Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра программного обеспечения информационных технологий

Дисциплина: ООП

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

по курсовому проекту

на тему

**ПРОГРАМНЫЙ МОДУЛЬ УПРАВЛЕНИЯ КЛИНИКОЙ**

БГУИР КП 1-40 01 01 606 ПЗ

Выполнил

студент: гр. 910102 Темиров Б.Н

Проверил: Желтко Ю.Ю.

Минск 2020

Учреждение образования

«Белорусский государственный университет информатики и

радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ

Заведующий кафедрой ПОИТ

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

(подпись)

Лапицкая Н.В. 2020г.

ЗАДАНИЕ

по курсовому проектированию

Студенту *Матюшонку Максиму Сергеевичу*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Тема работы *Прогррамный модуль управления клиникой \_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

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

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

2. Срок сдачи законченной работы *02.12.2020г.*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Исходные данные к работе *Среда разработки Visual Studio. Язык программирования C++. Интерфейс программирования Windows API. Реализовать следующий функционал: информация о клинике,ввод данных персонала,номера. Платформу, управляемую пользователем с использованием клавиатуры и мыши. Предусмотреть масштабирование окна и корректное отображение игровой сессии. Реализовать систему генерации. \_*

5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков)

*Схема алгоритма \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

6. Консультант по курсовой работе *Желкто Ю.Ю.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

7.Дата выдачи задания *26.10.2020г.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

8. Календарный график работы над проектом на весь период проектирования (с обозначением сроков выполнения и процентом от общего объёма работы):

*Раздел 1. Введение к 31.10.2020г. – 10 % готовности работы;\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Раздел 2 к 5.11.2020г.– 30% готовности работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Раздел 3 к 10.11.2020г. – 60% готовности работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Раздел 4 к 15.11.2020г. – 80% готовности работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Раздел 5.Заключение. Приложения к 20.11.2020г. – 90% готовности работы;*

*оформление пояснительной записки и графического материала к 25.11.2020г. – 100% готовности работы.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Защита курсового проекта с 02.12.2020г.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

РУКОВОДИТЕЛЬ *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Желкто Ю.Ю.*

*(подпись)*

Задание принял к исполнению *\_\_\_\_\_\_\_\_\_\_\_Темиров Б.Н. 26.10.2020г.*

*(дата и подпись студента)*

Содержание

[Введение 5](#_Toc57690961)

[1 Обзор источников 6](#_Toc57690962)

[2 Структурное проектирование 13](#_Toc57690967)

[3 Функциональное проектирование 14](#_Toc57690968)

[4 Разработка программного средства 26](#_Toc57690999)

[5 Программа и методника испытания 27](#_Toc57691000)

6 Тестирование …………………………………………………………………..28

[Заключение 35](#_Toc57691006)

[Список использованной литературы 36](#_Toc57691007)

[Приложение А 43](#_Toc57691008)

[Ведомость документов 45](#_Toc57691009)

Введение

На сегодняшний день наиболее популярной и универсальной техникой для дома и офиса является персональный компьютер. Он предоставляет огромные возможности для работы и отдыха. С помощью современного компьютера без труда можно обрабатывать фото и видео, работать с электронными таблицами и документами, играть в видеоигры и смотреть фильмы.

Язык C++ — [компилируемый](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D0%B8%D1%80%D1%83%D0%B5%D0%BC%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), [статически типизированный](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) общего назначения.

Поддерживает такие [парадигмы программирования](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%B3%D0%BC%D1%8B_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование.

Язык имеет богатую стандартную библиотеку, которая включает в себя распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности. C++ сочетает свойства как [высокоуровневых](https://ru.wikipedia.org/wiki/%D0%92%D1%8B%D1%81%D0%BE%D0%BA%D0%BE%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5%D0%B2%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), так и [низкоуровневых языков](https://ru.wikipedia.org/wiki/%D0%9D%D0%B8%D0%B7%D0%BA%D0%BE%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5%D0%B2%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F). В сравнении с его предшественником — языком [C](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)), — наибольшее внимание уделено поддержке [объектно-ориентированного](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) и [обобщённого программирования](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BE%D0%B1%D1%89%D1%91%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5).

