

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

КУРСОВАЯ РАБОТА
ЗАЩИЩЕНА С ОЦЕНКОЙ
РУКОВОДИТЕЛЬ

Ст. преподаватель
должность, уч. степень, звание

подпись, дата

Е.О. Шумова
инициалы, фамилия

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К КУРСОВОЙ РАБОТЕ

Разработка системы классов для обеспечения работы поликлиники

по дисциплине: ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4831

подпись, дата

К. А. Корнющенко
инициалы, фамилия

Санкт-Петербург 2020

23) Разработка системы классов для обеспечения работы поликлиники (*В системе должны поддерживаться режимы учета докторов и пациентов, поиска пациента по заданному критерию, анализа заболеваемости по районам, по месяцам*).

Основная часть:

1. Постановка задачи
 1. 1.1. Анализ предметной области
 2. 1.2. Формулировка технического задания
2. Проектирование классов
 1. 2.1. Классы сущностей
 2. 2.2. Управляющие классы
 3. 2.3. Интерфейсные классы
 4. 2.4. Используемые паттерны проектирования
3. Разработка приложения
 1. 3.1. Разработка интерфейса приложения
 2. 3.2. Реализация классов
 3. 3.3. Разработка тестового приложения
4. Тестирование

1. Постановка задачи

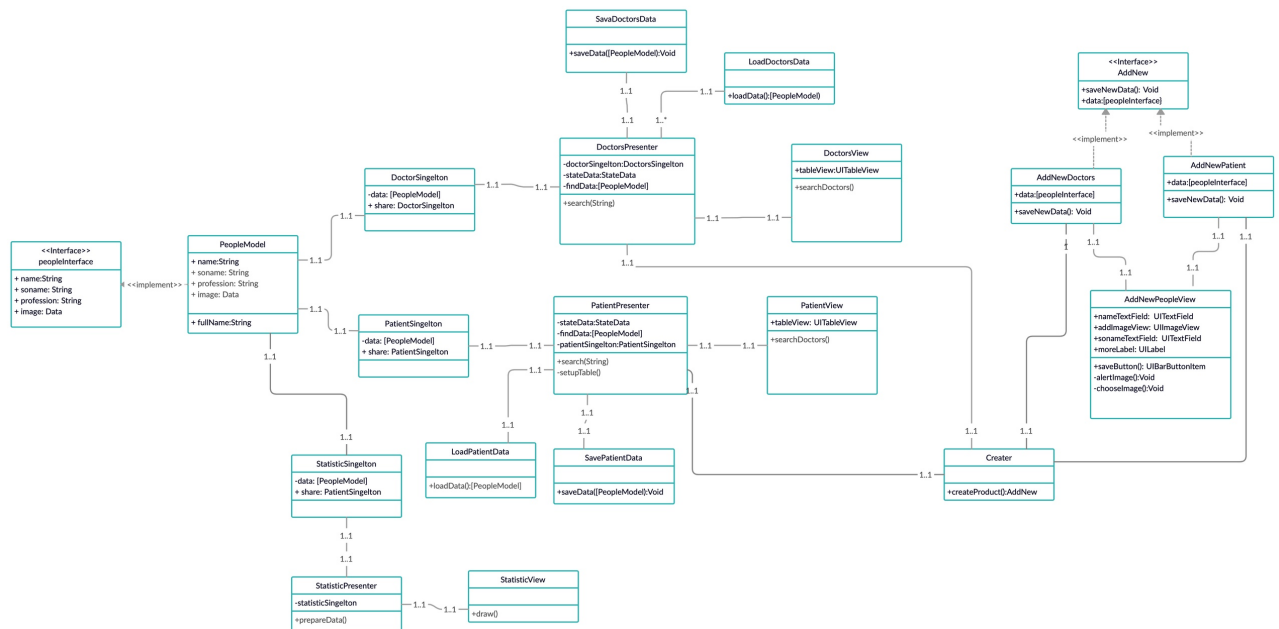
1.1 Анализ предметной области

Разработка информационной системы для обеспечения работы поликлиники важная часть для упрощения жизни врачей и клиентов данного медицинского заведения. Данная информационная система должна позволить посетителям легко и просто записаться к врачам, отслеживать статистику и историю посещений. А также врачи могут смотреть историю болезней и всю информацию посетителей через одну информационную систему, что упрощает и оптимизирует их рутинную работу.

1.2 Формулировка технического задания

В ходе выполнения курсовой работы необходимо разработать информационную систему для обеспечения полноценной работы поликлиники. В системе должны поддерживаться режимы учета докторов и пациентов, поиска пациентов и докторов по заданному критерию, анализ заболеваемости по районам и месяцам. Информация о докторах и пациентах должна содержать имя и фамилию, а также фотографию. У пациентов необходимо сохранять историю посещения врачей. Поиск пациентов и докторов производится по ФИО человека или профессии. В системе должна быть возможность добавлять новых людей, как пациентов, так и врачей. А также возможность редактировать и удалять любую информацию.

2. Проектирование классов



2.1. Классы сущностей

PeopleModel – класс сущности, который используется в качестве модели для хранения данных об одном докторе или посетителе.

2.2 Управляющие классы

DoctorSingleton – класс для хранения и работы с данными о врачах.

PatientSingleton – класс для хранения и работы с данными о пациентах.

StatisticSingleton – класс для хранения и работы с данными о статистике.

StatisticPresenter – класс для подготовки данных статистики для отображения.

LoadPatientData – класс для загрузки сохраненных данных о пациентах.

SaveDoctorsData – класс для сохранения данных о врачах.

DoctorsPresenter – класс для подготовки данных о врачах для отображения.

PatientPresenter – класс для подготовки данных о пациентах для отображения.

SavePatientData – класс для сохранения данных о пациентах.

LoadDoctorsData – класс для загрузки сохраненных данных о врачах.

Creator – класс для создания объекта интерфейса AddNew.

AddNewDoctors – класс обработки новых данных о врачах.

AddNewPatient – класс обработки новых данных о пациентах.

2.3 Интерфейсные классы

StatisticView – класс для работы с графическим интерфейсом статистики.

DoctorsView – класс для работы с графическим интерфейсом связанный с информацией о врачах.

PatientView – класс для работы с графическим интерфейсом связанный с информацией о пациентах.

AddNewPeopleView – класс для работы с графическим интерфейсом для добавления новых данных.

2.4 Используемые паттерны проектирования

В программе используются 3 проектирующих паттерна программирования Одиночка для удобного хранения и передачи данных между классами. Данный паттерн реализуется в классах DoctorSingleton, PatientSingleton, StatisticSingleton.

Так же в программе Фабричный метод, который служит для создания объектов, которые поддерживают интерфейс AddNew. Фабричный метод реализован с помощью интерфейса AddNew, который поддерживают 2 класса AddNewDoctors и AddNewPatient и для создания и выбора необходимо объекта есть специальный класс Creator.

3. Разработка приложения

4. Тестирование

Заключение:

Список использованных источников:

Приложения: