**Министерство образования и науки российской федерации**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ГОУ ВО «ПГТУ»)**

**Кафедра ИиСП**

**ОТЧЁТЫ ПО КУРСОВОМУ ПРОЕКТУ**

**По дисциплине « Разработка на платформе JAVA»**

**НАПРАВЛЕНИЯ ПОДГОТОВКИ**

**09.03.04. – ПРОГРАММНАЯ ИНЖЕНЕРИЯ**

**Разработал: студент ПС-31**

**Матюков Николай Юрьевич**

**Проверил:**

**Курбатов Е.М.- доцент кафедры ИиСП**

**Йошкар-Ола**

**2019 год**

Оглавление

Постановка задачи……………………………………………………………..3

Детали реализации……………………………………………………………..5

Используемые библиотеки и фреймворки……………………………………9

Руководство пользователя……………………………………………………..9

Тестирование………………………………………………………………….10

Отчет по курсовому проекту.

Тема: Patch visualizer

Выполнили: Савельев А.А., Матюков Н.Ю.

Постановка задачи

Реализовать программу визуализации дельты (изменений) между двумя версиями файлов. Изменения должны визуализироваться в html формате.

Пример файла патча: https://github.com/mrkurbatov/diffTest/commit/0a6463fb6372f5a8ba3fcc5e27eaa48d6db308be.patch

Пример визуализации:

https://github.com/mrkurbatov/diffTest/commit/0a6463fb6372f5a8ba3fcc5e27eaa48d6db308be

Функциональные требования:

* Отображение информации о патче (автор, время изменения, имя файла)
* Построчная визуализация сравнения как на Github, но отображать файл полностью
* Два режима визуализации:
  + Unified - https://github.com/mrkurbatov/diffTest/commit/0a6463fb6372f5a8ba3fcc5e27eaa48d6db308be?diff=unified
  + Split - https://github.com/mrkurbatov/diffTest/commit/0a6463fb6372f5a8ba3fcc5e27eaa48d6db308be?diff=split

Общие нефункциональные требования:

* Java 8
* Следование SOLID принципам
* Репозиторий на github
* Maven для сборки проекта
* JUnit 4 для модульного тестирования
* Коэффициент покрытия тестами не менее 70%
* Состав команды не более 3 человек
* Отчет о работе от каждого участника проекта:
* Постановка задачи
* Детали реализации (модель, алгоритмы)
* Используемые библиотеки и фреймворки
* Руководство пользователя
* Тестирование

Детали реализации

Был реализован класс парсинга патч файла PatchFileParser. Конструктор класса принимает один параметр - путь до файла. Метод parse() читает файл и парсит его на блоки, которые содержат такие параметры как отступ, высота изменения как для текущего коммита, так и для предыдущего, сами строки.

Был реализован класс BaseHtmlGenerator, от которого унаследованы HtmlGeneratorModeSplit и HtmlGeneratorModeUnified. Конструктор класса принимает 3 аргумента: список распарсенных строк, путь до файла вывода и информацию о пользователе. В зависимости от режима метод generate() класса HtmlGenerator вызывает метод одного из обьектов классов HtmlGeneratorModeSplit или HtmlGeneratorModeUnified и генерируется Html файл

Был реализован класс DifferenceBlockValidator который проверяет на соответствие патч файл и файл с данными. Он проходит по блокам с изменениями, вычисляет в зависимости от типа строки и отступа позицию строки в файле с данными. Если строки не совпадают выбрасывается исключение.

Пакет controller содежит в себе обработчики параметров и самого приложения. CommandLineArgumentsControllerImpl реализует интерфейс CommandLineArgumentsController. (рисунок 1)

Пакет model содержит в себе все модели используемые для реализации данного приложения. Так же модели содержат интерфейсы, которые реализутся. (рисунок 2)

Пакет service содержит все сервисы по парсингу начальных файлов и генерации html страницы.

