните дерево иерархии.

Добавление, удаление и изменение порядка столбцов данных

Столбцы результатов можно добавлять или удалять из окна **"Результаты метрик кода**". Кроме того, можно изменить порядок столбцов результатов, чтобы они отображались в нужном порядке.

Добавление или удаление столбца

1. Нажмите кнопку **"Добавить или удалить столбцы**" или щелкните правой кнопкой мыши любой заголовок столбца и выберите команду **"Добавить или удалить столбцы**".
2. В диалоговом окне **"Добавление и удаление столбцов**" установите или снимите флажок для столбца, который требуется добавить или удалить, и нажмите кнопку **"ОК**".

Изменение порядка столбцов

1. Нажмите кнопку **"Добавить и удалить столбцы**".

2. В диалоговом окне **"Добавление и удаление столбцов**" выберите столбец, который требуется переместить, а затем щелкните стрелку вверх или стрелку вниз.

3. Если столбец расположен в нужном месте, нажмите **кнопку "ОК**".

Копирование данных в буфер обмена или Excel

Вы можете выбрать и скопировать выбранную строку данных метрик кода в буфер обмена в виде текстовой строки, содержащей одну строку для имени и значения каждого столбца данных. Вы также можете щелкнуть **"Открыть выбор" в Microsoft Excel**, чтобы экспортировать все результаты метрик кода в электронную таблицу Excel.

Создание рабочего элемента на основе результата

1. Щелкните результат правой кнопкой мыши.

2. Наведите указатель мыши на **создание рабочего элемента** и выберите тип рабочего элемента, который нужно создать (**ошибка**, **задача** и т. д.).

3. Заполните форму рабочего элемента, заполнив все обязательные поля.

4. В меню **"Файл"** нажмите кнопку **"Сохранить все**", чтобы сохранить рабочий элемент.

Создание ошибки на основе результата

1. Щелкните результат, чтобы выбрать его.

2. Нажмите кнопку **"Создать рабочий элемент**".

3. Заполните форму рабочего элемента, заполнив все обязательные поля.

4. В меню **"Файл"** нажмите кнопку **"Сохранить все**", чтобы сохранить рабочий элемент.

Оценка значений метрик.

Если сопоставить внутренней структуре программы набор метрик, по значению которых имеется возможность судить о ее качестве, можно ввести некоторую функцию суммарной корректности внутренней структуры программного обеспечения. Значение величины суммарной корректности показывает близость качества внутренней структуры программного обеспечения к ожидаемому. Аргументами функции являются множества текущих значений всех метрик - метрический снимок программы. Увеличение значения функции в результате какого-либо изменения исходного кода может говорить об улучшении внутренней структуры программного обеспечения.

Суммарная корректность определяется как сумма некоторых функций корректности метрик каждой программной единицы. Функция корректности отражает степень близости значения метрики ее идеальному значению и может задаваться в зависимости от требований к конкретному проекту. Область значений функции: [0..1]. Значение функции равное 1 соответствует корректному значению метрики, значение 0 максимальному удалению значения метрики от корректного значения. В простейшем случае функция корректности метрики равна 1, если значение метрики укладывается в рекомендуемый диапазон, и 0 — если не укладывается. В других случаях, функция может принимать значение 1, если величина метрики совпадает с каким-либо оптимальным значением и увеличиваться, стремясь к 0, по мере удаления от него и приближения к границам диапазона корректных значении. Для задания значимости степени близости конкретной метрики корректному значению необходимо введение весового коэффициента для каждой метрики. Сумма весовых коэффициентов всех метрик должна равняться 1. Варьируя весовыми коэффициентами можно задавать направление оптимизации для достижения требуемого корректного значения определенного набора метрик. В простейшем случае при наличии N метрик весовой коэффициент может быть равен 1/N для каждой метрики. Суммарную корректность можно определить как сумму произведений значений функции корректности и весового коэффициента для каждой составляющей метрики.

Обзор современных средств расчёта метрик

1.1.7Средства расчёта метрик для сред разработки

Для обзора средств расчёта метрик были выбраны две наиболее популярные среды разработки Eclipse и NetBeans. Обе среды свободно распространяемы и являются «open source» проектами.

Eclipse Metrics Plugin для среды Eclipse. Поддерживает следующие метрики:

- McCabe’s Cyclomatic Complexity – цикломатическая сложность метода;
- Efferent Couplings – исходящая связность;
- Lack of Cohesion in Methods - недостаток связности;
- Lines Of Code in Method - количество строк кода в методе;
- Number Of Fields – количество полей;
- Number Of Levels – количество вложенностей в методе;
- Number Of Parameters – количество параметров функции
- Number Of Statements – количество инструкций в методе

• Weighted Methods Per Class – взвешенные методы на класс

CC (max)	LOCm (max)	NLS (max)	NOL (max)	NOP (max)	NOS (max)	Ce	LCOM-CK	LCOM-HS %	LCOM-PFI %	LCOM-TC %	NOF	WMC	Line	Type
2	9	1	2	2	6	21	0	68	89	55	4	40	22	AbstractASTVisitorCalculator
2	5	0	2	3	3	7	0	0	0	0	1	7	7	AbstractMetricProcessor
-	-	-	-	1	-	-	-	-	-	-	-	-	5	Agent

Рис. 1.5. Отображение метрик Eclipse Metrics Plugin.

