**ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**"ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"**

Лабораторная работа № 5

Тема: «Метрики объектно-ориентированных программных систем»

Проверил: Выполнил:

\_\_\_\_\_\_\_\_\_\_\_\_\_ ст. гр. ПИ-18б

\_\_\_\_\_\_\_\_\_\_\_\_\_

Моргунов А.Г.

\_\_\_\_.\_\_\_\_.2021г.

Проверил

\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_

Донецк – 2021

**Задание**

Для своего программного проекта (для которого написали техническое задание) сделать оценку с использованием приведенных метрик.( Набор метрик Чидамбера и Кемерера, или Метрики Лоренца и Кидда, или Набор метрик Фернандо Абреу)

Вариант 11 – Метрики Лоренца и Кидда



Рисунок 5.1 – Диаграмма классов файловой системы

FileSystem – класс который отвечает за действия связанные с внутреннем строением файловой системы. В функции класса входит работа с FAT и суперблоком, а также управление потоком записи на диск. Методы данного класса описаны в таблице 5.1.

Таблица 5.1 – Методы класса FileSystem

|  |  |
| --- | --- |
| Название метода | Описание |
| formatting | Функция для форматирования диска |
| writeZeroCharactersOnDisc | Записывает в файл count нолей (если count больше размера файла, то файл расширяется) |
| getSuperBlock | Вернуть суперблок (для тестов) |
| findFirstFreeBlockNumber | Ищет первый свободный блок в FAT |
| calculateCountOfFreeBlocks | Вычисляет количество свободных блоков на диске по FAT |
| writeSuperBlock | Запись суперблока на диск |
| writeFAT | Записывает FAT на диск |
| deleteFATBlockChain | Отчищает цепочку блоков в FAT |
| newFATBlockChain | Создает новую цепочку блоков в FAT |
| resizeFATBlockChain | Изменяет размер цепочки блоков в FAT, которая начинается с blockNumber, на blocksCount блоков |
| getLastBlockOfChain | Возвращает номер последнего блока в FAT |
| addBlockToFATBlockChain | Добавляет 1 блок в конец цепочки блоков в FAT и возвращает номер добавленного блока |
| getFirstByteOfBlock | Возвращает первый байт блока с указанным номером |
| getFileStream | Возвращает указатель на поток работы с файлом |
| getNextFATChainBlockNumber | Берет значение указанной ячейки FAT |
| makeBlockLastInFATChain | Делает указанный блок последним в FAT цепочке (и удаляет оставшиеся за ним записи в FAT цепочке) |
| getBlockSize | Возвращает размер кластера |

Все служебные структуры, используемые системой, описаны в 1 разделе пояснительной записки.

File – класс, который объединяет в себе функции, предназначенные для взаимодействия с файлами: запись, чтение, изменение прав доступа, создание файла, удаление файла. Описание методов класса File представлены в таблице 5.2.

Таблица 5.2 – Описание методов класса File

|  |  |
| --- | --- |
| Метод | Описание |
| write | Записать в файл |
| read | Прочитать из файла |
| getDirectoryNote | Возвращает каталоговую запись файла |
| seekGet | Смещение указателя чтения |
| seekPut | Смещение указателя записи |
| addDirectoryNote | Запись каталоговой записи на диск |
| findDirectoryNote | Поиск каталоговой записи по имени |
| readDirectoryNote | Считывание 1 каталоговой записи |
| getPositionOfNoteInRootDirectory | Возвращает позицию каталоговой записи в корневом каталоге |
| createNewFile | Создает новый файл |
| findFreeDirectoryNote | Поиск свободной каталоговой записи, подходящей по размеру |
| deleteFile | Удаление файла |
| getCurrentGetPosition | Вернуть текущую позицию указателя чтения |
| changeAttributes | Изменить атрибуты файла |
| deleteData | Удалить данные файла |

