**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное автономное образовательное учреждение**

**высшего образования**

**«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

ИНСТИТУТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И

ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

КАФЕДРА ТЕХНОЛОГИЙ ПРОГРАММИРОВАНИЯ

Направление: 09.03.03 – «Прикладная информатика»

КУРСОВАЯ РАБОТА

ПО ДИСЦИПЛИНЕ «ПРОЕКТИРОВАНИЕ ЧЕЛОВЕКО-МАШИННОГО ИНТЕРФЕЙСА»

ПО ТЕМЕ

**«ПОЛИКЛИНИКА»**

Студент 2 курса

Группа 09-952

“\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.    Е.А.Иванов

Научный руководитель

к.ф.-м.н.,

доцент КТП ИВМиИТ КФУ

“\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.     И.С.Балафендиева

Казань-2020

Содержание

Введение

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Требования к системе

1.2 Основной функционал

1.3 Глоссарий

1.4 Действующие лица

1.5 Варианты использования

2. ОПИСАНИЕ ПРИЛОЖЕНИЯ

2.1 Хранение информации

2.2 Классы системы

2.3 Структура системы

2.4 Пользовательский интерфейс

3. РУКОВОДСТВО ПО ИСПОЛЬЗОВАНИЮ СИСТЕМЫ

4. ЗАКЛЮЧЕНИЕ

Введение

В наше время организациям приходится работать с большим количеством информации. Обработка этого объёма информации является трудной задачей, поэтому организации прибегают к помощи автоматизированных систем, которые облегчают эту задачу.

Целью данной курсовой работы является создание системы, позволяющей поликлинике вести учёт докторов и пациентов.

Процесс создания данного приложения должен сформировать у автора навыки проектирования человеко-машинного интерфейса.

1.АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Требования к системе

Пользователю приложения должны быть доступны операции для учета докторов и пациентов, такие как добавление, удаление и поиск, а также анализ заболеваемости пациентов. Для пациентов также существует поиск по заданным критериям: дата рождения, район проживания, добавление новых справок о болезни.

Необходимая информация будет храниться в следующем формате: Доктор: ID (long), имя, фамилия(String). Пациент: ID(long), имя, фамилия(String), дата рождения(Date), справки(Date[]), район(District\*)

Поле ID создаётся из текущей даты в формате миллисекунд, это позволяет каждый раз создавать уникальный номер.

\*District – класс перечисления, который содержит все районы города

1.2 Основной функционал

Для того, чтобы добавить нового доктора/пациента, пользователю нужно нажать кнопку под списком докторов/пациентов и ввести необходимые данные.

Чтобы удалить доктора/пациента, нужно выбрать его из списка и нажать кнопку в разделе информации о докторе/пациенте.

В приложении должна быть реализована функция поиска пациентов по заданному критерию и анализа заболеваемости пациентов.

1.3 Глоссарий

|  |  |
| --- | --- |
| **Термин** | **Значение** |
| Доктор | Работник поликлиники |
| Пациент | Человек, находящийся на учете в поликлинике |
| Список докторов | Список людей, работающих в поликлинике |
| Список пациентов | Список людей, находящихся на учете в поликлинике |
| Информация | Раздел с информацией о докторе/пациенте |
| Справка | Запись о том, что пациент был в указанную дату на приёме |

1.4 Действующие лица

Пользователь приложения - добавляет/удаляет новых докторов/пациентов, добавляет новые справки пациентам, смотрит анализ заболеваемости.

Поисковая система - ищет и выводит пациентов по заданным критериям.

1.5 Варианты использования

1. Добавление/удаление новых докторов/пациентов
2. Поиск пациентов/докторов по именам
3. Поиск пациентов по заданному критерию
4. Анализ заболеваемости по районам и датам справок о болезни

1.5.1 Добавить/удалить доктора/пациента

Пользователь нажимает кнопку добавления под списком, после чего выводится модальное окно для ввода необходимых данных. После нажатия кнопки “Добавить” будет выведено сообщение об успешном добавление доктора/пациента в систему.

Если не будут заполнены все необходимые поля, то будет выведено сообщение с просьбой их заполнить.

1.5.2 Поиск пациентов/докторов по именам

Пользователь вводит имя и/или фамилию в строку поиска над списком, после чего список обновляется, выводя только тех, кто подходит под введённые значения.

Если никого не будет найдено, будет выведен пустой список.

1.5.3 Поиск пациентов по заданному критерию

Пользователь выбирает необходимые критерии над списком пациентов, после чего список обновляется, выводя только тех, кто подходит под все заданные критерии.

Если никого не будет найдено, будет выведен пустой список.

1.5.4 Анализ заболеваемости по районам и датам справок о болезни

Пользователь выбирает необходимый интервал времени и нажимает кнопку “Анализировать”. Появится круговая диаграмма, которая в процентном соотношении покажет в каких районах было больше посещений за выбранный промежуток времени.