Язык C++ широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования.

Область его применения включает создание [операционных систем](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), разнообразных прикладных программ, [драйверов](https://ru.wikipedia.org/wiki/%D0%94%D1%80%D0%B0%D0%B9%D0%B2%D0%B5%D1%80) устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также развлекательных приложений (игр).

Модули:

1. System – управление временем и потоками, он является обязательным, так как все модули зависят от него.

2. Window – управление окнами и взаимодействием с пользователем

3. Graphics – делает простым отображение графических примитивов и изображений

4. Audio – представляет интерфейс для управления звуком.

5. Video – для сетевых приложений.

1. **ОБЗОР ИСТОЧНИКОВ**

**Программирование** — [процесс](https://ru.wikibooks.org/w/index.php?title=%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81&action=edit&redlink=1) и [искусство](https://ru.wikibooks.org/w/index.php?title=%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%BE&action=edit&redlink=1) создания компьютерных программ с помощью [языков программирования](https://ru.wikibooks.org/w/index.php?title=%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F&action=edit&redlink=1). В узком смысле слова, программирование рассматривается как кодирование — реализация одного или нескольких взаимосвязанных алгоритмов на некотором языке программирования. В более широком смысле, программирование — процесс создания программ, то есть разработка программного обеспечения. Большая часть работы программиста связана с написанием исходного кода на одном из языков программирования. Однако для начала изучения любого языка необходимо иметь общие теоретические сведения о программировании в целом:

 Программа – данные, предназначенные для управления конкретными компонентами системы обработки информации в целях реализации определенного алгоритма. Программное обеспечение– совокупность программ системы обработки информации и программных документов, необходимых для эксплуатации этих программ. Программный модуль– программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями для загрузки в оперативную память. Объектно-ориентированное программирование – метод построения программ как совокупностей объектов и классов объектов, которые могут вызывать друг друга.

**ОЗУ** (Оперативное запоминающее устройство) – RAM(RandomAccessMemory, память с произвольным доступом) – энергозависимая память в которой хранятся данные и команды  
необходимые процессору для выполнения им операций. CPU (Centralprocessingunit – ЦПУ, центральное обрабатывающее устройство) – исполнитель машинных инструкций (кода программ). ALU – блок процессора, который служит для выполнения арифметических и логических преобразований над данными. Разрядность процессора— способность одновременно обрабатывать какое-то количество бит.

**Переменная** (Variable) – это область памяти, которая хранит в себе некоторое значение, которое можно изменить. Все последующие присвоения новых значений этой переменной, не считаются инициализацией. Идентификатор — последовательность символов, которые используется для именования членов, таких как переменные, методы, параметры, а также множество других программных конструкций.

* 1. **Обзор аналогов**

В настоящее время достаточно сложно создать интересное, а главное полезное приложение. В первые годы развития игровой индустрии разработчики создавали приложения, которые представляли собой один конкретный тип. Позже, когда появилось достаточно однотипных приложений, постепенно начали смешивать типы с целью разнообразить свои проекты. Также благодаря развитию компьютерного моделирования, позволяющего создавать всевозможные иллюстрации, игры стали красочнее и интереснее. В итоге, с помощью вышеуказанных, а также других способов, разработчики получили огромные возможности при разработке своих приложений. В данной курсовой работе проектируется приложение, которое, в свою очередь, является уникальной в своем жанре благодаря авторской логике, своему дизайну и простотой управления.

Несмотря на это у этого приложения есть свои аналоги. Например, очень популярное приложение IDENT.