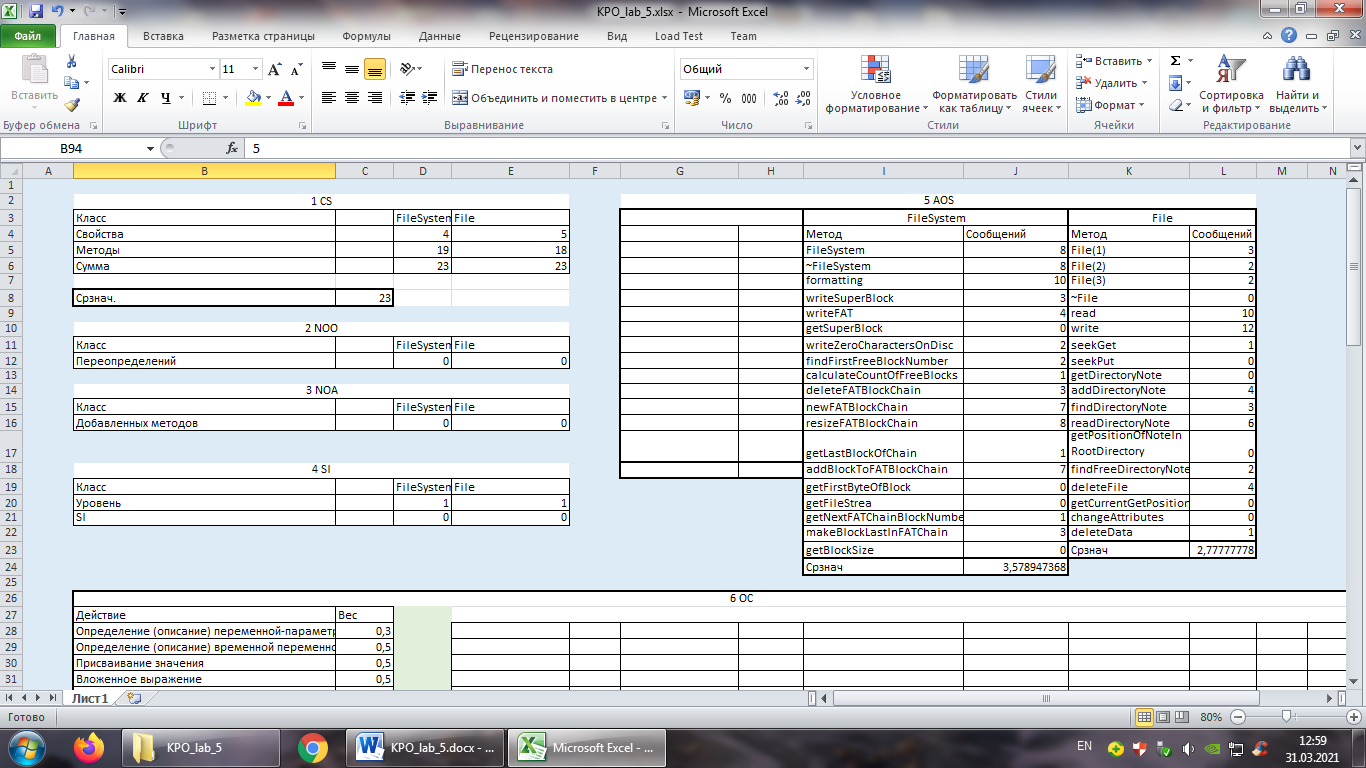
**Порядок выполнения работы**

**Метрика 1: Размер класса CS (Class Size)**

Общий размер класса определяется с помощью следующих измерений:

* общее количество операций (вместе с приватными и наследуемыми экземплярными операциями), которые инкапсулируются внутри класса;
* количество свойств (вместе с приватными и наследуемыми экземплярными свойствами), которые инкапсулируются классом.

Рекомендуемое значение CS  20 методов.



**Метрика 2: Количество операций, переопределяемых подклассом, NOO (Number of Operations Overridden by a Subclass)**

Переопределением называют случай, когда подкласс замещает операцию, унаследованную от суперкласса, своей собственной версией.

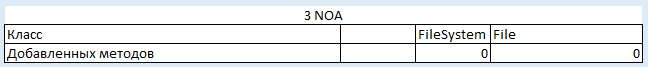
Рекомендуемое значение NOO  3 методов.



**Метрика 3: Количество операций, добавленных подклассом, NOA(Number of Operations Added by a Subclass)**

Подклассы специализируются добавлением приватных операций и свойств. С ростом NOA подкласс удаляется от абстракции суперкласса. Обычно при увеличении высоты иерархии классов (увеличении DIT) должно уменьшаться значение NOA на нижних уровнях иерархии.

Для рекомендуемых значений CS = 20 и DIT = 6 рекомендуемое значение NOA  4 методов (для класса-листа).