2.ОПИСАНИЕ ПРИЛОЖЕНИЯ

2.1 Выбор программного средства

Для создания проекта был выбран язык программирования Java и платформа JavaFX + Scene Builder. В качестве среды разработки была выбрана IntelliJ IDEA

2.1 Хранение информации

Для хранения информации было создано три текстовых файла - для хранения информации о пользователях, докторах и пациентах

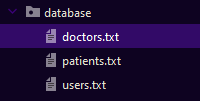


Рис.1 (структура файлов в папке)

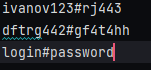


Рис.2(структура хранения пользователей)

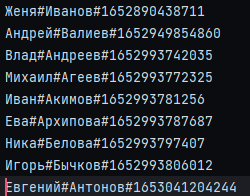


Рис.3(структура хранения докторов)

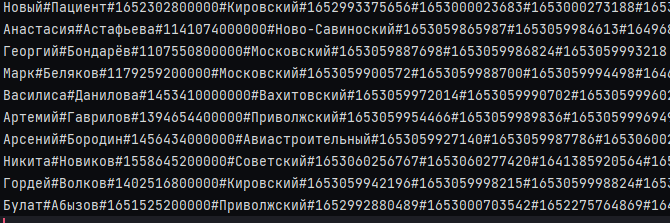


Рис.4(структура хранения пациентов)

Каждая строка в файле - это отдельная запись об объекте. Все необходимые данные разделены символом #.

2.2 Классы системы

2.2.1 Класс доктор

Содержит всю информацию о докторе, а также методы для обработки информации, касающейся докторов

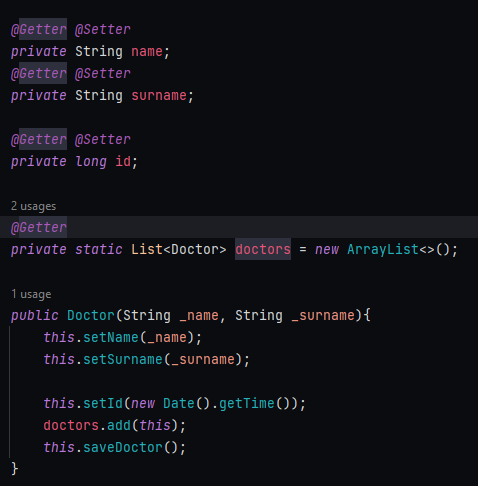


Рис.5(основные поля и конструктор класс доктор)

2.2.2 Класс пациент

Содержит всю информацию о пациенте, а также методы для обработки информации, касающейся пациентов



Рис.6(основные поля класса пациент)

2.2.3 Перечисление районов

Содержит в себе все районы города

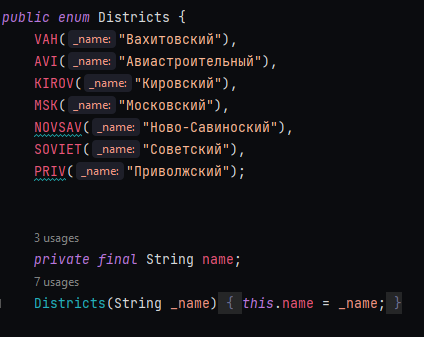


Рис.7(перечисление районов)

2.3 Структура системы

Для удобства некоторые классы были разделены на папки

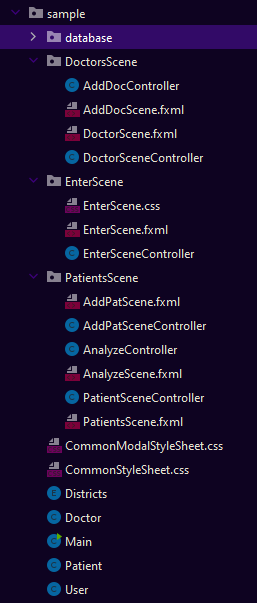


Рис.8(файловая структура)

2.4 Пользовательский интерфейс

2.4.1 Основные блоки

Были использованы блоки со строгими границами и минимальным количеством лишних эффектов

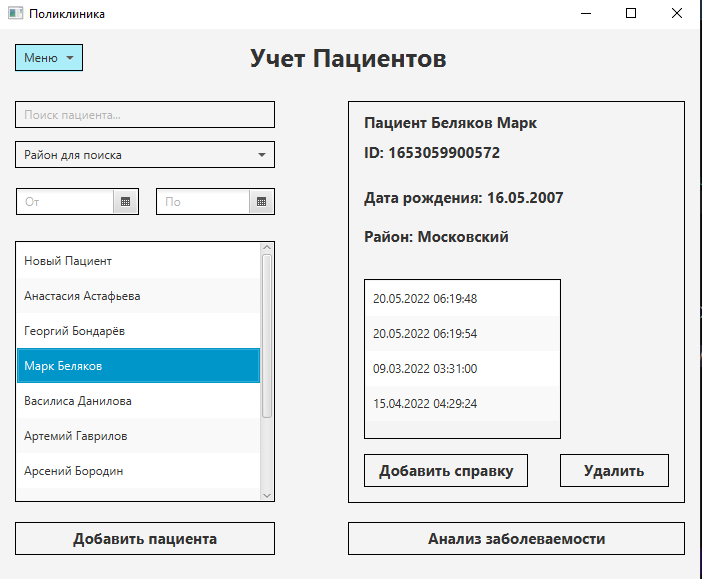


Рис.9(страница учета пациентов)

