

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

КУРСОВОЙ ПРОЕКТ
ЗАЩИЩЕНА С ОЦЕНКОЙ
РУКОВОДИТЕЛЬ

Ст. преподаватель		Е.О. Шумова
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К КУРСОВОМУ ПРОЕКТУ

по дисциплине: ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ
ПРОГРАММИРОВАНИЕ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТКА ГР.№	4731		А.Д. Ельцова
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2019

Содержание

Задание на курсовое проектирование	3
ВВЕДЕНИЕ	4
1 Постановка задачи	5
1.1 Анализ предметной области	5
1.2 Формулировка технического задания	5
2 Проектирование классов	6
2.1 Используемые паттерны проектирования	7
3 Разработка приложения	9
3.1 Разработка интерфейса приложения	9
3.2 Реализация классов	13
4 Тестирование	19
ЗАКЛЮЧЕНИЕ	31
Список использованных источников	32

Задание на курсовое проектирование

В ходе курсового проекта необходимо разработать приложение, позволяющее протестировать взаимодействие объектов классов, спроектированных и реализованных студентом для решения конкретной задачи при заданных критериях.

При разработке программного обеспечения следует использовать шаблоны проектирования. Также предпочтение должно быть отдано графическому приложению.

При этом подлежат разработке следующие вопросы:

- а) анализ задания;
- б) разработка классов
- в) разработка тестового приложения;
- г) оформление пояснительной записки по результатам выполнения проекта
- д) создание презентации к проекту

Вариант задания на курсовое проектирование должен быть согласован с преподавателем.

Каждый из вариантов определяет предметную область, для моделирования которой должна быть разработана система классов (не менее 10 классов).

ВВЕДЕНИЕ

Разработка системы классов в рамках курсового проектирования является актуальным заданием в учебных целях, так как помогает развить навыки проектирования, закрепить знания по использованию паттернов и реализации их на практике при помощи объектно-ориентированного языка программирования; а также может послужить началом для более детальной разработки системы по предметной области в профессиональном будущем студента. Цели, поставленные на данный курсовой проект, также подтверждают актуальность данной работы в качестве подготовки к написанию ВКР.

Основные задачи также перекликаются с целями проекта, они призваны развить и закрепить навыки проектирования при разработке классов, отточить программирование при разработке программы и потренировать себя в ораторском искусстве при подготовке пояснительной записки и презентации.

При выполнении данного проекта были достигнуты все его основные цели. На каждом этапе, при этом, возникали разные трудности. Основной сложностью при проектировании классов стал правильный выбор шаблонов проектирования, от которого зависела будущая структура классов и непосредственная работа приложения. При непосредственной разработке приложения основной сложностью стал выбор языка программирования таким образом, чтобы работа была актуальной, а также имелись нужные инструменты для создания графического интерфейса.

1 Постановка задачи

1.1 Анализ предметной области

Предметной областью данного проекта является работа поликлиники. В данной предметной области можно выделить двух основных актеров – пациент и врач. Кроме того, в своей работе я предполагаю наличие регистратуры, которая независимо от врача может назначать пациенту прием в соответствии с графиком работы, а также регистрировать новых пациентов и врачей. Предполагается, что основными пользователями системы будут именно работники регистратуры.

Спецификой предметной области является работа с многочисленными документами, такими как паспорт, СНИЛС, медицинский полис. Для их учета и внесения в систему предусмотрены соответствующие поля, с проверкой ввода номеров данных документов.

1.2 Формулировка технического задания

Разработать систему для обеспечения работы поликлиники.

В системе должны быть реализованы следующие функции:

- Добавление и поиск пациентов
- Добавление и поиск врачей
- Добавление приема к врачу, а также просмотр всех доступных записей
- Анализ статистики заболеваемости по месяцам и районам

В системы должен быть реализован графический интерфейс.

Система должна поддерживать фильтрацию ввода для основных полей.

2 Проектирование классов

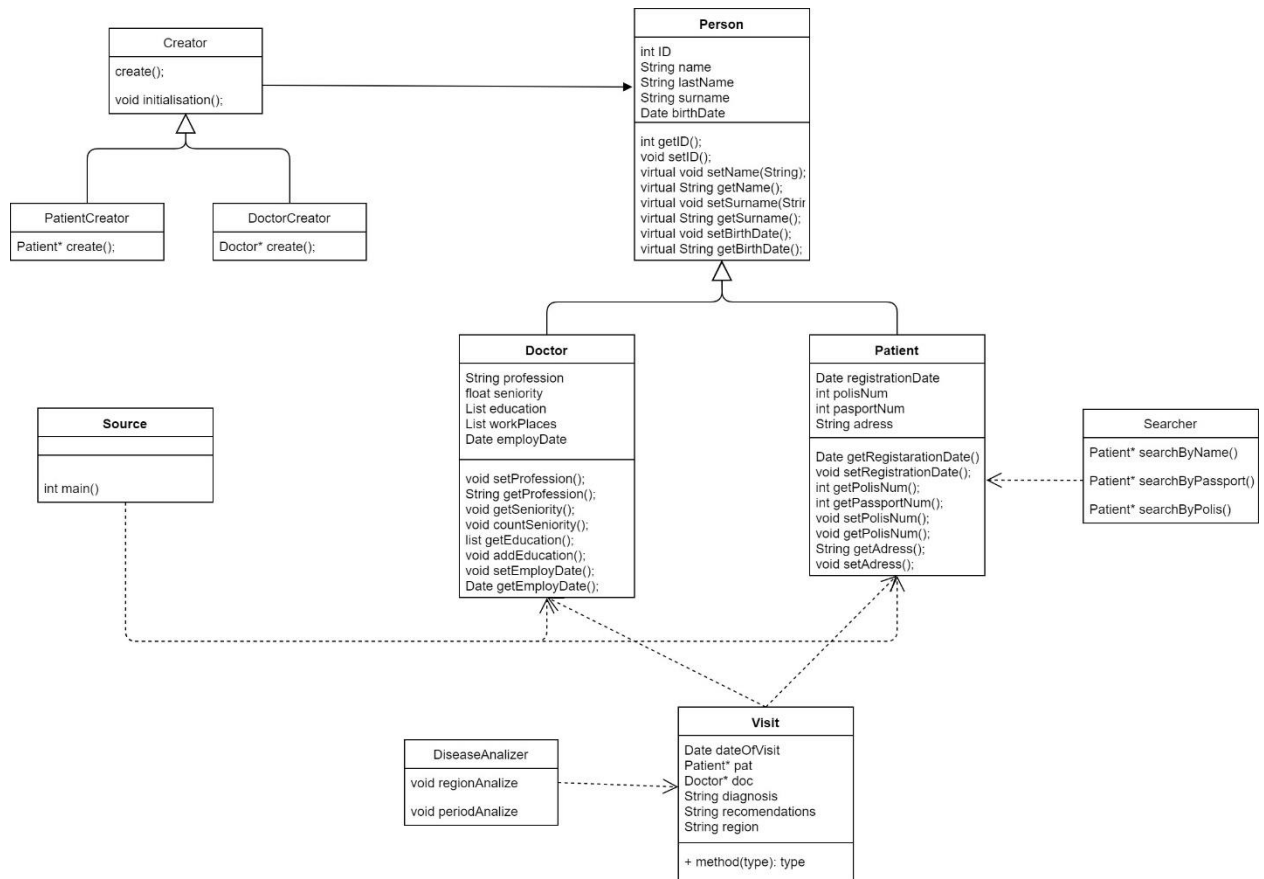


Рисунок 1

Иерархия классов на рис.1 была получена во время этапа проектирования и в последствии была дополнена до вида, изображенного на рис.2.

Классы Creator, PatientCreator и DoctorCreator реализуют шаблон проектирования фабричный метод для классов Doctor и Patient. Классы Doctor и Patient наследуются от класса Person и вместе с Visit являются классами сущностей. Классы Searcher и Stats служат для выполнения поиска и формирования статистики соответственно.

