МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тихоокеанский государственный университет»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

«Разработка и реализация системного приложения “Файловый менеджер с поддержкой архивов”»

Текстовый документ курсовой работы

по дисциплине «Операционные системы и системное ПО»

КР. 180008865.ТД

Выполнил студент Пшеничный Д.О.

Факультет, группа ФКФН, ПО(аб) – 81

Руководитель работы Федосеев А.А.

Виза: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(доработать, к защите и т.д.)

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

Хабаровск – 2021г.

СОДЕРЖАНИЕ

[1 Постановка задачи 3](#_Toc73947434)

[2 Диаграмма классов 4](#_Toc73947435)

[2.1 Описание классов 4](#_Toc73947436)

[Заключение 10](#_Toc73947438)

[Список использованных источников 11](#_Toc73947439)

[Приложение А 12](#_Toc73947440)

# ПОСТАНОВКА ЗАДАЧИ

Требуется реализовать интерактивное приложение “Файловый менеджер с поддержкой архивов”, ­ обладающее следующими свойствами: навигация по директориям, открытие и создание архивов. Для разработки приложения будет использован язык программирования C#, и среды разработки IntelliJ Rider и Microsoft VisualStudio.

# ДИАГРАММА КЛАССОВ

На рисунке 1 рассматривается диаграмма классов.

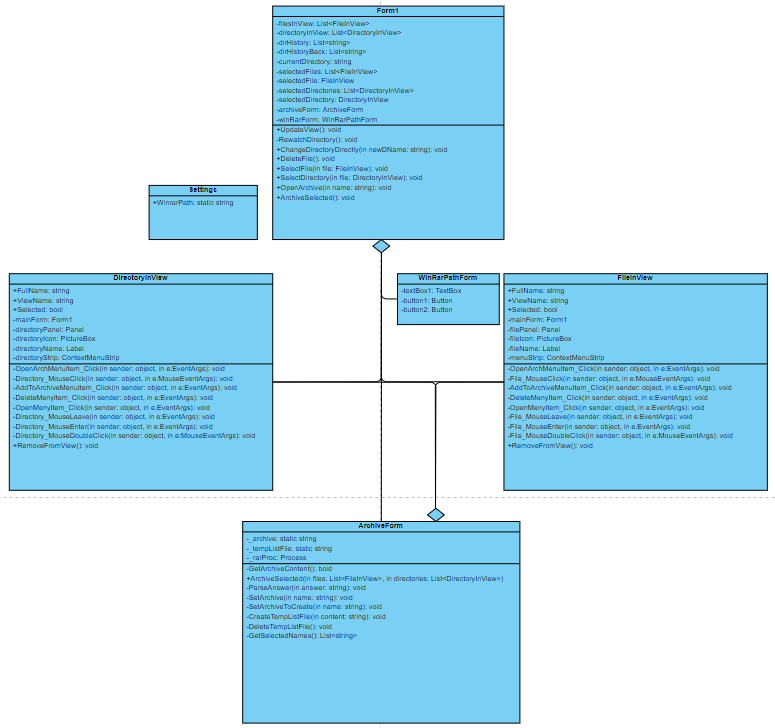


Рисунок 1 - Диаграмма классов

* 1. Описание классов
     1. FileInView – класс, представляющий файл в отображении. Отвечает за управления операциями с файлами в представлении.

Поля:

* string FullName – полное имя файла;
* string ViewName – имя файла в отображении;
* bool Selected – флаг выбора файла;
* Panel filePanel– панель, хранящая визуальные компоненты представления файла;
* Form1 mainForm– главная форма программы;
* PictureBox fileIcon – иконка файла;
* Label fileName – имя файла в отображении;
* ContextMenuStrip menuStrip – контекстное меню, используемое для файла;

Методы:

* void RemoveFromView() – удаляет визуальные компоненты представления файла с формы;
  + 1. DirectoryInView – класс, по составу и реализации аналогичный классу FileInView, но работающий для директорий.

2.1.3 Form1 – главный класс, представляющий основное окно приложения

Поля:

* DownloadPool downloadPool – поле, отвечающее за загрузку файла;
* Stage window – контейнер, используемый для вывода главного окна;
* TableView<DownloadThread> table – таблица, выводящая основную информацию о загружаемом файле.

Методы:

* static void main(String[] args) – метод, запускающий приложение;
* void stop() – переопределение метода, отвечающее за действия после закрытия приложения;
* void setTable() – создание таблицы;
* void setButtons() – настройка кнопок;
* void start(Stage stage) – переопределение метода, отвечающий за обработку главного окна при открытие.