JDepend Plugin для среды Eclipse. Поддержка метрик:

• CC - Concrete Class Count – количество классов;
• AC - Abstract Class (and Interface) Count – количество абстрактных классов и интерфейсов;
• Ca - Afferent Couplings (Ca) – входящая связность;
• Ce - Efferent Couplings (Ce) – исходящая связность;
• A - Abstractness (0-1) – абстрактность;
• I - Instability (0-1) – неустойчивость;
• V - Volatility (0-1) – изменчивость

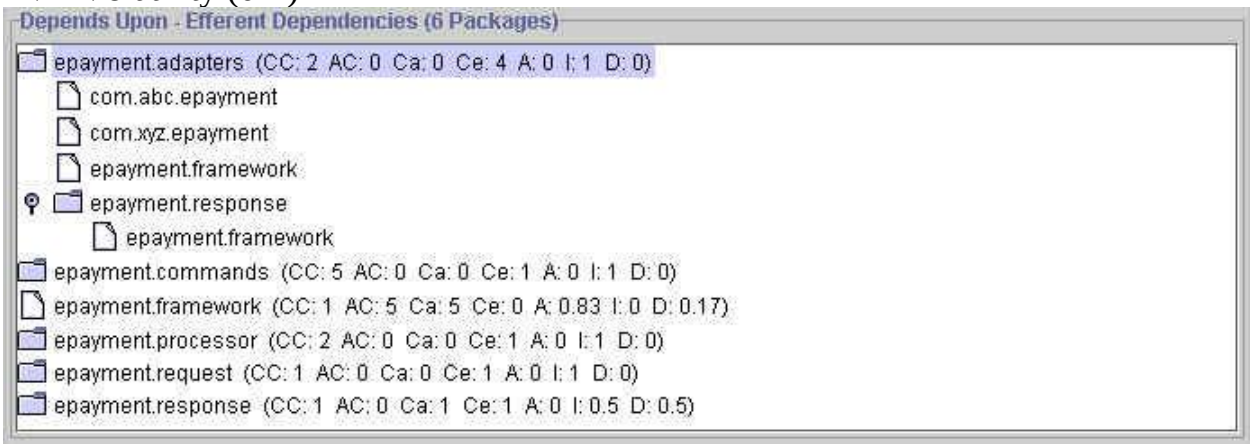

Рис. 1.6. Отображение метрик JDepend Plugin.

Обзор средств расчёта метрик для среды NetBeans выявил недостаток. Набравшая большую популярность в последнее время она не имеет собственных средств оценки качества кода. Что касается встраиваемых дополнений в виде модулей, то все они имеют больше минусов, чем плюсов. Рассмотрим некоторые из них.

Одним из первых модулей для расчёта метрик являлся плагин “NetBeans Metrics Module” (рис. 1.7), разработанный сообществом NetBeans. В его состав входило 7 метрик. Модуль имел много недостатков и его так и не включили в стандартный пакет.[8] Большим минусом было то, что модуль не имел никакого отношения к исходным кодам. Он работал напрямую с откомпилированными .class файлами вместо исходных кодов Java (.java).