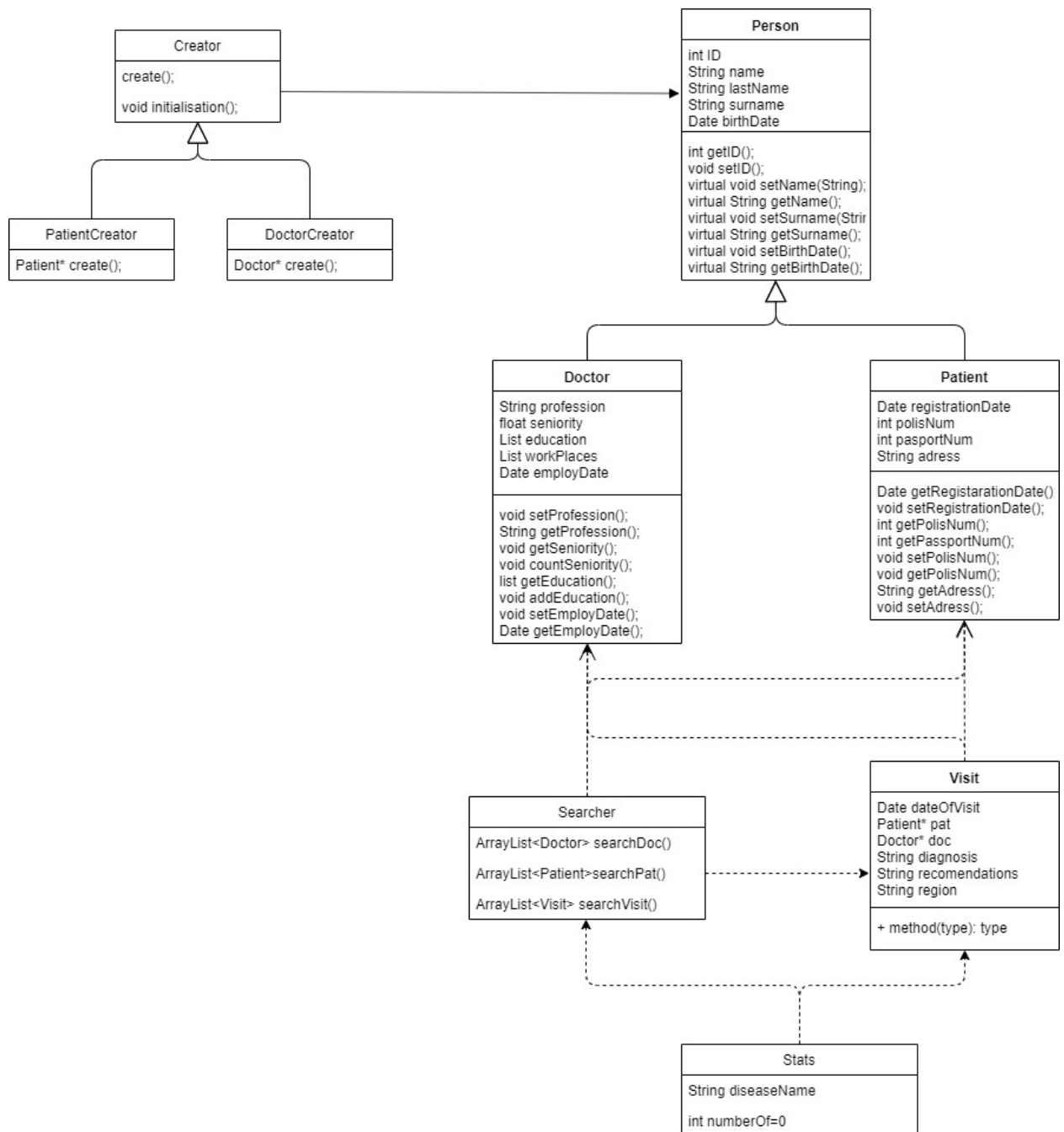


Рисунок 2

2.1 Используемые паттерны проектирования

В данной работе реализуется такой шаблон проектирования как фабричный метод. Его общая структура представлена на рис.3. В качестве класса Product в программе реализован класс Person, наследниками которого являются Doctor и Patient. В шаблоне они представляют классы вида ConcreteProduct. Класс Creator остается без изменений, а ConcreteCreator будет соответственно разделяться на DoctorCreator и PatientCreator.

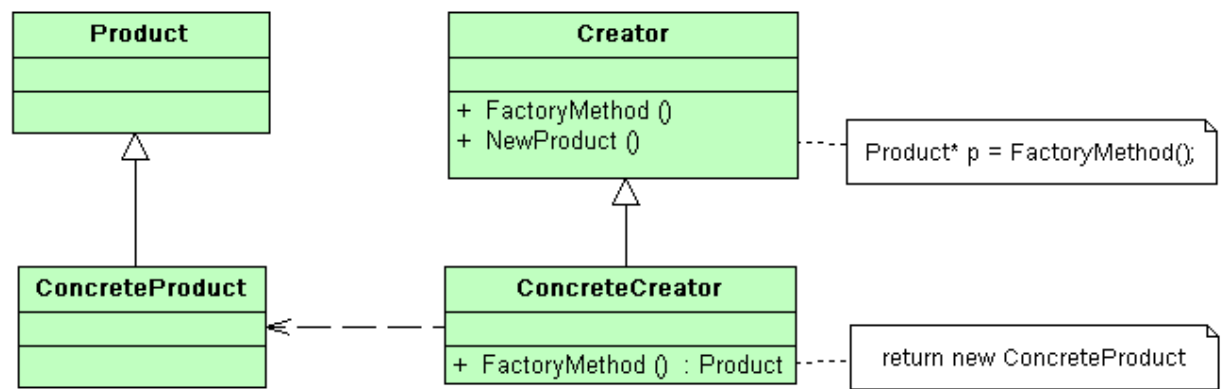


Рисунок 3

3 Разработка приложения

3.1 Разработка интерфейса приложения

- Главное меню (MainMenu)

MainMenu

Быстрый доступ

Новая запись к врачу Все записи к врачам Помощь

Пациенты

Добавить пациента

по фамилии Поиск

Врачи

Добавить врача Поиск

Статистика

по месяцам Январь Показать

Рисунок 4

Является основной формой для взаимодействия с приложением. Разделена на 4 логических части: Быстрый доступ, Пациенты, Врачи, Статистика. Каждый раздел обозначен соответствующей надписью. В разделе «Быстрый доступ» располагаются кнопки для создания новой записи к врачу (ведет на форму «Новая запись»), просмотра всех записей к врачам (ведет на форму «Все записи»), а также кнопка «Помощь». В разделах «Пациенты» и «Врачи» представлен схожий набор элементов управления для добавления пациента (врача) и поиска по заданному критерию. Кнопки добавления ведут соответственно на формы «Новый пациент (врач)», а поиск в обоих случаях ведет на «Табличную форму». Раздел статистика посвящен анализу заболеваемости. Для того, чтобы воспользоваться им нужно выбрать в левом выпадающем списке тип анализа (по месяцам или по районам) и далее выбрать в правом списке нужный месяц (район), после чего нажать кнопку «Показать», ведущую на форму «Статистика».

- Форма «Новая запись» (NewVisit)

Дата обращения

Пациент

Врач

Диагноз

Рекомендации

Район: Адмиралтейский

OK Cancel

Рисунок 5

Форма содержит поля для заполнения данных о новой записи к врачу. Если поля корректны, то после нажатия кнопки ОК, запись к врачу добавляется в список доступных записей и окно закрывается. При нажатии Cancel сохранения не происходит и окно просто закрывается.

