**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

Кафедра «Электронные вычислительные машины»

**Отчет о лабораторной работе № 5**

по теме

«Интерфейсы»

**Выполнили:**

Студенты гр. 045

Бригады №10

Анохин В.А.

Вашкулатов Н.А.

**Проверил:**

Асс. Бастрычкин А.С.

Цель работы:изучение методов реализации интерфейсов, приобретение навыков использования интерфейсов в Java-программах.

Ход работы

**Пациент (Patient).** Возможности: регистрировать пациента; оформить договор обслуживания; получить персональные данные: id, фамилия, имя, отчество, адрес, телефон, номер медицинской карты; изменить персональные данные; записаться на прием к врачу; записать диагноз/назначенное лечение; сдать лабораторные исследования; получить историю болезни; получить диагноз/назначение лечение; оплатить услуги; отказаться от обслуживания; возобновить обслуживание. Добавить специализированные методы для Пациента стационара, Пациента с COVID-19.

Код программы:

Класс PatientInfo

public class PatientInfo {  
 private String name, surname, patronymic;  
 private String address = "Не указан";  
 private String phone = "Не указан";  
 private static int numOfPatient;  
 private final int id;  
 private int medCard;  
  
 PatientInfo(String name, String surname, String patronymic){  
 this.name = name;  
 this.surname = surname;  
 this.patronymic = patronymic;  
 id = numOfPatient;  
 numOfPatient++;  
 }  
  
 PatientInfo(String name, String surname, String patronymic, String address, String phone){  
 this(name, surname, patronymic);  
 this.address = address;  
 this.phone = phone;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getSurname() {  
 return surname;  
 }  
  
 public void setSurname(String surname) {  
 this.surname = surname;  
 }  
  
 public String getPatronymic() {  
 return patronymic;  
 }  
  
 public void setPatronymic(String patronymic) {  
 this.patronymic = patronymic;  
 }  
  
 public String getAddress() {  
 return address;  
 }  
  
 public void setAddress(String address) {  
 this.address = address;  
 }  
  
 public String getPhone() {  
 return phone;  
 }  
  
 public void setPhone(String phone) {  
 this.phone = phone;  
 }  
  
 public int getMedCard() {  
 return medCard;  
 }  
  
 public void setMedCard(int medCard) {  
 this.medCard = medCard;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public static int getNumOfPatient() {  
 return numOfPatient;  
 }  
  
 @Override  
 public String toString() {  
 return "ID пациента: " + id + '\n' +  
 "ФИО: " + name +" "+ surname + " " + patronymic+'\n' +  
 "Номер медицинской карты: " + medCard + '\n' +  
 "Адрес: " + address + '\n' +  
 "Телефон: " + phone+ '\n';  
 }  
}

Интерфейс PatientInfoManager

public interface PatientInfoManager {  
 String getName();  
 void setName(String name);  
 String getSurname();  
 void setSurname(String surname);  
 String getPatronymic();  
 void setPatronymic(String patronymic);  
 String getAddress();  
 void setAddress(String address);  
 String getPhone();  
 void setPhone(String phone);  
 int getMedCard();  
 void setMedCard(int medCard);  
 int getId();  
}

Интерфейс PatientManager

public interface PatientManager extends PatientInfoManager{  
 default void regPatient(int medCard){  
 setMedCard(medCard);  
 }  
 default void issueContract(){  
 System.out.println("Договор оформлен");  
 }  
 default void stopContract() {  
 System.out.println("Отказ от обслуживания выполнен");  
 }  
 default void continueContract() {  
 System.out.println("Обслуживание возобновлено");  
 }  
 void makeAnAppointment();  
 void recordDiagnosis(String diagnosis);  
 void recordTreatment(String treatment);  
 void passTests();  
 String GetDiagnosis();  
 String GetTreatment();  
 String getMedicalHistory();  
 void payServices();  
}

Класс HospitalPatient

public class HospitalPatient extends PatientInfo implements PatientManager {  
 String treatment, diagnosis;

HospitalPatient(String name, String surname, String patronymic)

{  
 super(name, surname, patronymic);  
 }  
  
 HospitalPatient(String name, String surname, String patronymic, String address, String phone) {  
 super(name, surname, patronymic, address, phone);  
 }  
  
 @Override  
 public void makeAnAppointment() {  
 System.out.println("Запись на прием к врачу выполнена");  
 }  
 @Override  
 public void recordDiagnosis(String diagnosis) {  
 this.diagnosis = diagnosis;  
 System.out.println("Диагноз пациента стационара записан");  
 }  
 @Override  
 public void recordTreatment(String treatment) {  
 this.treatment = treatment;  
 System.out.println("Назначенное лечение пациента стационара записано");  
 }  
 @Override  
 public void passTests() {  
 System.out.println("Лабоаторые исследования сданы");  
 }  
 @Override  
 public String GetDiagnosis() {  
 return "Вывод диагноза: "+diagnosis;  
 }  
 @Override  
 public String GetTreatment() {  
 return "Вывод назначенного лечения: "+treatment;  
 }  
 @Override  
 public String getMedicalHistory() {  
 return "Вывод истории болезни пациента стационара";  
 }  
 @Override  
 public void payServices() {  
 System.out.println("Услуги оплачены");  
 }  
}

Класс CovidPatient

public class CovidPatient extends PatientInfo implements PatientManager{  
 private String treatment, diagnosis;  
 private static int covidCounter;

CovidPatient(String name, String surname, String patronymic) {  
 super(name, surname, patronymic);  
 }  
  
 CovidPatient(String name, String surname, String patronymic, String address, String phone) {  
 super(name, surname, patronymic, address, phone);  
 }  
  
