

Практическое занятие №18

Тема: «Выполнение измерений характеристик кода в среде Eclipse».

Цель занятия: Приобрести практические навыки измерений характеристик кода в программе Eclipse.

Оборудование: Компьютер, установленное ПО Eclipse.

Количество часов: 2 часа.

Порядок выполнения:

Задание: Провести измерение известных метрик кода на тестовом примере.

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 равно 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 значений. 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

1.1.7 Средства 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

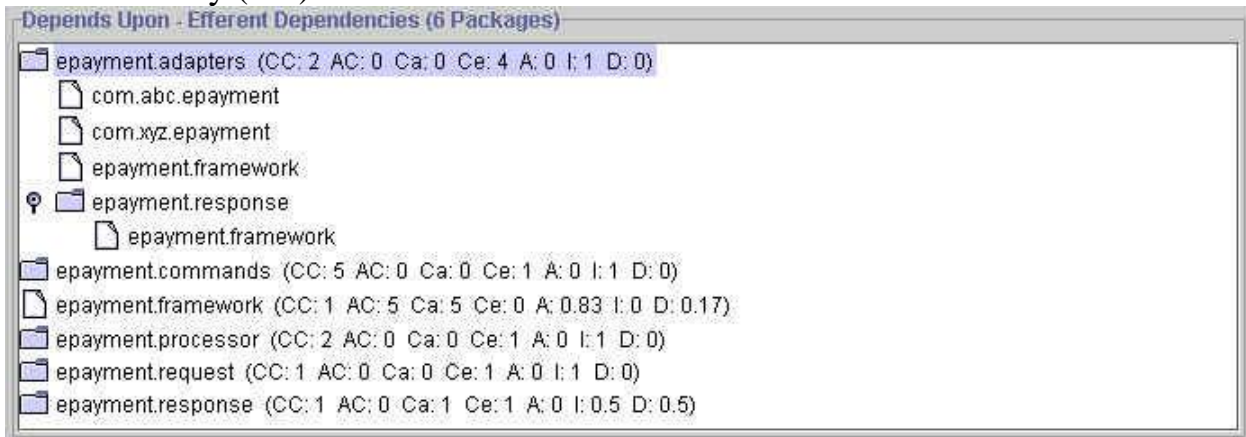


Рис. 1.6. Отображение метрик JDepend Plugin.

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).

Class Name	CBO	DIT	MPC	NOC	NT	RFC	WMC
org.netbeans.modules.metrics.ApproveMetricsPanel.Approv...	14	4	32	0	0	23	4
org.netbeans.modules.metrics.MetricValue	14	1	50	0	0	31	27
org.netbeans.modules.metrics.CBOMetric.Factory	6	1	2	0	0	5	3
org.netbeans.modules.metrics.MetricsTableModel	15	2	51	0	0	32	20
org.netbeans.modules.metrics.options.MetricsContextOption	6	4	18	0	0	19	6
org.netbeans.modules.metrics.ApproveMetricsPanel	11	5	41	0	0	24	2
org.netbeans.modules.metrics.TrafficNode.2	5	1	5	0	0	7	3
org.netbeans.modules.metrics.DITMetric.Factory	6	1	2	0	0	5	3
org.netbeans.modules.metrics.MethodMetrics	29	1	139	0	0	101	73
org.netbeans.modules.metrics.options.DITMetricSettingsBea...	2	2	3	0	0	5	3
org.netbeans.modules.metrics.TableMap	8	2	10	1	0	20	13
org.netbeans.modules.metrics.ApprovalsFile	18	1	85	0	0	51	16
org.netbeans.modules.metrics.TableSorter.1	4	2	8	0	0	10	5
org.netbeans.modules.metrics.RFCMetric	10	2	41	0	0	30	17
org.netbeans.modules.metrics.ApprovalAcceptor	4	1	0	0	0	0	0
org.netbeans.modules.metrics.ShowMetricsAction	13	5	54	0	1	42	22
org.netbeans.modules.metrics.Scanner	4	1	3	2	0	7	5
org.netbeans.modules.metrics.ApproveMetricsPanel.Approv...	4	2	3	0	0	6	3
org.netbeans.modules.metrics.TrafficNode.1	5	1	5	0	0	7	3
org.netbeans.modules.metrics.MetricsFilterFactory	16	2	57	0	2	50	19
org.netbeans.modules.metrics.MetricsLoader	11	1	39	0	0	32	10
org.netbeans.modules.metrics.NTMetric.Factory	6	1	2	0	0	5	3
org.netbeans.modules.metrics.MetricDetailsInvoker	15	2	29	0	0	32	12
org.netbeans.modules.metrics.ApprovalsFile.1	2	1	0	0	0	0	0

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]

1.1.8 Специализированные 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 узконаправлены, 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.

Контрольные вопросы:

1. Приведите классификацию метрик качества ПО.
2. Для каких целей служат метрики сложности программного обеспечения?
3. Приведите примеры метрик размера программного обеспечения.