- Форма «Все записи» (AllVisits)

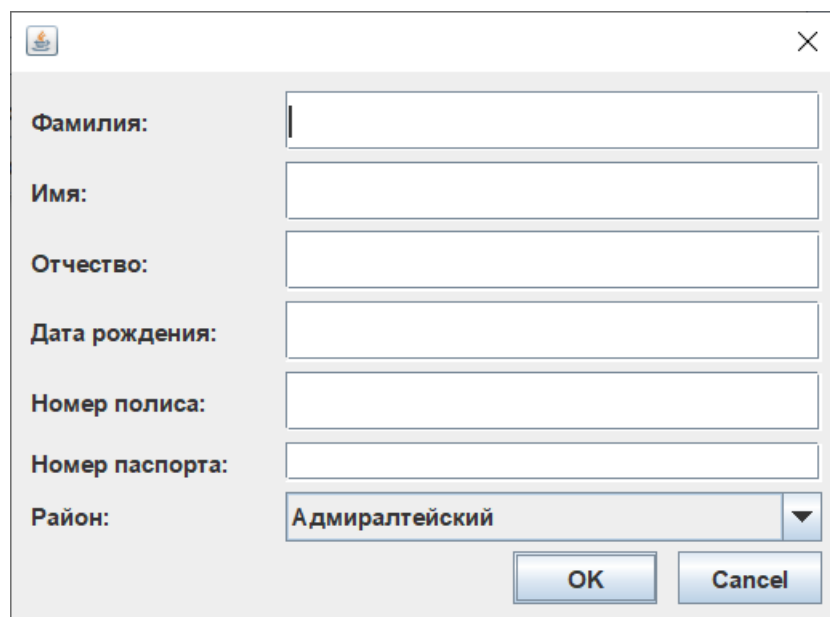
Дата	Врач	Пациент	Диагноз	Район
12.12.2019	Петров Иван	Иванов Кирилл	Корь	
25.12.2018	Петров	Иванов	ОРЗ	Кировский

OK Cancel

Рисунок 6

Отображает все занесенные в список записей записи к врачу в виде таблицы.

- Форма «Новый пациент»(NewPat)



Фамилия:

Имя:

Отчество:

Дата рождения:

Номер полиса:

Номер паспорта:

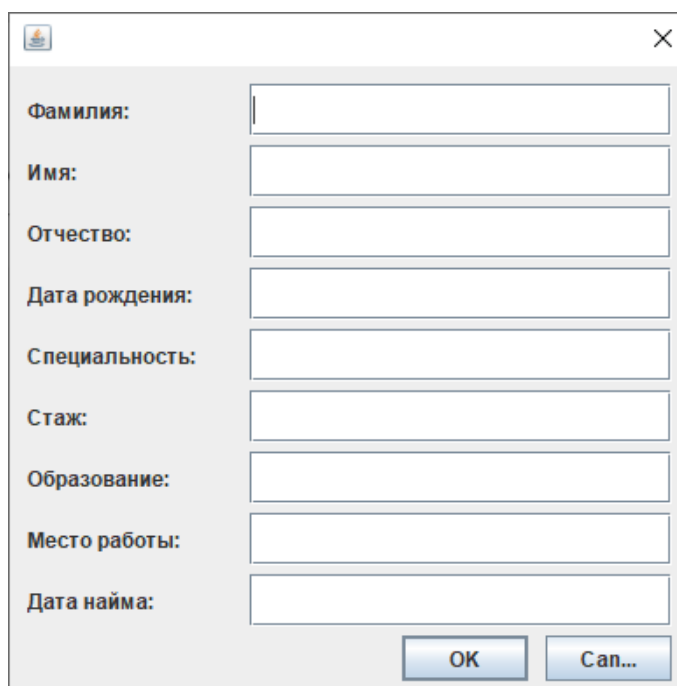
Район: Адмиралтейский

OK Cancel

Рисунок 7

Форма содержит поля для заполнения данных о новом пациенте. Поля «Дата рождения», «Номер полиса» и «Номер паспорта» обладают фильтрацией ввода. Если поля корректны, то после нажатия кнопки ОК, запись к врачу добавляется в список доступных записей и окно закрывается. При нажатии Cancel сохранения не происходит и окно закрывается.

- Форма «Новый врач»(NewDoc)



Фамилия:

Имя:

Отчество:

Дата рождения:

Специальность:

Стаж:

Образование:

Место работы:

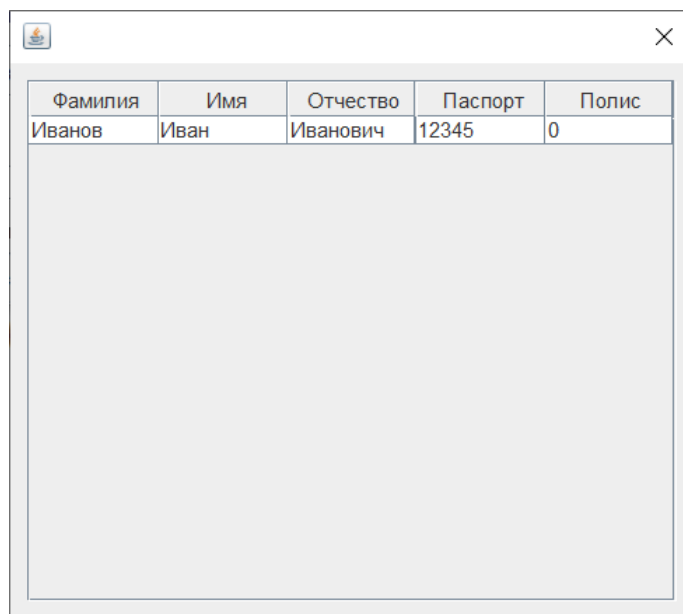
Дата найма:

OK Can...

Рисунок 8

Форма содержит поля для заполнения данных о новом враче. Поля «Дата рождения» и «Дата найма» обладают фильтрацией ввода. Если поля корректны, то после нажатия кнопки ОК, запись к врачу добавляется в список доступных записей и окно закрывается. При нажатии Cancel сохранения не происходит и окно закрывается.

- «Табличная форма»(TableForm)

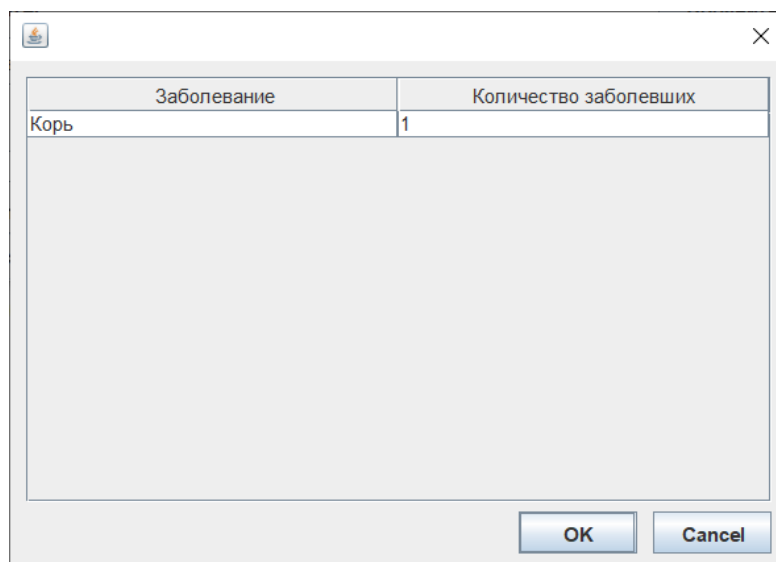


Фамилия	Имя	Отчество	Паспорт	Полис
Иванов	Иван	Иванович	12345	0

Рисунок 9

В зависимости от места вызова в форме показываются данные о найденных врачах или пациентах.

- Форма «Статистика»



Заболевание	Количество заболевших
Корь	1

OK Cancel

Рисунок 10

В форме отображается статистика заболеваний в зависимости от выбранного типа отображения.

3.2 Реализация классов

В данном разделе не будут описываться классы графического интерфейса кроме MainMenu так как все их основные функции были описаны выше.

- MainMenu

Используется для создания главного меню и в качестве хранителя основного класса программы

- Свойства класса:

- Свойства - управляющие элементы формы

```
private JPanel rootPanel;  
private JButton addVisitButtonQ;  
private JButton addPatButton;  
private JTextField patInfoTextField;  
private JComboBox patSearchComboBox;  
private JButton searchPatButton;  
private JButton addDocButton;  
private JTextField docInfoTextField;  
private JComboBox docSearchComboBox;  
private JButton searchDocButton;  
private JButton statsSearch;  
private JComboBox statsTypeComboBox;  
private JButton helpButton;
```

Свойства – списки для хранения данных

```
static ArrayList<Patient> patients = new ArrayList<>();  
static ArrayList<Doctor> doctors = new ArrayList<>();  
static ArrayList<Visit> visits = new ArrayList<>();
```

- Свойства, реализующие фабричный метод

```
PatientCreator patFactory = new PatientCreator();  
DoctorCreator docFactory = new DoctorCreator();
```

- Методы класса:

- Конструктор класса public MainMenu()

Основное содержание метода – слушатели оконных событий, нужные для реакции на события элементов управления на форме. Также содержит команды добавления пунктов в выпадающий список. Назначение – инициализация класса при его создании

- Метод public static void main(String[] args)

Основной класс программы. Создает форму главного меню и заполняет списки данных значениями для тестирования.

- Creator

Абстрактный класс, реализующий шаблон проектирования фабричный метод. Содержит единственный абстрактный метод `Person create()`;

- DoctorCreator

Реализация абстрактного класса `Creator` – единственный метод - `public Doctor create()`, возвращающий ссылку на объект типа `Doctor`

- PatientCreator

Реализация абстрактного класса `Creator` – единственный метод - `public Patient create()`, возвращающий ссылку на объект типа `Patient`

- Person

Класс, являющийся родительским для сущностей `Врач` и `Пациент`.

- Свойства класса:

```
protected int ID;  
protected String name;  
protected String lastName;  
protected String surname;  
protected Date birthDate;
```

- Методы класса

- Сеттеры

```
public void setID(int ID) { this.ID = ID;}  
public void setName(String name) { this.name = name;}  
public void setLastName(String lastName) { this.lastName = lastName;}  
public void setSurname(String surname) { this.surname = surname;}  
public void setBirthDate(Date birthDate) { this.birthDate = birthDate;}
```

- Геттеры

```
public int getID() { return ID;}  
public String getName() { return name;}  
public String getLastName() { return lastName;}  
public String getSurname() { return surname;}  
public Date getBirthDate() { return birthDate;}
```

- Doctor

Один из классов сущностей, советуемый актеру Врач.

- Свойства класса:

```
private String profession;  
private float seniority;  
private String education;  
private String workPlace;  
private Date employDate;
```

- Методы класса

- Сеттеры

```
public void setProfession(String profession) { this.profession = profession;}  
public void setSeniority(float seniority) { this.seniority = seniority;}  
public void setEducation(String education) { this.education = education;}  
public void setWorkPlace(String workPlace) { this.workPlace = workPlace;}  
public void setEmployDate(Date employDate) { this.employDate = employDate;}
```

- Геттеры

```
public String getProfession() { return profession;}  
public float getSeniority() { return seniority;}  
public String getEducation() { return education;}  
public String getWorkPlace() { return workPlace;}  
public Date getEmployDate() { return employDate;}
```

- Patient

Один из классов сущностей, советуемый актеру Пациент.

- Свойства класса:

```
private Date registrationDate;  
private int polisNum;  
private int passportNum;  
private String address;
```

- Методы класса

- Сеттеры

```
public void setID(int ID) { this.ID = ID;}  
public void setName(String name) { this.name = name;}  
public void setLastName(String lastName) { this.lastName = lastName;}
```

```
public void setSurname(String surname) { this.surname = surname;}  
public void setBirthDate(Date birthDate) { this.birthDate = birthDate;}
```

- Геттеры

```
public int getID() { return ID;}  
public String getName() { return name;}  
public String getLastName() { return lastName;}  
public String getSurname() { return surname;}  
public Date getBirthDate() { return birthDate;}
```

- Searcher

Реализует в зависимости от метода реализует поиск в списках данных.

- Методы класса

- Поиск среди врачей

```
ArrayList<Doctor> searchDoc(ArrayList<Doctor> docs, int typeOfSearch, String  
searchStr){
```

Осуществляет поиск строки searchStr в зависимости от typeOfSearch по различным полям объектов типа Doctor в списке docs. Возвращает список найденных врачей

- Поиск среди пациентов

```
ArrayList<Patient> searchPat(ArrayList<Patient> pats, int typeOfSearch, String  
searchStr)
```

Осуществляет поиск строки searchStr в зависимости от typeOfSearch по различным полям объектов типа Patient в списке pats. Возвращает список найденных пациентов

- Поиск среди записей

```
Map<String, ArrayList<Stats>> searchVisit(ArrayList<Visit> visits, int typeOfSearch)
```

Метод применяется для формирования статистики.

Алгоритмом осуществляется поиск всех уникальных болезней, а далее, в зависимости от типа typeOfSearch проводится подсчет количества записей с данным диагнозом в каждом месяце или районе. Данные представляются в виде структуры Map, где ключом список типа Stats, в котором хранится название месяца/района и соответствующее ему количество заболеваний.

- Visit

Один из классов сущностей, советующий визиту пациента к врачу.

- Свойства класса

```
private Date dateOfVisit;  
private String pat;  
private String doc;  
private String diagnosis;  
private String recommends;  
private String region;
```

- Методы класса

- Сеттеры

```
public void setDateOfVisit(String dateOfVisit) {  
    SimpleDateFormat formatter = new SimpleDateFormat("dd.MM.yyyy");  
    try {  
        this.dateOfVisit = formatter.parse(dateOfVisit);  
    } catch (ParseException ex) {  
        ex.printStackTrace();  
    }  
}
```

```
public void setPat(String pat) { this.pat = pat;}
```

```
public void setDoc(String doc) { this.doc = doc;}
```

```
public void setDiagnosis(String diagnosis) { this.diagnosis = diagnosis;}
```

```
public void setRecommends(String recommends) { this.recommends =  
recommends;}
```

```
public void setRegion(String region) { this.region = region;}
```

- Геттеры

```
public Date getDateOfVisit() { return dateOfVisit;}
```

```
public String getPat() { return pat;}
```

```
public String getDoc() { return doc;}
```

```
public String getDiagnosis() { return diagnosis;}
```

```
public String getRecommends() { return recommends;}
```

```
public String getRegion() { return region;}
```

- Stats

Класс для формирования списка статистики в методе searchVisit класса Searcher

- Свойства класса

```
String diseaseName;  
int numberOf=0;
```

- Методы класса

- Конструктор

```
public Stats(String diseaseName){ this.diseaseName=diseaseName;}
```

- Геттеры

```
public int getNumberOf() { return numberOf;}
```

```
public String getDiseaseName() { return diseaseName;}
```

- Другие

```
public void setDiseaseName(String diseaseName) { this.diseaseName =  
diseaseName;}
```

```
public void increaseNumberOf() {  
    this.numberOf++;}
```

4 Тестирование

1. Поиск пациента по фамилии

Поиск тестового пациента по фамилии Иванов

The screenshot shows a window titled "MainMenu" with standard Windows window controls (minimize, maximize, close). The interface is organized into several sections:

- Быстрый доступ (Quick Access):** Contains three buttons: "Новая запись к врачу" (New appointment with a doctor), "Все записи к врачам" (All appointments with doctors), and "Помощь" (Help).
- Пациенты (Patients):** Includes a text input field containing "Иванов", a dropdown menu set to "по фамилии" (by last name), and a "Поиск" (Search) button. There is also a "Добавить пациента" (Add patient) button.
- Врачи (Doctors):** Includes an empty text input field, a dropdown menu set to "по фамилии" (by last name), and a "Поиск" (Search) button. There is also a "Добавить врача" (Add doctor) button.
- Статистика (Statistics):** Includes a dropdown menu set to "по месяцам" (by month), a text input field containing "Январь" (January), and a "Показать" (Show) button.