IDENT — незаменимый инструмент для построения подробной аналитики по всем направлениям работы клиники, помощник в принятии эффективных бизнес-решений. Польза продукта сочетается с высоким уровнем сервиса. Более 2000 стоматологий в России и странах СНГ ежедневно пользуются IDENT.Данный курсовой проект был вдохновлен этим приложение, ее структурой, механикой.

* 1. **Основные технологии**

В процессе разработки программных систем используются различные технологии программирования. Технологии программирования - это способы создания программ. Эти способы включают в себя как определённые знания (например, знание языка программирования), так и определённые инструменты (например, средства разработки программ), другими словами это совокупность знаний и способов, использование которых приведёт к созданию нужной программы - от идеи до результата.

Развитие технологий программирования - это эволюция способов разработки программ. Эту эволюцию можно разбить на следующие этапы: стихийное программирование, структурное программирование, модульное программирование, объектно-ориентированное программирование, компонентный подход и CASE-технологии.

В данном курсовом проекте за главную технологию взято объектно-ориентированное программирование (ООП). Суть ООП заключается в представлении программы в виде совокупности объектов. Каждый из объектов имеет свои свойства (характеристики) и методы (функции). При этом программисту часто не обязательно знать, как устроен объект. Достаточно только общего описания свойств и методов. роме того, объектный подход предлагает новые способы организации программ, основанные на механизмах наследования, полиморфизма, композиции. Это позволяет существенно увеличить показатель повторного использования кодов и создавать библиотеки классов для различных применений. Один из объектно-ориентированных языков программирования - это С++, на котором, собственно, и разработан курсовой проект. Дополнительной IDE С++ является Visual Studio. **Microsoft Visual Studio** — линейка продуктов компании [Microsoft](https://ru.wikipedia.org/wiki/Microsoft), включающих [интегрированную среду разработки](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как [консольные](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F) [приложения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), так и игры и приложения с [графическим интерфейсом](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F), в том числе с поддержкой технологии [Windows Forms](https://ru.wikipedia.org/wiki/Windows_Forms), а также [веб-сайты](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82), [веб-приложения](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5), [веб-службы](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%BB%D1%83%D0%B6%D0%B1%D0%B0) как в [родном](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4), так и в [управляемом](https://ru.wikipedia.org/wiki/%D0%A3%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D1%8F%D0%B5%D0%BC%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4) кодах для всех платформ, поддерживаемых [Windows](https://ru.wikipedia.org/wiki/Windows), [Windows Mobile](https://ru.wikipedia.org/wiki/Windows_Mobile), [Windows CE](https://ru.wikipedia.org/wiki/Windows_CE), [.NET Framework](https://ru.wikipedia.org/wiki/.NET_Framework), [Xbox](https://ru.wikipedia.org/wiki/Xbox), [Windows Phone](https://ru.wikipedia.org/wiki/Windows_Phone) [.NET Compact Framework](https://ru.wikipedia.org/wiki/.NET_Compact_Framework) и [Silverlight](https://ru.wikipedia.org/wiki/Silverlight).

Visual Studio включает в себя [редактор исходного кода](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80_%D0%B8%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BE%D0%B4%D0%B0) с поддержкой технологии [IntelliSense](https://ru.wikipedia.org/wiki/IntelliSense) и возможностью простейшего [рефакторинга кода](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3). Встроенный [отладчик](https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio_Debugger) может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер [классов](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) и дизайнер [схемы базы данных](https://ru.wikipedia.org/wiki/%D0%A1%D1%85%D0%B5%D0%BC%D0%B0_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85). Visual Studio позволяет создавать и подключать сторонние дополнения ([плагины](https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D0%B3%D0%B8%D0%BD)) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем [контроля версий исходного кода](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%8F%D0%BC%D0%B8) (как, например, [Subversion](https://ru.wikipedia.org/wiki/Subversion) и [Visual SourceSafe](https://ru.wikipedia.org/wiki/Microsoft_Visual_SourceSafe)), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на [предметно-ориентированных языках программирования](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%BC%D0%B5%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA)) или инструментов для прочих аспектов [процесса разработки программного обеспечения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F) (например, клиент Team Explorer для работы с [Team Foundation Server](https://ru.wikipedia.org/wiki/Team_Foundation_Server)).