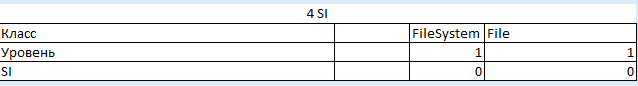


**Метрика 4: Индекс специализации SI (Specialization Index)**

Обеспечивает грубую оценку степени специализации каждого подкласса. Специализация достигается добавлением, удалением или переопределением операций:

SI = (NOO x уровень) /*Mобщ*,

где *уровень —* номер уровня в иерархии, на котором находится подкласс, *Мобщ —* общее количество методов класса.



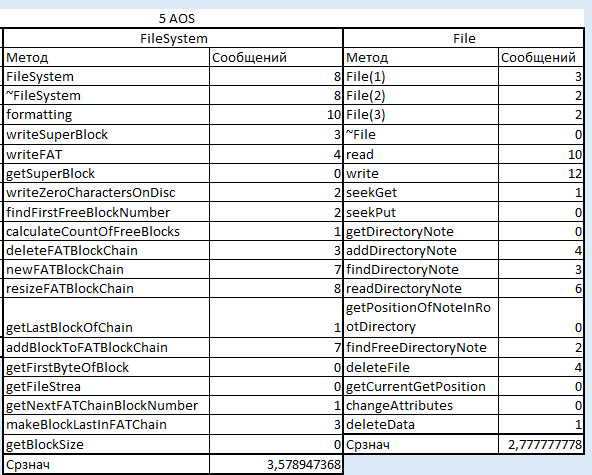
Чем выше значение SI, тем больше вероятность того, что в иерархии классов есть классы, нарушающие абстракцию суперкласса.

Рекомендуемое значение SI 0,15.

**Метрика 5: Средний размер операции OSAVG (Average Operation Size)**

В качестве индикатора размера может использоваться количество строк программы, однако LOC-оценки приводят к известным проблемам. Альтернативный вариант — *«количество сообщений, посланных операцией».*

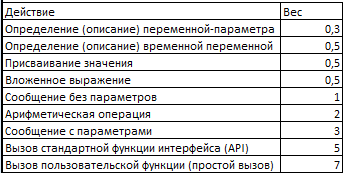
Рост значения метрики означает, что обязанности размещены в классе не очень удачно. Рекомендуемое значение OSAVG  9.



**Метрика 6: Сложность операции ОС (Operation Complexity)**

Сложность операции может вычисляться с помощью стандартных метрик сложности, то есть с помощью LOC- или FP-оценок, метрики цикломатической сложности, метрики Холстеда.

М. Лоренц и Д. Кидд предлагают вычислять ОС суммированием оценок с весовыми коэффициентами, приведенными в табл.



Поскольку операция должна быть ограничена конкретной обязанностью, желательно уменьшать ОС.

Рекомендуемое значение ОС  65 (для предложенного суммирования).

Класс FileSystem

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| FileSystem | | | | | | | | | | |
| Метод | 0,3 | 0,5 | 0,5 | 0,5 | 1 | 2 | 3 | 5 | 7 | Сумм |
| FileSystem | 1 | 1 | 1 | 0 | 3 | 1 | 2 | 4 | 0 | 32,3 |
| ~FileSystem | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 19 |
| formatting | 1 | 3 | 8 | 3 | 2 | 4 | 3 | 1 | 6 | 73,3 |
| writeSuperBlock | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 15 |
| writeFAT | 0 | 1 | 1 | 0 | 0 | 0 | 2 | 3 | 0 | 22 |
| getSuperBlock | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| writeZeroCharactersOnDisc | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 2 | 0 | 12,3 |
| findFirstFreeBlockNumber | 0 | 0 | 0 | 1 | 1 | 3 | 4 | 0 | 1 | 26,5 |
| calculateCountOfFreeBlocks | 0 | 1 | 1 | 2 | 0 | 1 | 0 | 3 | 0 | 19 |
| deleteFATBlockChain | 1 | 1 | 5 | 0 | 1 | 2 | 3 | 2 | 0 | 27,3 |
| newFATBlockChain | 1 | 2 | 4 | 2 | 0 | 6 | 4 | 1 | 2 | 47,3 |
| resizeFATBlockChain | 2 | 0 | 5 | 2 | 0 | 6 | 4 | 1 | 2 | 47,1 |
| getLastBlockOfChain | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 6,3 |
| addBlockToFATBlockChain | 1 | 1 | 1 | 0 | 0 | 1 | 5 | 0 | 2 | 32,3 |
| getFirstByteOfBlock | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 3,3 |
| getFileStrea | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| getNextFATChainBlockNumber | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 5,3 |
| makeBlockLastInFATChain | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 2 | 1 | 19,8 |
| getBlockSize | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