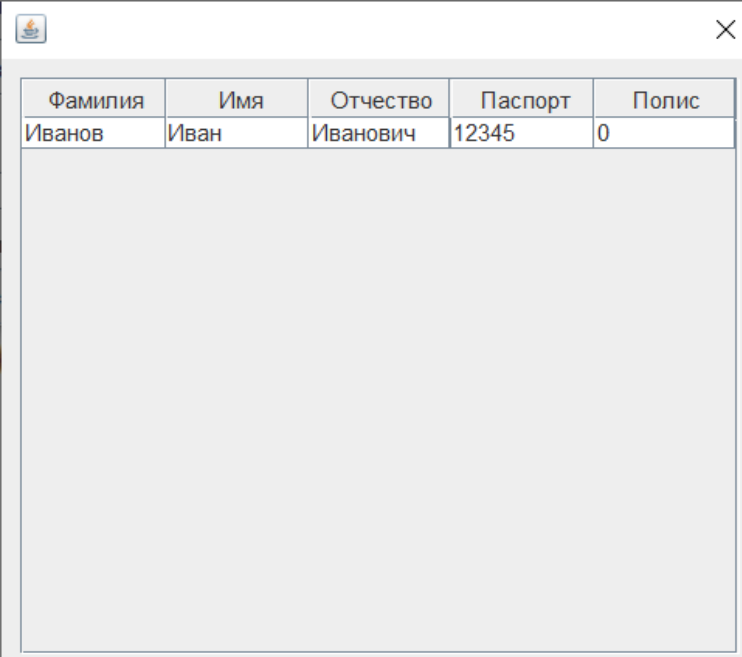
После нажатия «поиск»

The screenshot shows a window displaying the search results in a table format. The table has five columns: "Фамилия" (Surname), "Имя" (Name), "Отчество" (Patronymic), "Паспорт" (Passport), and "Подпис" (Signature). The first row contains the data for the test patient: "Иванов", "Иван", "Иванович", "12345", and "0". Below the table is a large empty rectangular area, likely intended for displaying additional information or a list of results.

Фамилия	Имя	Отчество	Паспорт	Подпис
Иванов	Иван	Иванович	12345	0

2. Добавление нового пациента

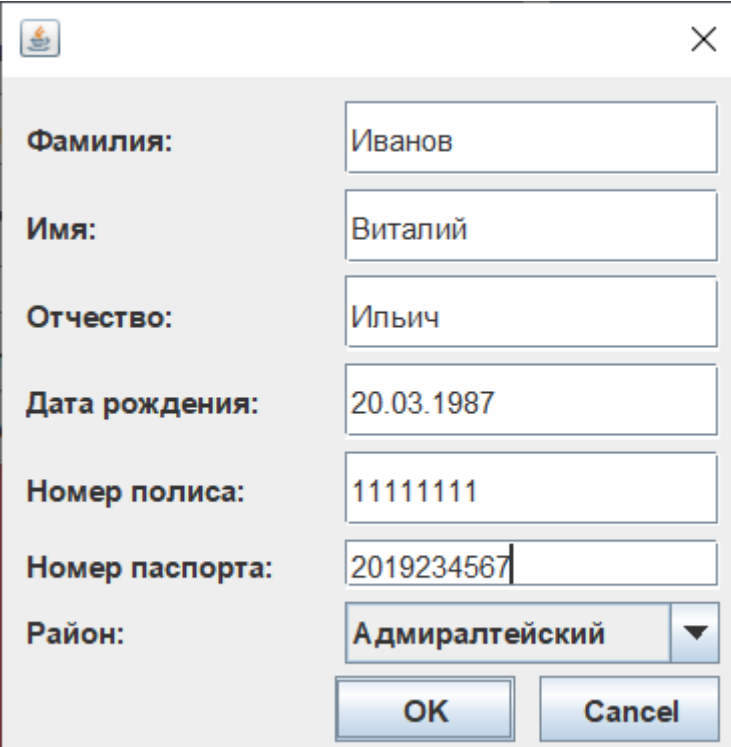
Поиск по «Иванов» до добавления



A screenshot of a search window. At the top, there is a table with five columns: 'Фамилия' (Surname), 'Имя' (Name), 'Отчество' (Patronymic), 'Паспорт' (Passport), and 'Полис' (Policy). The first row contains the values 'Иванов', 'Иван', 'Иванович', '12345', and '0'. Below the table is a large, empty rectangular area.

Фамилия	Имя	Отчество	Паспорт	Полис
Иванов	Иван	Иванович	12345	0

Заполнение формы добавления



A screenshot of a patient addition form. It contains several labeled input fields: 'Фамилия:' (Ivanov), 'Имя:' (Vitaliy), 'Отчество:' (Ilyich), 'Дата рождения:' (20.03.1987), 'Номер полиса:' (11111111), 'Номер паспорта:' (2019234567), and 'Район:' (Адмиралтейский). At the bottom, there are 'OK' and 'Cancel' buttons.

Фамилия: Иванов

Имя: Виталий

Отчество: Ильич

Дата рождения: 20.03.1987

Номер полиса: 11111111

Номер паспорта: 2019234567

Район: Адмиралтейский

OK Cancel

Поиск по Иванов после добавления

Фамилия	Имя	Отчество	Паспорт	Полис
Иванов	Иван	Иванович	12345	0
Иванов	Виталий	Ильич	2019234567	1111111