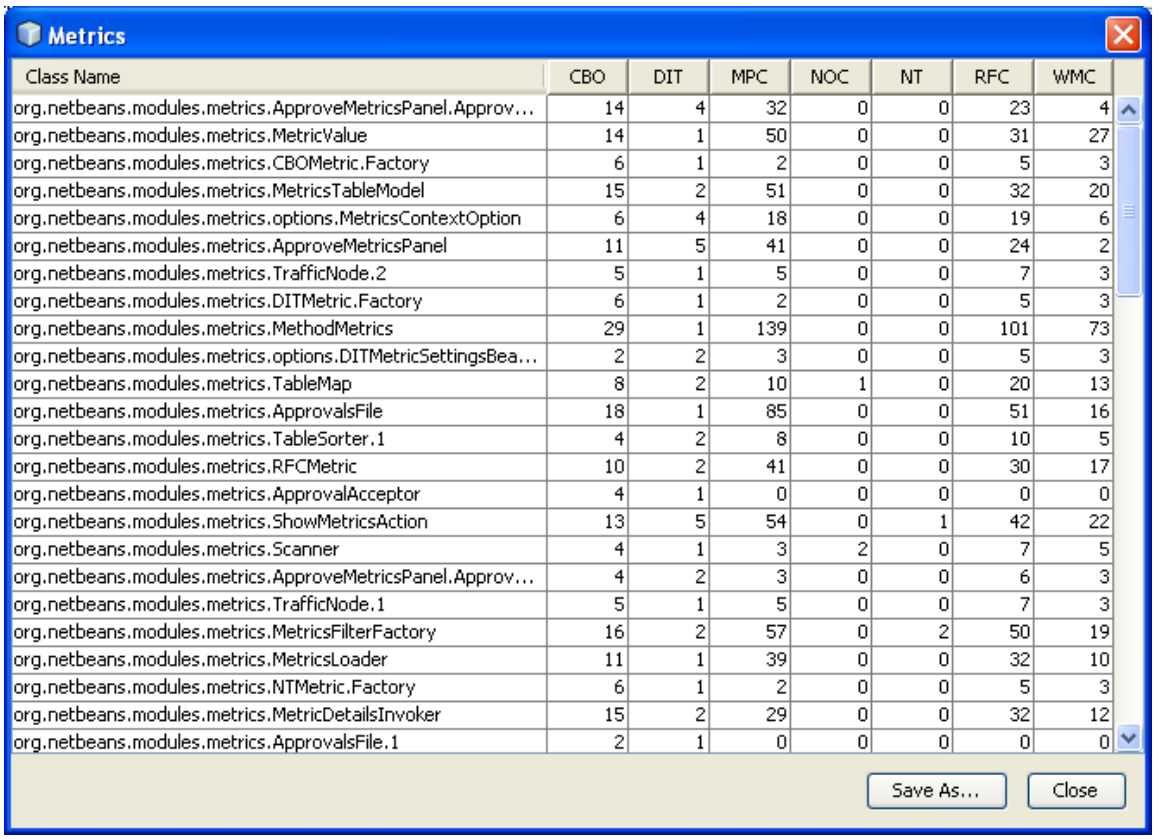

Рис. 1.7. Отображение метрик NetBeans Metrics Module

Другим средством расчёта был плагин JRefactory. Помимо своего главного назначения применять рефакторинг он также мог рассчитывать метрики. Автор программы Майк Аткинсон пытался поддерживать работу плагина параллельно в 4 средах разработки. По мере развития сред проект медленно переставал развиваться. Последний раз он был модифицирован в 2004 году. Минусом данного проекта было наличия своего собственного парсера, который тащил за собой плагин. Некоторые дефекты парсера не позволяли полностью распознать все конструкции языка Java.[9]

1.1.8Специализированные программные продукты анализа кода

Borland Together – коммерческий инструмент UML-моделирования, дополненный возможностями вычисления метрик исходного кода. Поддерживается обширное число различных метрик, значительная часть которых – объектно-ориентированные: SLOC, количественные метрики классов (число атрибутов, классов, конструкторов, операций), цикломатическая сложность, метрики сложности классов (LOCOM1, LOCOM2, LOCOM3, WMPC1, WMPC2, NORM), метрики связности, Холстеда, наследования, полиморфизма, процентные соотношения (доля комментариев, приватных, публичных и защищенных членов классов), максимальные значения (уровня вложенности, числа параметров и операций).

Verisoft Complexity Measures Tool – коммерческий продукт. Поддерживаются только языки C/C++ и Java (поставляется в виде двух различных редакций). С помощью этого продукта можно рассчитывать следующие метрики: SLOC, цикломатическую сложность, метрики Холстеда, индекс сопровождаемости . Имеет графический интерфейс (с возможностью

работы в режиме командной строки), позволяет формировать отчеты в текстовой форме или HTML.

Существуют также специализированные продукты в основном рассчитанные на набор метрик Чидамбера и Кемерера и обычно являются коммерческими.(табл. 1.2)

Таблица 1.2. Специализированные коммерческие продукты для расчета метрик

Пакет	**Производитель**	**Поддерживаемые языки**	**^ Считаемые метрики**
RSM Metrics	M Squared Technologies	C++, Java	DIT, NOC
Project Analyzer	Aivosto	Visual Basic	Весь набор
SDMetrics	SDMetrics	C++, Java	WMC, DIT, NOC, CBO, RFC
Krakatau Metrics	Power Software	C++, Java, Visual Basic	Весь набор
Software Metrics	McCabe & Associates	C++, Java	WMC, DIT, NOC, RFC, LCOM

Проанализировав данные программные пакеты можно сказать, что все они достаточно узконаправленны, это очевидно является недостатком. Метрики для конкретно поставленной задачи часто ограничивают пользователя. Приходится прибегать к сторонним поискам, находить другие пакеты. В итоге это всё часто приводит к параллельной работе пользователя сразу в нескольких средств.