КлассFile

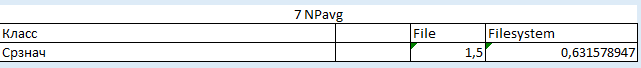
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| File | | | | | | | | | | |
| Метод | 0,3 | 0,5 | 0,5 | 0,5 | 1 | 2 | 3 | 5 | 7 | Сумм |
| File(1) | 10 | 0 | 27 | 1 | 1 | 5 | 0 | 2 | 1 | 45 |
| File(2) | 1 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 2 | 16,8 |
| File(3) | 3 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 2 | 16,9 |
| ~File | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| read | 2 | 5 | 12 | 3 | 0 | 10 | 0 | 2 | 5 | 75,6 |
| write | 2 | 5 | 13 | 4 | 0 | 1 | 0 | 2 | 8 | 79,6 |
| seekGet | 2 | 0 | 3 | 0 | 0 | 0 | 1 | 0 | 0 | 5,1 |
| seekPut | 2 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 2,6 |
| getDirectoryNote | 0 | 0 | 0 | 2 | 0 | 3 | 0 | 3 | 1 | 29 |
| addDirectoryNote | 1 | 0 | 0 | 2 | 1 | 4 | 0 | 2 | 1 | 27,3 |
| findDirectoryNote | 1 | 1 | 0 | 2 | 0 | 1 | 0 | 0 | 2 | 17,8 |
| readDirectoryNote | 1 | 0 | 0 | 3 | 2 | 3 | 2 | 2 | 1 | 32,8 |
| getPositionOfNoteInRootDirectory | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| findFreeDirectoryNote | 1 | 1 | 0 | 2 | 0 | 1 | 0 | 0 | 2 | 17,8 |
| deleteFile | 0 | 0 | 3 | 0 | 1 | 0 | 0 | 0 | 1 | 9,5 |
| getCurrentGetPosition | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| changeAttributes | 1 | 0 | 9 | 0 | 0 | 0 | 9 | 0 | 0 | 31,8 |
| deleteData | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 7,5 |

**Метрика 7: Среднее количество параметров на операцию NPAVG**

**(Average Number of Parameters per operation)**

Чем больше параметров у операции, тем сложнее сотрудничество между объектами. Поэтому значение NPAVGдолжно быть как можно меньшим.

Рекомендуемое значение NPAVG = 0,7.



**Метрика 8: Количество описаний сценариев NSS (Number of Scenario Scripts)**

Это количество прямо пропорционально количеству классов, требуемых для реализации требований, количеству состояний для каждого класса, а также количеству методов, свойств и сотрудничеств. Метрика NSS — эффективный индикатор размера программы.

Рекомендуемое значение NSS — не менее одного сценария на публичный протокол подсистемы, отражающий основные функциональные требования к подсистеме.



**Метрика 9: Количество ключевых классов NKC (Number of Key Classes)**

Ключевой класс прямо связан с коммерческой проблемной областью, для которой предназначена система. Маловероятно, что ключевой класс может появиться в результате повторного использования существующего класса. Поэтому значение NKC достоверно отражает предстоящий объем разработки. М. Лоренц и Д. Кидд предполагают, что в типовой ОО-системе на долю ключевых классов приходится 20-40% от общего количества классов. Как правило, оставшиеся классы реализуют общую инфраструктуру (GUI, коммуникации, базы данных).

Рекомендуемое значение: если NKC < 0,2 от общего количества классов системы, следует углубить исследование проблемной области (для обнаружения важнейших абстракций, которые нужно реализовать).

Т.к. в проекте нет графического интерфейса, а целью разработки являлась непосредственно реализация функций файловой системы, то все классы являются ключевыми (100%).



**Метрика 10: Количество подсистем NSUB (NumberofSUBsystem)**

Количество подсистем обеспечивает понимание следующих вопросов: размещение ресурсов, планирование (с акцентом на параллельную разработку), общие затраты на интеграцию.

Рекомендуемое значение: NSUB > 3.

Подсистемы:

* Подсистема интерпретатора
* Подсистема управления файлами
* Подсистема работы с суперблоком
* Подсистема работы с FAT
* Подсистема работы с файлами