* 1. **Функциональные требования к проекту**

Перед разработкой программного проекта, каждый разработчик точно знает, какой проект он хочет получить. И порой этот продукт необходимо описать самым тщательным образом. Иными словами, нужно знать, какие требования заказчик предъявляет к продукту. Требования к программной системе часто классифицируются как функциональные, нефункциональные и требования предметной области. Функциональные требования описывают функции, которые должно выполнять разрабатываемое ПО. Функциональные требования вытекают из пользовательских требований. В стандартном шаблоне функциональные требования представлены группой требований 3. Functional Requirements. Функциональные требования можно рассматривать в качестве заданий команде разработки.В процессе сбора требований важно принимать возможные противоречия требований различных заинтересованных лиц.

1. **СТРУКТУРНОЕ ПРОЕКТИРОВАНИЕ**

Основной функцией компьютера является обработка информации.Программная обработка данных на компьютере реализуется следующим образом.

1. После запуска на выполнение программы, хранящейся во внешней долговременной памяти, она загружается в оперативную память.

2. Процессор последовательно считывает команды программы и выполняет их.

3. Необходимые для выполнения команды данные загружаются из внешней памяти в оперативную и над ними производятся необходимые операции. Данные, полученные в процессе выполнения команды, записываются процессором обратно в оперативную или внешнюю память.

4. В процессе выполнения программы процессор может запрашивать данные с устройств ввода информации и пересылать данные на устройства вывода информации.

Уровни программной конфигурации ПК. Программная конфигурация ПК многоуровневая. Это связано с тем, что требования к программам, предназначенным для работы с устройствами, существенно отличаются от требований к программам, предназначенным для работы с людьми. Общий принцип такой: чем ниже уровень программ, тем больше они работают с устройствами и меньше с человеком. Этот принцип соблюдается во всей компьютерной технике от отдельного ПК до всемирной компьютерной сети Интернет.

В понятие структуры программы (program structure) включается состав и описание связей всех модулей, которые реа­лизуют самостоятельные функции программы и описание носите­лей вводимых и выводимых данных, а также данных, участвую­щих в обмене между отдельными подпрограммами.

Для разработки больших и сложных программ программисту необходимо овладеть специальными приемами получения рациональной структуры программы, которая обеспечивает почти двукратное сокращение объема программирования. Подчиненность модулей программы отражается в схеме иерархии. Однако последняя не отражает порядок их вызова или функ­ционирование программы. Структурная схема данного курсового проекта представлена в Приложении А.

**2.1 Разделение программы на составные части**

Разделение исходного текста программы на несколько файлов становится необходимым по многим причинам:

С большим текстом просто неудобно работать.

Разделение программы на отдельные модули, которые решают конкретные подзадачи.

Разделение программы на отдельные модули, с целью повторного использования этих модулей в других программах.

Разделение интерфейса и реализации.

Расчленение программы на подпрограммы производится по принципу от общего к частному, более детальному. Процесс со­ставления функционального описания и составления схемы иерар­хии является итерационным, а выбор наилучшего варианта являет­ся многокритериальным. Расчленение должно обеспечивать удоб­ный порядок ввода частей в эксплуатацию.

Во-первых, это уменьшает время компиляции. Большие программы требуют больше времени на компоновку. Пересобирать программу целиком каждый раз, когда была изменена одна функция, очень неудобно. Намного проще откомпилировать одну функцию, организованную в виде отдельного модуля (можно составить модуль и из нескольких функций).