2.4.2 Кнопки

Было выделено два класса кнопок для удаления и для всех остальных действий



Рис.10(кнопка для всех действий)

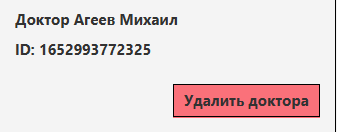


Рис.11(кнопка для удаления)

3 РУКОВОДСТВО ПО ИСПОЛЬЗОВАНИЮ СИСТЕМЫ

При запуске программы открывается окно входа в систему, где необходимо ввести логин и пароль

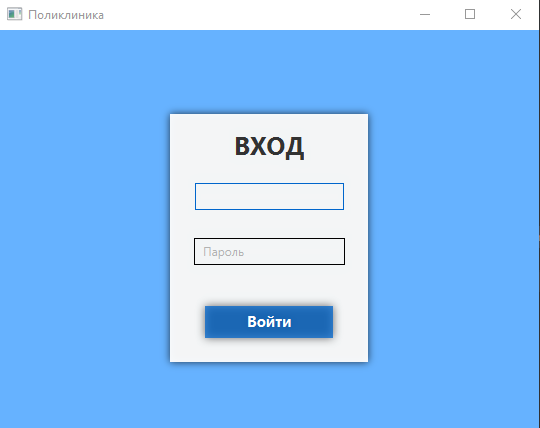


Рис.12(вход в систему)

При успешном входе открывается окно учёта докторов. В верхнем левом углу находится кнопка “Меню” для навигации между страницами учёта докторов и пациентов

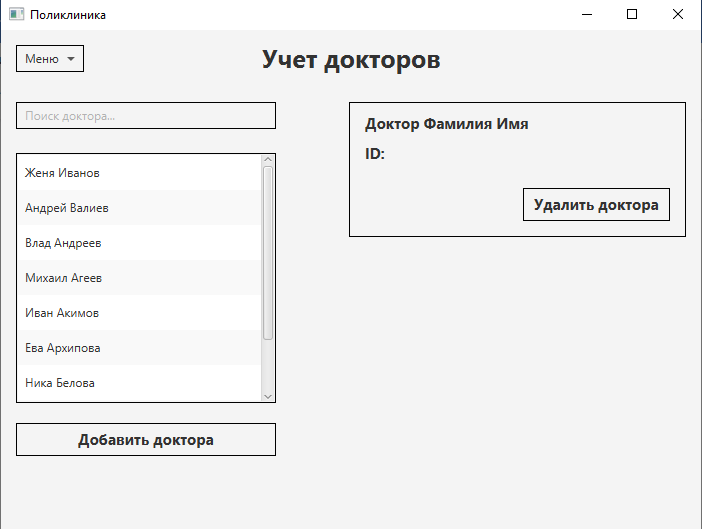


Рис.13(страница докторов)

Для добавления докторов или пациентов выходит модальное окно для ввода необходимых данных

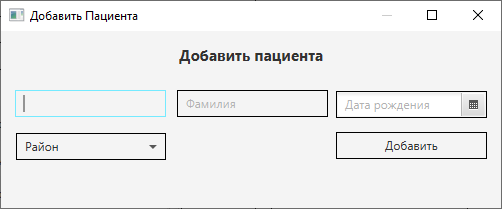


Рис.14(окно добавления пациента)

Для анализа заболеваемости необходимо выбрать временной интервал и нажать кнопку “Анализ”

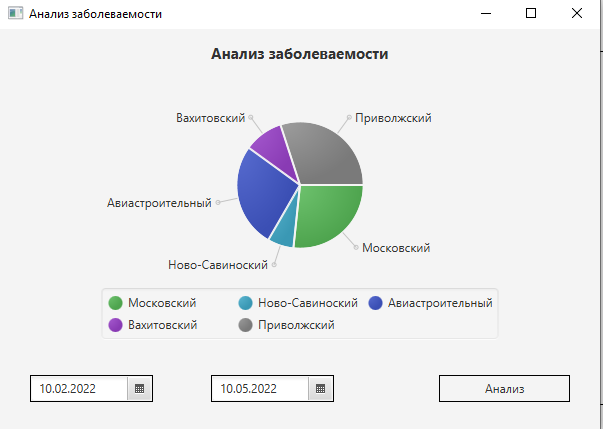


Рис.15(страница анализа заболеваемости)

4 ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы была создана система, позволяющая вести учёт докторов и пациентов в поликлинике. Был получен опыт в проектировании человеко-машинного интерфейса, а также использовании таких средств как JavaFX и Scene Builder.