 @Override  
 public void regPatient(int medCard){  
 setMedCard(medCard);  
 covidCounter++;  
 }  
 @Override  
 public void makeAnAppointment() {  
 System.out.println("Запись на прием к врачу выполнена");  
 }  
 @Override  
 public void recordDiagnosis(String diagnosis) {  
 this.diagnosis = diagnosis;  
 System.out.println("Диагноз пациента с COVID-19 записан");  
 }  
 @Override  
 public void recordTreatment(String treatment) {  
 this.treatment = treatment;  
 System.out.println("Назначенное лечение пациента с COVID-19 записано");  
 }  
 @Override  
 public void passTests() {  
 System.out.println("Тест на COVID-19 сдан");  
 }  
 @Override  
 public String GetDiagnosis() {  
 return "Вывод диагноза: "+diagnosis;  
 }  
 @Override  
 public String GetTreatment() {  
 return "Вывод назначенного лечения: "+treatment;  
 }  
 @Override  
 public String getMedicalHistory() {  
 return "Вывод истории болезни пациента с COVID-19";  
 }  
 @Override  
 public void payServices() {  
 System.out.println("Услуги оплачены");  
 }  
  
 public static int getCovidCounter(){  
 return covidCounter;  
 }  
}

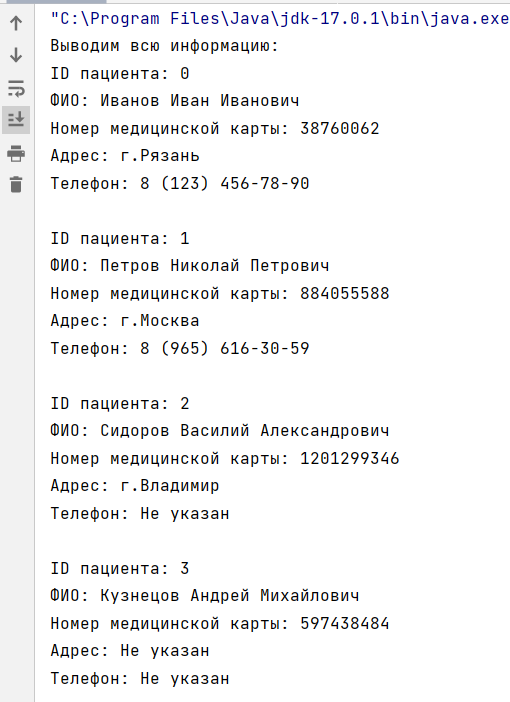
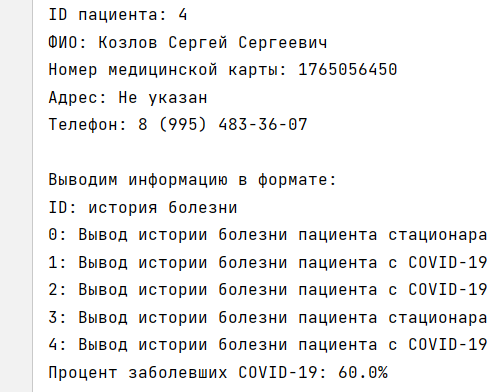
Основная программа

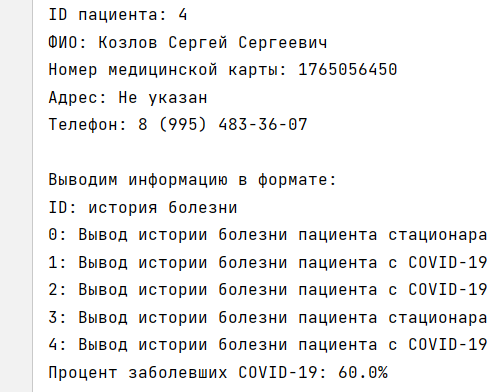
import java.util.ArrayList;  
import java.util.Random;  
import java.util.function.Function;  
  
public class App {  
 public static void main(String[] args) {  
 ArrayList<PatientManager> patients = getPatients();  
 System.out.println("Выводим всю информацию:");  
 printInfo(patients, PatientManager::toString);  
 System.out.println("Выводим информацию в формате:");  
 System.out.println("ID: история болезни");  
 printInfo(patients, p -> p.getId() + ": "+p.getMedicalHistory());  
 System.out.println("Процент заболевших COVID-19: " +  
(double)CovidPatient.getCovidCounter()/PatientInfo.getNumOfPatient()

\*100+"%");  
 }  
  
 public static void printInfo(ArrayList<PatientManager> patients, Function<PatientManager, String> action){  
 patients.forEach(patient -> System.out.println(action.apply(patient)));  
 }  
  
 public static ArrayList<PatientManager> getPatients(){  
 ArrayList<PatientManager> arr = new ArrayList<PatientManager>();  
 arr.add(new HospitalPatient("Иванов", "Иван", "Иванович", "г.Рязань", "8 (123) 456-78-90"));  
 arr.add(new CovidPatient("Петров", "Николай", "Петрович","г.Москва", "8 (965) 616-30-59"));  
 arr.add(new CovidPatient("Сидоров", "Василий", "Александрович"));  
 arr.get(arr.size()-1).setAddress("г.Владимир");  
 arr.add(new HospitalPatient("Кузнецов", "Андрей", "Михайлович"));  
 arr.add(new CovidPatient("Козлов", "Сергей", "Сергеевич"));  
 arr.get(arr.size()-1).setPhone("8 (995) 483-36-07");  
 Random random = new Random();

arr.forEach(patient -> patient.regPatient(Math.abs(random.nextInt())));  
 return arr;  
 }  
}

Окно вывода:





**Вывод:** мы изучили методы реализации интерфейсов, приобрели навыки использования интерфейсов в Java-программах.