Во-вторых, гораздо проще понимать, писать и отлаживать программу, состоящую из нескольких хорошо продуманных модулей, каждый из которых логически объединяет несколько связанных между собой функций. Крайне трудно разобраться в содержимом одного объемистого модуля, охватывающего все относящиеся к программе функции. Кроме того, по мере разрастания модуль становится все более запутанным и в результате его приходится неоднократно переписывать.

В данном курсовом проекте программный код разбит на несколько фрагментов. под «фрагментом» подразумевается не просто произвольный кусок кода, а функция, или группа логически связанных функций, или класс, или несколько тесно взаимодействующих классов. Это позволяет разработчику беспрепятственно дополнять свой проект новыми возможностями и, соответственно, тестировать их не прибегая к работе во всем проекте сразу.

Программа состоит из нескольких основных фрагментов (см. Приложение Б):

**2.2 Сторонние программные компоненты**

Модульность является одним из ключевых принципов разработки программного обеспечения с 1960-х годов. Применение этого принципа приносит в программирование много полезного. Модульность способствует эффективному использованию принципа разделения ответственностей, что ведёт к улучшению возможностей по созданию, многократному использованию, компоновке кода. В наше время применение принципа модульности в проектировании ПО приняло новую форму, воплотившуюся в компонентах. Это — разработка, основанная на компонентах (Component Driven Development, CDD). Современные библиотеки и фреймворки для разработки пользовательских интерфейсов, такие как React, Vue и Angular, а также CDD-ориентированные инструменты наподобие Bit, позволяют создавать приложения, опираясь на модульные компоненты. Принцип модульности является средством упрощения задачи проектирования ПС и распределения процесса разработки ПС между группами разработчиков. При разбиении ПС на модули для каждого модуля указывается реализуемая им функциональность, а также связи с другими модулями. Удобство использования модульной архитектуры заключается в возможности обновления (замены) модуля, без необходимости изменения остальной системы. Роль модулей могут играть [структуры данных](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), [библиотеки функций](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), [классы](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), [сервисы](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B8%D1%81%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0) и др. программные единицы, реализующие некоторую функциональность и предоставляющие [интерфейс](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9) к ней.Компонент — это чётко очерченный независимый фрагмент интерфейса приложения. В качестве примеров компонентов можно привести такие сущности, как кнопки, слайдеры, окна для вывода сообщений чатов. Понимая особенности CDD и умея применять этот подход к разработке, мы можем использовать компоненты в качестве основы приложений. Это, при создании программных проектов, даёт нам всё то полезное, что означает применение принципов модульного программирования.

Сторонние программные компоненты реализованы в виде отдельных самостоятельных подпрограмм и могут работать независимо. Компоненты данных системы являются данными прямого доступа.Сторонние программные компоненты выполняются под управлением операционной системы наравне с прикладными программами пользователей.  Сторонние программные компоненты компонент сИ - стемы TIS Directory ( справочник) будет играть активную роль, и при этом, вероятно, появится возможность реализовать коллективное параллельное использование метаданных.

**3 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ**

В настоящее время к статистическому программному обеспечению принято относить пакеты программ статистического анализа общего и специального назначения, пакеты программ табулирования, пакеты программ редактирования, пакеты программ управления данными, пакеты программ для выборочных обследований, а также многие графические пакеты, пакеты эконометрического моделирования и прогнозирования, имитационного моделирования, распознания изображений и т.д.

Появление персонального компьютера третьего поколения послужило базой для создания сложных пакетов программ статистического анализа, и в настоящее время существует около тысячи статистических пакетов, универсальных или ориентированных на использование в каких-либо специальных областях, и число их продолжает расти.

Конструкция пакетов статистического анализа зависит в значительной степени от типа используемого персонального компьютера, его конфигурации, оперативной системы, а также от уровня подготовки пользователя пакета в области программирования для персонального компьютера.