Диаграмма классов пакета controller

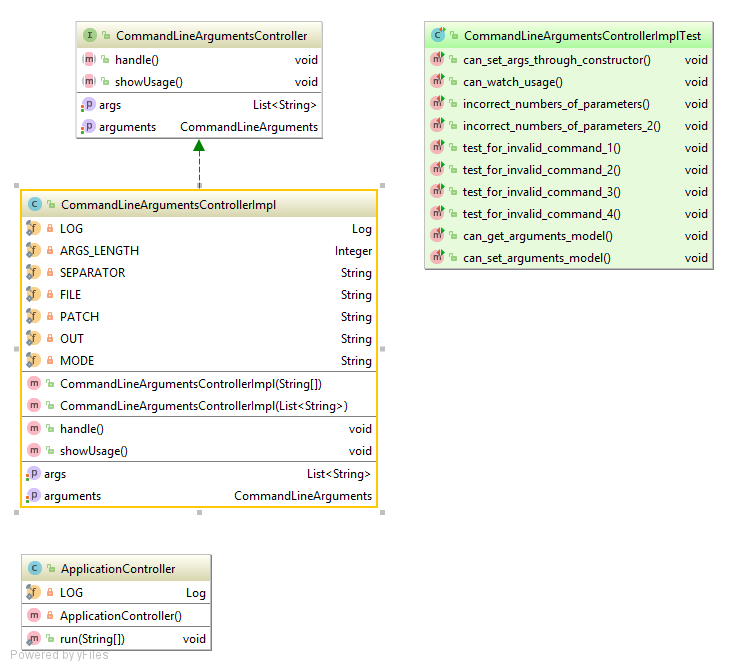


рисунок 1

Диаграмма классов пакета model

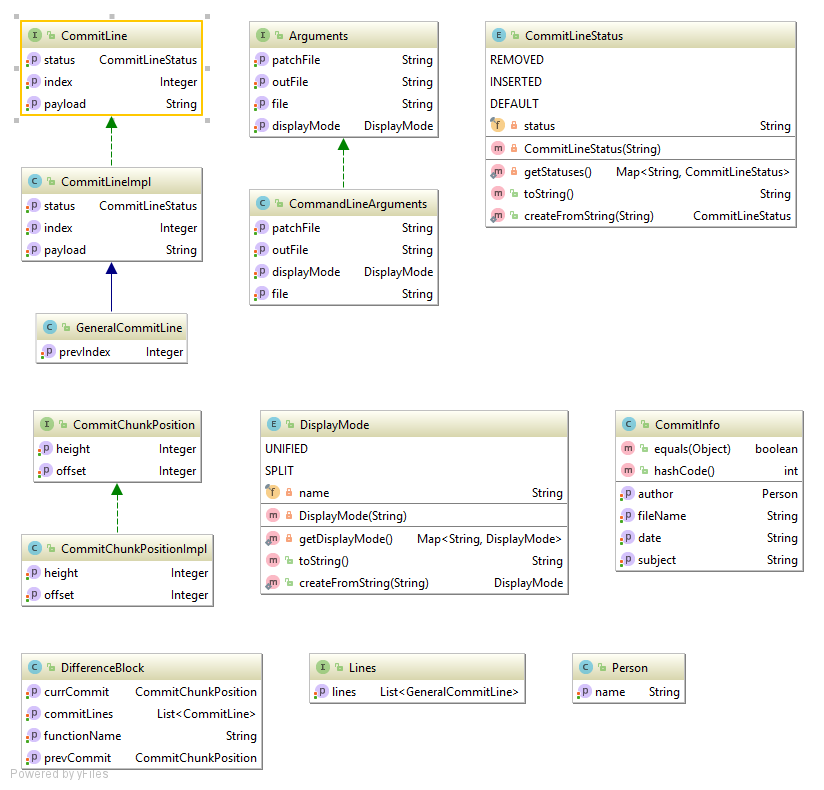


рисунок 2

Диаграмма классов пакета service

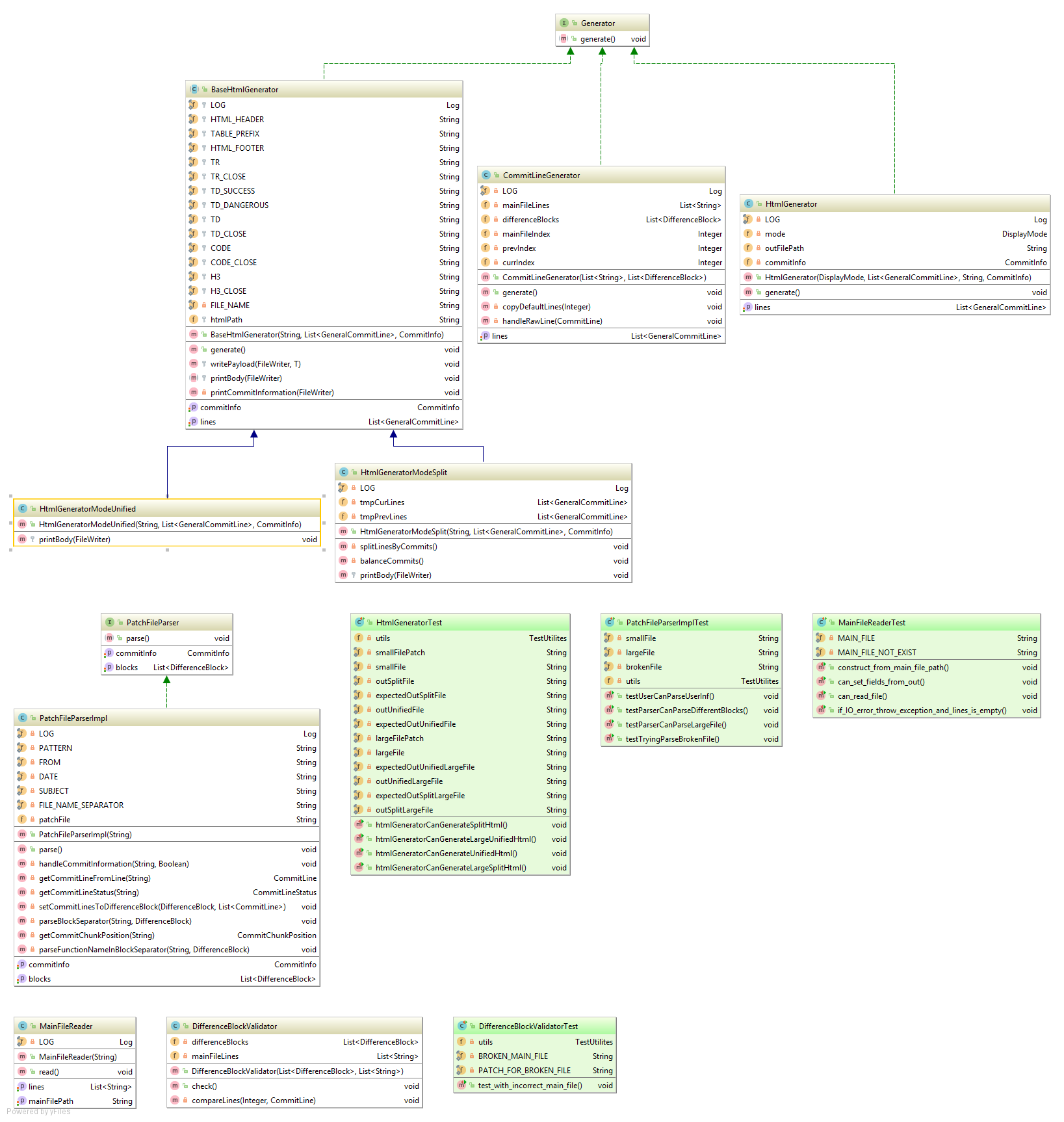


рисунок 3

Используемые библиотеки и фреймворки

При разработке нам понадобились следующие библиотеки и фреймворки:

* junit - фреймворк для unit-тестов;
* Apache Maven – фреймворк для автоматизации сборки проекта
* Bootstrap - фреймворк для отображения HTML кода
* commons-logging - для вывода информации в log.

Руководство пользователя

Программа принимает 4 параметра:

* “—file” - исходный файл
* “--patch” - файл патча
* “--out” - файл визуализации
* “--mode” - режим визуализации

Тестирование

Приложение было протестировано разными наборами входных значений. Файлы с минимальными изменениями, с изменениями в разных местах кода. Также некорректные файлы. Тестировались как отдельные классы, так и результат выдаваемый программой. Общий процент покрытия тестами составляет 80%

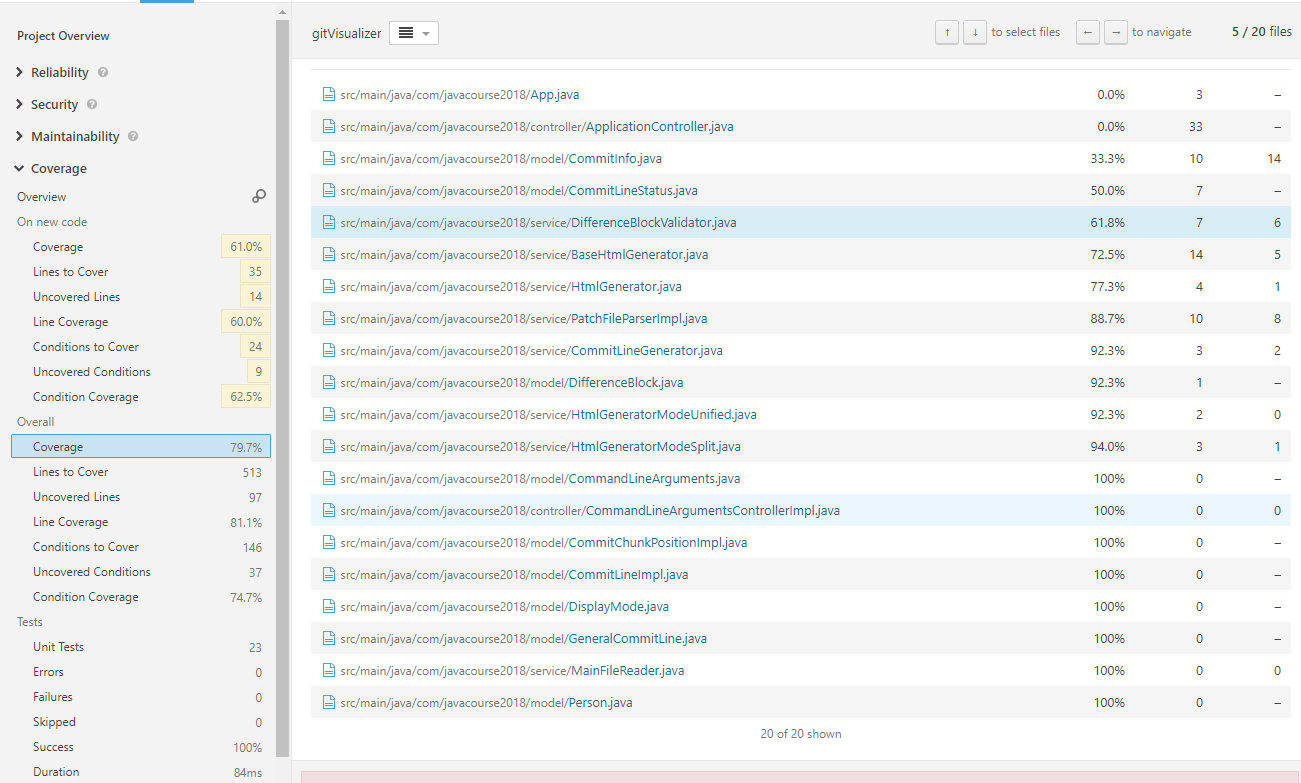


рисунок 4