3. Поиск по номеру паспорта

Ввод данных

MainMenu

Быстрый доступ

Новая запись к врачу

Все записи к врачам

Помощь

Пациенты

2019234567

по паспорту

Добавить пациента

Поиск

Врачи

по фамилии

Добавить врача

Поиск

Статистика

по месяцам

Январь

Показать

Результат поиска

Фамилия	Имя	Отчество	Паспорт	Полис
Иванов	Виталий	Ильич	2019234567	1111111

4. Поиск по номеру полиса

Ввод данных

MainMenu

Быстрый доступ

Новая запись к врачу

Все записи к врачам

Помощь

Пациенты

1111111

по полису

Поиск

Врачи

по фамилии

Поиск

Статистика

по месяцам

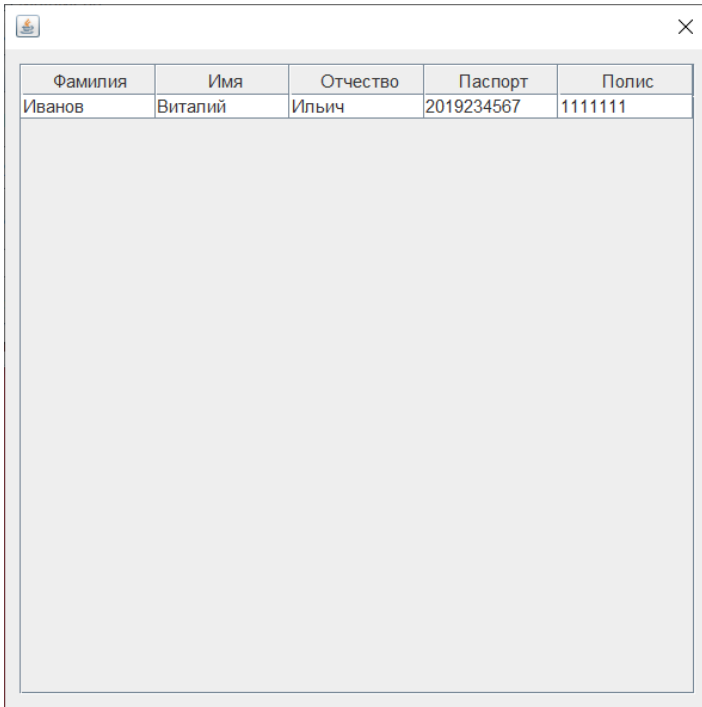
Январь

Показать

Добавить пациента

Добавить врача

Результат поиска

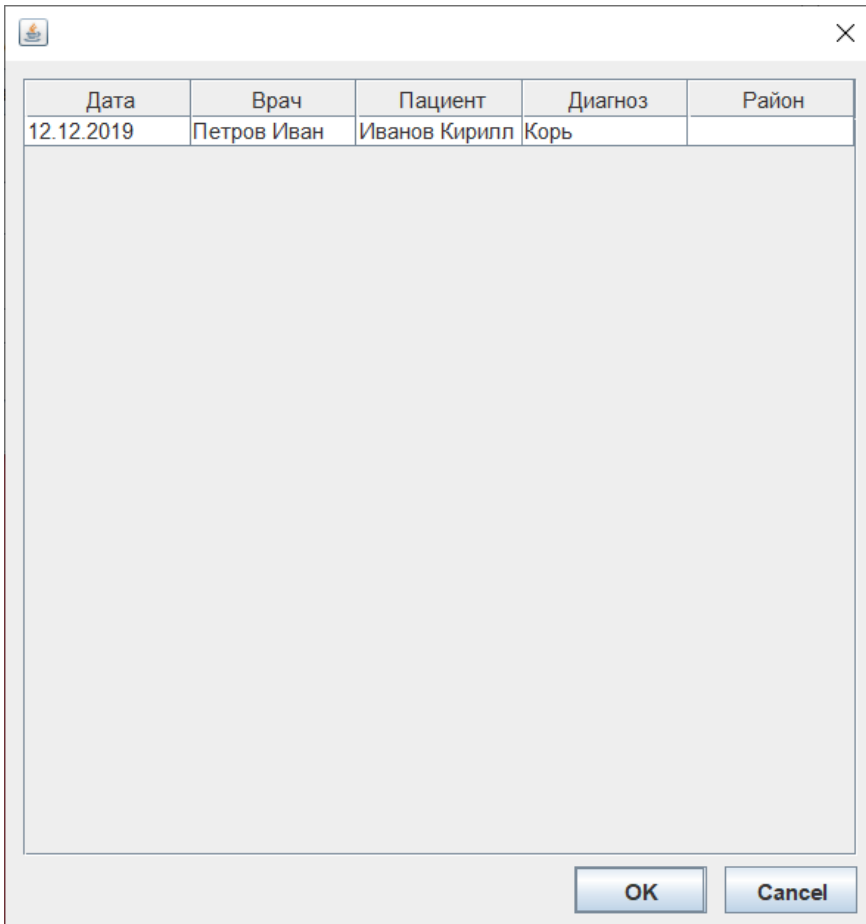


A screenshot of a software window with a close button (X) in the top right corner. The window contains a table with five columns: Фамилия, Имя, Отчество, Паспорт, and Полис. The first row of data contains the values: Иванов, Виталий, Ильич, 2019234567, and 1111111. Below the table is a large, empty rectangular area.

Фамилия	Имя	Отчество	Паспорт	Полис
Иванов	Виталий	Ильич	2019234567	1111111

5. Добавление новой записи к врачу

Все записи до добавления



A screenshot of a software window with a close button (X) in the top right corner. The window contains a table with five columns: Дата, Врач, Пациент, Диагноз, and Район. The first row of data contains the values: 12.12.2019, Петров Иван, Иванов Кирилл, Корь, and an empty cell. Below the table is a large, empty rectangular area. At the bottom right of the window are two buttons: OK and Cancel.

Дата	Врач	Пациент	Диагноз	Район
12.12.2019	Петров Иван	Иванов Кирилл	Корь	

Добавление новой записи

×

Дата обращения

20.12.2019

Пациент

Борисов Денис

Врач

Краснов Даниил

Диагноз

ОРЗ

Рекомендации

Постельный режим

Район

Московский

▼

OK

Cancel

Все записи после добавления

×

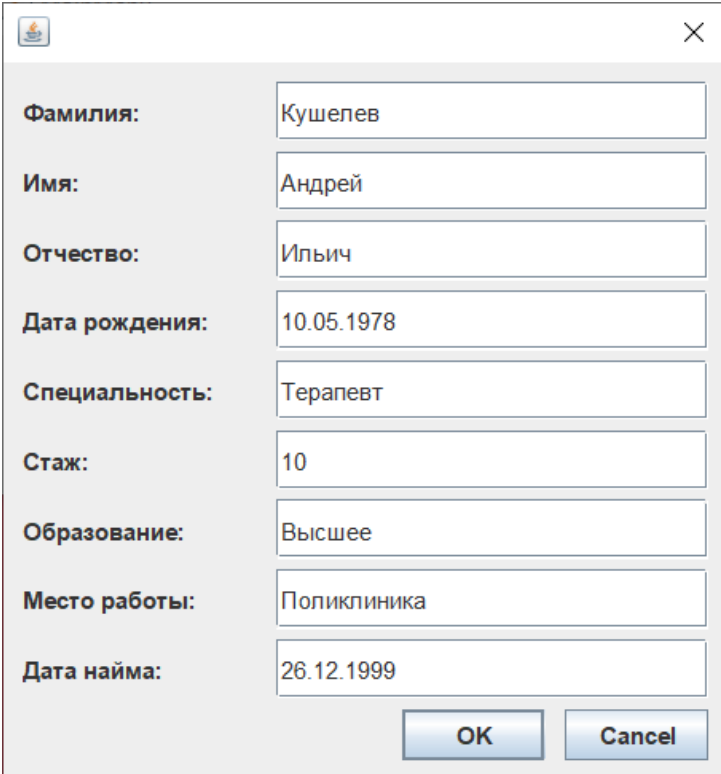
Дата	Врач	Пациент	Диагноз	Район
12.12.2019	Петров Иван	Иванов Кирилл	Корь	
20.12.2019	Краснов Дани...	Борисов Денис	ОРЗ	Московский

OK

Cancel

6. Добавление нового врача

Заполнение формы добавления



Фамилия: Кушелев

Имя: Андрей

Отчество: Ильич

Дата рождения: 10.05.1978

Специальность: Терапевт

Стаж: 10

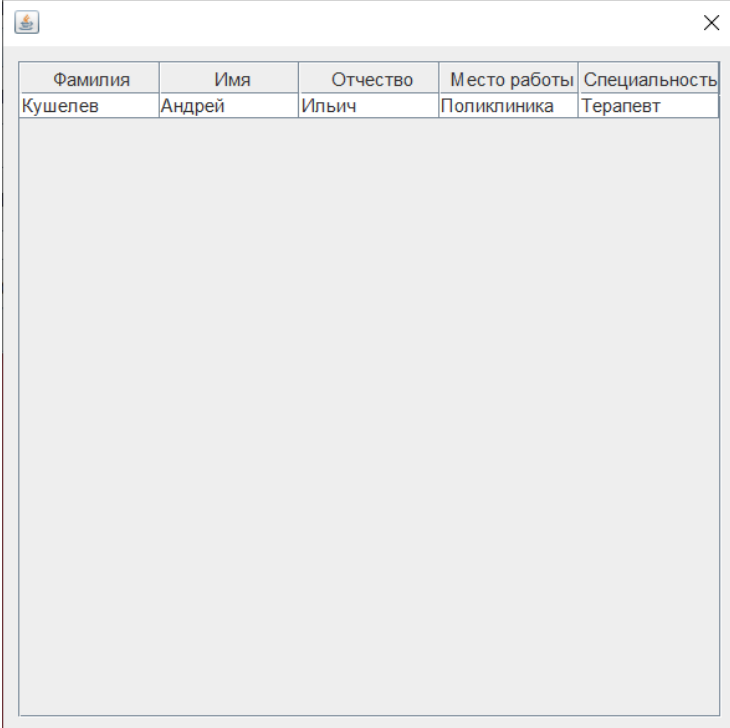
Образование: Высшее

Место работы: Поликлиника

Дата найма: 26.12.1999

OK Cancel

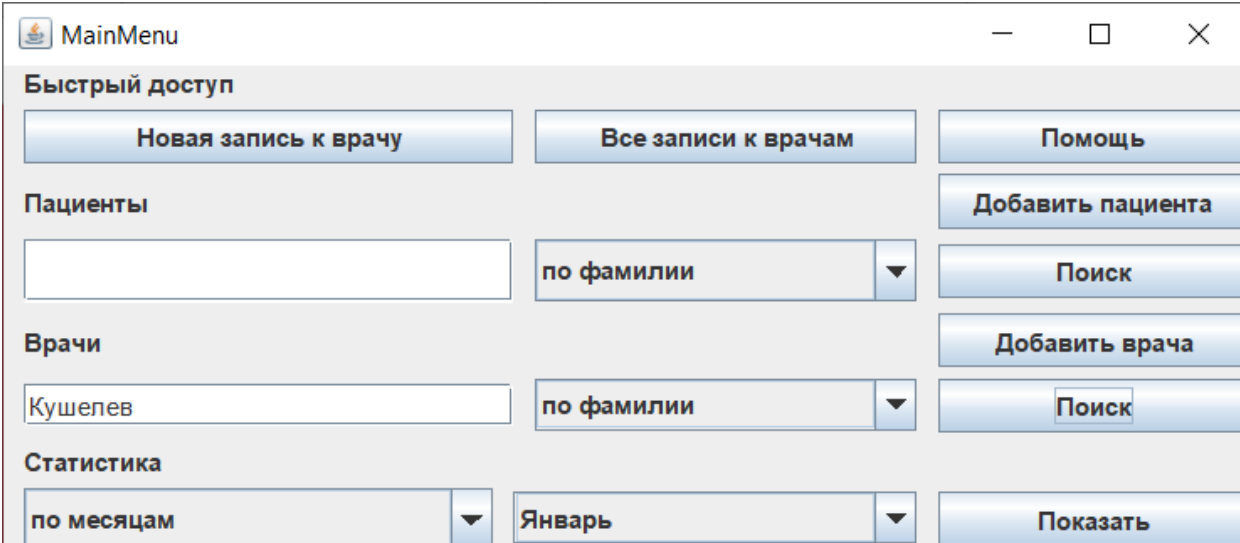
Поиск по «Кушелев»