2.1.4 DownloadMetadata – класс, содержащий все метаданные о загрузке

Поля:

* SimpleObjectProperty<URL> url – ссылка на файл;
* SimpleObjectProperty<Integer> downloadID – id загрузки;
* SimpleObjectProperty<String> filename – имя файла;
* SimpleObjectProperty<Long> size = new SimpleObjectProperty<>() – размер файла;
* SimpleObjectProperty<Boolean> accelerated – поле, показывающее поддерживает ли сервер разбиение файла на части;
* SimpleObjectProperty<DownloadStatus> status – статус загрузки.

2.1.5 DownloadPart – класс, представляющий часть загрузки

Поля:

* SimpleObjectProperty<DownloadPartMetadata> metadata – метаданные о части загружаемого объекта;
* ConcurrentLinkedQueue<DownloadAction.Command> queueCommand – очередь команд;
* ConcurrentLinkedQueue<DownloadAction.Response> queueResponse – очередь ответов.

Методы:

* void pause() – приостановление загрузки;
* void resume() – возобновление загрузки;
* void stop() – остановка загрузки;
* BufferedInputStream getConnectionStream() – настройка соединения с загружаемым файлом;
* boolean copyToStream(BufferedInputStream inputStream, BufferedOutputStream fileStream) – копирует данные из inputStream в fileStream;
* void download() – начинает загрузку файла;
* void run() – переопределение метода для запуска класса в отдельном потоке.
  + 1. DownloadPartMetadata – класс, представляющий метаданные о каждой части загрузки

Поля:

* SimpleObjectProperty<Integer> partID – id части загрузки;
* SimpleObjectProperty<DownloadStatus> status – статус загрузки;
* String filename – имя файла;
* DownloadMetadata downloadMetadata – метаданные;
* SimpleObjectProperty<Part> part – размер загружаемой части;
* SimpleObjectProperty<Long> completedBytes – количество загруженных байтов.
  + 1. DownloadPartThread – класс, представляющий поток для загружаемой части файла

Поля:

* Thread thread – поток для загрузки;
* SimpleObjectProperty<DownloadPart> downloadPart – загружаемая часть;
* ConcurrentLinkedQueue<DownloadAction.Command> queueCommand – очередь команд;
* ConcurrentLinkedQueue<DownloadAction.Response> queueResponse – очередь ответов;
* SimpleObjectProperty<DownloadPartMetadata> downloadPartMetadata – метаданные о загружаемой части файла.
  + 1. DownloadPool – класс, отвечающий за загрузку файла

Поля:

* ObservableList<DownloadThread> downloadThreads – список потоков для загрузки.

Методы:

* void waitUntilCommand(DownloadThread downloadThread, DownloadAction.Response response) – метод, получающий response для downloadThread;
* void issueCommand(DownloadThread downloadThread, DownloadAction.Command command) – метод, выдающий command для downloadThread;
* void stopDownload(DownloadThread downloadThread) – останавливает загрузку для конкретного DownloadThread;
* void pauseDownload(DownloadThread downloadThread) – приостанавливает загрузку для конкретного DownloadThread;
* void resumeDownload(DownloadThread downloadThread) - возобновляет загрузку для конкретного DownloadThread;
* void removeDownload(DownloadThread downloadThread) – останавливает и удаляет конкретный DownloadThread;
* void stopAll() – останавливает все загрузки;
* void joinThread(DownloadThread downloadThread) – присоединяет downloadThread;
* void joinThreads() – присоединяет все downloadThread;
* void newDownload(String url) – старт загрузки нового файла по указанному в адресе url.
  + 1. DownloadStatus – класс, представляющий различные состояния загрузки файла
    2. DownloadThread – класс, представляющий поток для загрузки. Включает объект Download и очереди для взаимодействия с потоками

Поля:

* SimpleObjectProperty<DownloadMetadata> downloadMetadata – метаданные о загружаемом файле;
* SimpleObjectProperty<Download> download – объект загрузки файла;
* Thread thread – поток;
* ConcurrentLinkedQueue<DownloadAction.Command> queueCommand – очередь команд;
* ConcurrentLinkedQueue<DownloadAction.Response> queueResponse – очередь ответов.
  + 1. Part – класс, представляющий размер загружаемой части

Поля:

* long startByte – начальный байт;
* long endByte – конечный байт.

# ЗАКЛЮЧЕНИЕ

Выполнение курсовой работы завершилось созданием программного продукта, который обладает функциональными возможностями, определенными заданием.

В приложении А приведены скриншоты, полученные в результате тестирования программы.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Metanit [Электронный ресурс]: онлайн учебник. – Режим доступа: https://metanit.com/ (дата обращения 12.25.2015)

2. MSDN [Электронный ресурс]: документация. [Microsoft]. – Режим доступа: https://docs.microsoft.com/ (дата обращения 12.25.2015)

3. Бен Албахари и Джозеф Албахари. C# 7.0 in a Nutshell: The Definitive Reference, 28 октября 2017 г.

4. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж.: Паттерны объектно-ориентированного проектирования — Прогресс книга, 2020 г. — 448 с.

# ПРИЛОЖЕНИЕ А

(обязательное)

Функционирование приложения

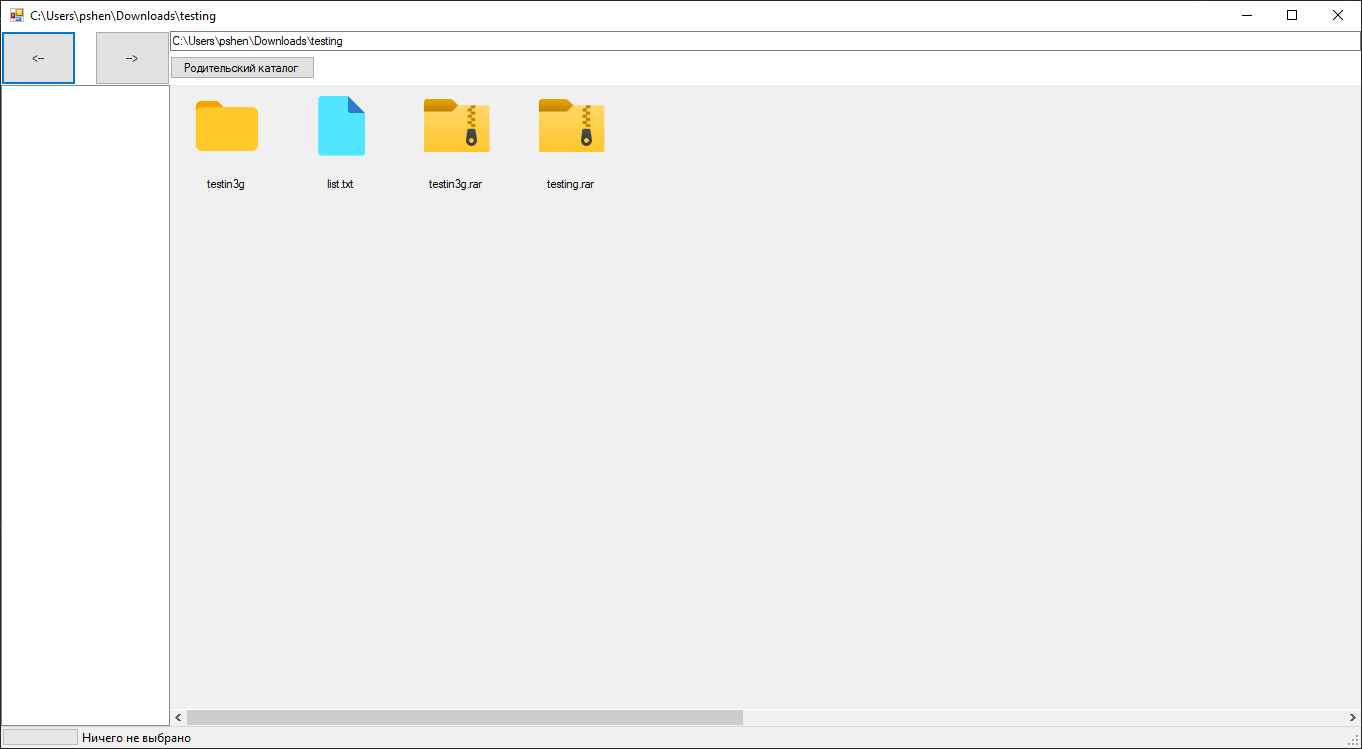


Рисунок А.1 – Основной вид приложения

Продолжение приложения А

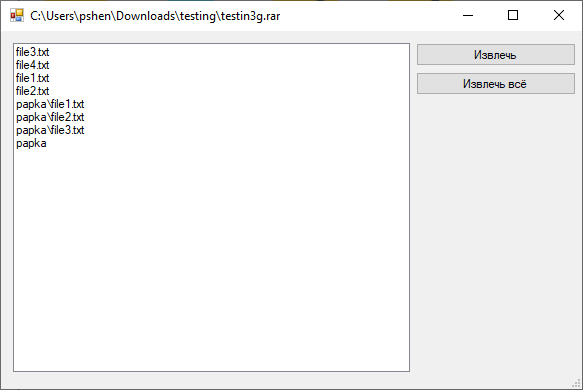


Рисунок А.2 – Окно работы с архивами

Продолжение приложения А

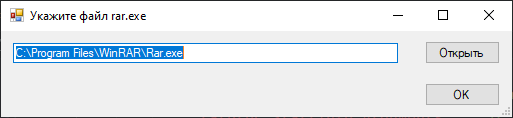


Рисунок А.3 – Окно задания пути к консольной версии WinRar.