Фамилия	Имя	Отчество	Место работы	Специальность
Кушелев	Андрей	Ильич	Поликлиника	Терапевт

7. Поиск врача по фамилии

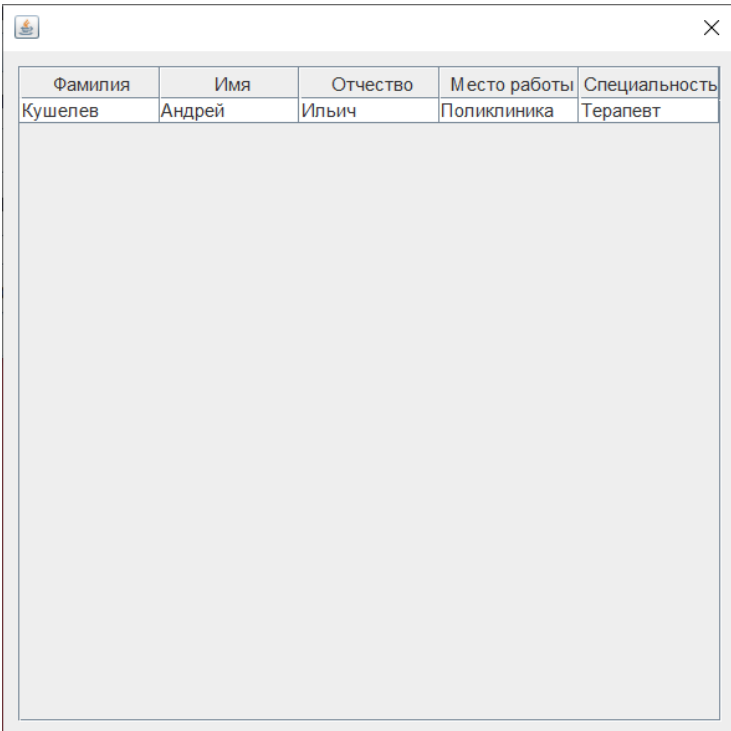
Заполнение формы поиска



The screenshot shows a window titled "MainMenu" with standard Windows window controls (minimize, maximize, close). The window contains several sections with buttons and input fields:

- Быстрый доступ** (Quick Access): Three buttons: "Новая запись к врачу" (New appointment with a doctor), "Все записи к врачам" (All appointments with doctors), and "Помощь" (Help).
- Пациенты** (Patients): A text input field, a dropdown menu set to "по фамилии" (by surname), and a "Поиск" (Search) button. A "Добавить пациента" (Add patient) button is also present.
- Врачи** (Doctors): A text input field containing "Кушелев", a dropdown menu set to "по фамилии" (by surname), and a "Поиск" (Search) button. A "Добавить врача" (Add doctor) button is also present.
- Статистика** (Statistics): A dropdown menu set to "по месяцам" (by month), a text input field containing "Январь", and a "Показать" (Show) button.

Результаты поиска



The screenshot shows a window displaying the search results in a table format. The table has five columns: "Фамилия" (Surname), "Имя" (Name), "Отчество" (Patronymic), "Место работы" (Place of work), and "Специальность" (Specialty). The first row contains the following data:

Фамилия	Имя	Отчество	Место работы	Специальность
Кушелев	Андрей	Ильич	Поликлиника	Терапевт

8. Поиск врача по специальности

Заполнение формы поиска

The screenshot shows a window titled "MainMenu" with standard Windows window controls (minimize, maximize, close). The window is divided into several sections:

- Быстрый доступ (Quick Access):** Contains three buttons: "Новая запись к врачу" (New appointment with a doctor), "Все записи к врачам" (All appointments with doctors), and "Помощь" (Help).
- Пациенты (Patients):** Includes a text input field, a dropdown menu currently showing "по фамилии" (by surname), and a "Поиск" (Search) button. There is also a "Добавить пациента" (Add patient) button.
- Врачи (Doctors):** Includes a text input field containing "Терапевт" (Therapist), a dropdown menu currently showing "по специальности" (by specialty), and a "Поиск" (Search) button. There is also a "Добавить врача" (Add doctor) button.
- Статистика (Statistics):** Includes a dropdown menu currently showing "по месяцам" (by month), a text input field containing "Январь" (January), and a "Показать" (Show) button.

Результаты поиска

The screenshot shows a window displaying search results in a table format. The table has five columns: "Фамилия" (Surname), "Имя" (Name), "Отчество" (Patronymic), "Место работы" (Place of work), and "Специальность" (Specialty). One result is shown:

Фамилия	Имя	Отчество	Место работы	Специальность
Кушелев	Андрей	Ильич	Поликлиника	Терапевт

9. Поиск врача по месту работы

Заполнение формы поиска

The screenshot shows the 'MainMenu' window with the following sections and controls:

- Быстрый доступ** (Quick Access):
 - Buttons: 'Новая запись к врачу' (New appointment with a doctor), 'Все записи к врачам' (All appointments with doctors), 'Помощь' (Help).
- Пациенты** (Patients):
 - Text input field.
 - Dropdown menu: 'по фамилии' (by surname).
 - Buttons: 'Добавить пациента' (Add patient), 'Поиск' (Search).
- Врачи** (Doctors):
 - Text input field containing 'Поликлиника' (Polyclinic).
 - Dropdown menu: 'по месту работы' (by workplace).
 - Buttons: 'Добавить врача' (Add doctor), 'Поиск' (Search).
- Статистика** (Statistics):
 - Dropdown menu: 'по месяцам' (by month).
 - Text input field containing 'Январь' (January).
 - Button: 'Показать' (Show).

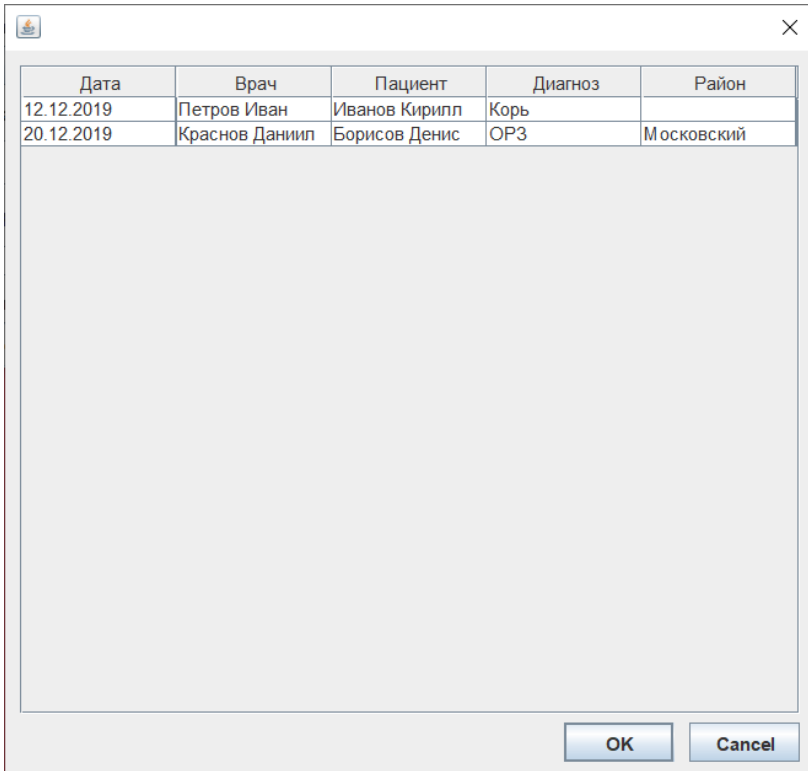
Результаты поиска

The screenshot shows a window with a table of search results. The table has the following data:

Фамилия	Имя	Отчество	Место работы	Специальность
Кушелев	Андрей	Ильич	Поликлиника	Терапевт

10. Проверка статистики по месяцам

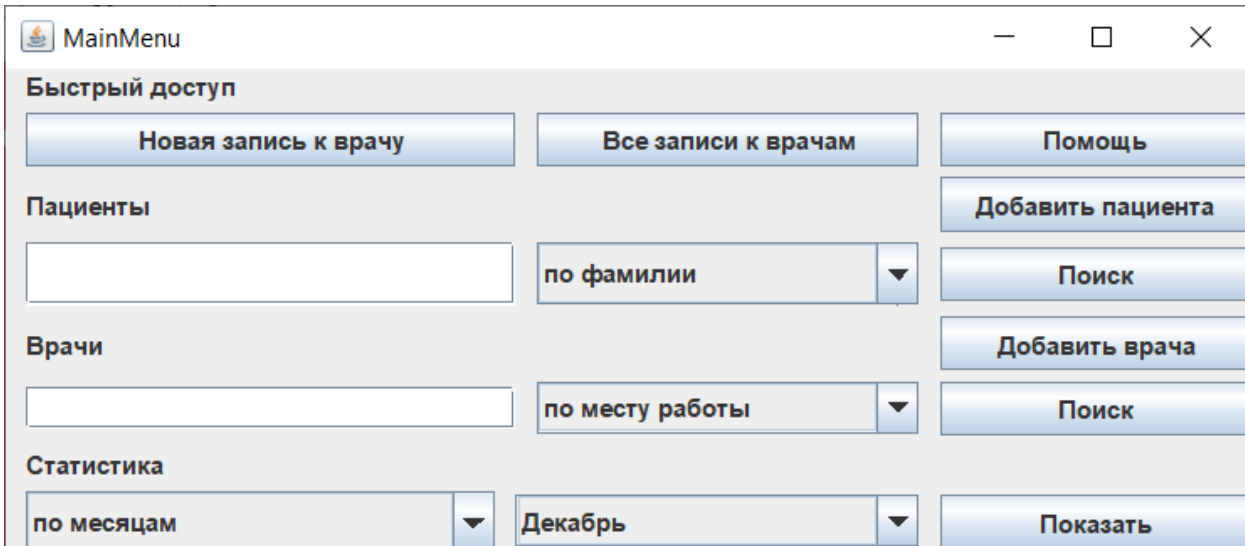
Все записи к врачам



A screenshot of a software window displaying a table of medical records. The table has five columns: Дата, Врач, Пациент, Диагноз, and Район. It contains two rows of data. Below the table is a large empty rectangular area. At the bottom right of the window are 'OK' and 'Cancel' buttons.

Дата	Врач	Пациент	Диагноз	Район
12.12.2019	Петров Иван	Иванов Кирилл	Корь	
20.12.2019	Краснов Даниил	Борисов Денис	ОРЗ	Московский

Выбор типа сбора статистики



A screenshot of the 'MainMenu' window. It features several sections with buttons and input fields. The 'Быстрый доступ' section has three buttons. The 'Пациенты' section has a text input, a dropdown menu set to 'по фамилии', and a 'Поиск' button. The 'Врачи' section has a text input, a dropdown menu set to 'по месту работы', and a 'Поиск' button. The 'Статистика' section has a dropdown menu set to 'по месяцам', a date field set to 'Декабрь', and a 'Показать' button. There are also buttons for 'Добавить пациента' and 'Добавить врача'.

Быстрый доступ

Новая запись к врачу Все записи к врачам Помощь

Пациенты

Добавить пациента

по фамилии ▼ Поиск

Врачи

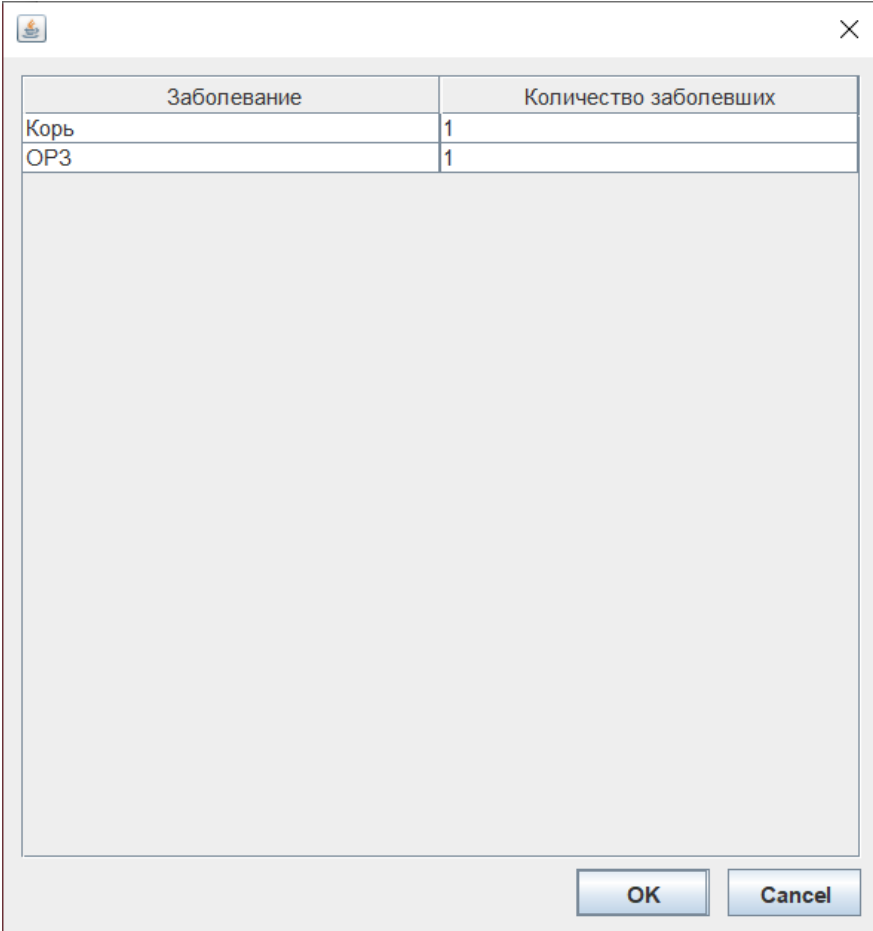
Добавить врача

по месту работы ▼ Поиск

Статистика

по месяцам ▼ Декабрь ▼ Показать

Результат запроса



Заболевание	Количество заболевших
Корь	1
ОРЗ	1

OK Cancel

ЗАКЛЮЧЕНИЕ

Приложение, реализованное, в рамках курсового проектирования, возможно не обладает всеми достоинствами полноценной системы учета в связи с отсутствием целостности данных, простым интерфейсом и небольшим функционалом, однако отвечает поставленным требованиям и в совокупности с данной документацией реализует все цели, поставленные на данный проект. Так, в рамках разработки приложения студентом были закреплены знания, полученные в ходе изучения дисциплины «Объектно-ориентированное программирование», в частности: наследование, использование шаблонов проектирования, виртуальные функции. Кроме того, в ходе выполнения проекта студентом были приобретены навыки практического программирования, выяснены сложности, с которыми придется сталкиваться в будущем. Несомненными преимуществами данной работы, в учебных целях, является наличие графического интерфейса, реализация шаблонов программирования, а также исчерпывающая документация в виде данной пояснительной записки. При этом нужно отметить, что работа определенно требует лучшего тестирования, добавления обработки исключений и возможно подключения базы данных для обеспечения целостности данных.

Список использованных источников

1. Е.О. Шумова ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ - ФГАОУ ВО СПбГУАП, 2019, 14 с
2. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. — Приемы объектно-ориентированного проектирования. Паттерны проектирования – Питер 2016, 366 с
3. Эрик Фримен, Элизабет Фримен, Кэтти Сьерра, Берт Бейтс — Паттерны проектирования – Питер 2018, 656 с
4. Сьерра, Берт Бейтс Изучаем Java – Питер 2е издание 2012 663 с