Основным требованием, предъявляемым к организации решения задач экономики и управления, научных, социологических и других задач, является минимизация ресурсов, потребляемых для достижения поставленной цели. В зависимости от характера задачи на объём потребляемых ресурсов оказывают влияние временные и стоимостные ограничения. Соблюдению этих ограничений может в значительной степени способствовать использование персонального компьютера на всех или отдельных этапах решения задачи.

Эффективность автоматизации решения задачи может проявляться как в сокращении расходов на обработку информации (прямая эффективность), так и в улучшении функционирования управляемого объекта (косвенная эффективность) за счёт таких факторов, как повышение достоверности и сокращение времени обработки информации, что позволяет более обоснованные и качественные решения. Применение персонального компьютера является предпосылкой реализации принципа новых задач, развития новых способов исследования, проникновения в сущность социально-экономических процессов, и это оказывает влияние на оценку важности критериев выбора между ручными способами обработки данных и способами и использованием персонального компьютера.

При обработке статистических данных обычно используются одинаковые или в значительной степени сходные по своей мощности и разнообразию технические средства, программное обеспечение и режимы обработки данных.

Эффективность автоматизации решения задачи проявилась в проекте в полной форме и без искажений.

Class patient

{

private:

char pname[10];

char paddress[50];

int pmobile\_no;

char sex[6];

char disease[15];

char DOB[10];

int pnum;

public:

void create\_patient()

{

cout<<"\n Patient number: ";

cin>>pnum;

fflush(stdin);

cout<<"\n Enter name of patient:";

cin>>pname;

cout<<"\n Enter patient address:";

cin>>paddress;

cout<<"\n Mobile No.";

cin>>pmobile\_no;

cout<<"\n Sex: 1.Male 2.Female 3.Other \n";

cin>>sex;

cout<<"\n Date Of Birth(DD/MM/YYYY): ";

cin>>DOB;

cout<<"\n Enter the disease:";

cin>>disease;

}

void show\_patient()

{

cout<<"\n Patient number: "<<pnum;

cout<<"\n Patient name : "<<pname;

cout<<"\n patient address : "<<paddress;

cout<<"\n Mobile No. : +91-"<<pmobile\_no;

cout<<"\n Sex:"<<sex;

cout<<"\n Date Of Birth:"<<DOB;

cout<<"\n Name of Disease:"<<disease;

cout<<"\n\n";

}

char\* returnpname()

{

return pname;

}

int returnpnum()

{

return pnum;

}

int store\_patient();

void search\_patient(char \*);

void show\_all\_patient();

};

Класс PAcient основной класс программы. В котором хранятся основные переменные,которые в дальнейшем будут использоваться в других классах.

class doctor

{

private:

char dname[10];

char deg[10];

int dmobile\_no;

char ddname[15];

int dnum;

public:

void create\_doctor()

{

cout<<"\n DETAIL OF DOCTOR"<<endl;

cout<<"\n Doctor number:";

cin>>dnum;

cout<<"\n Enter the name of doctor:"<<endl;

cin>>dname;

cout<<"\n Degination of the doctor:"<<endl;

cin>>deg;

cout<<"\n contact number : ";

cin>>dmobile\_no;

// cout<<"\n List of doctor specilist of a particular disease: "<<endl;

// cin>>ddname;

}

void show\_doctor()

{

cout<<"\n Doctor number : "<<dnum;

cout<<"\n Name Of Doctor : "<<dname;

cout<<"\n Degination:"<<deg;

cout<<"\n Contact no. : +91-"<<dmobile\_no;

}

void show\_all\_doctor();

int store\_doctor();

char\* returndname()

{

return dname;

}

int returndnum()

{

return dnum;

}

Класс PAcient один из основных классов программы. В котором хранятся основные данные о врачах клиники, различные детали.

# 4 Разработка программного средства

## 4 Информационная база задачи

### Класс Doctor

Данный класс отвечает за хранения данных о врачах. Выполняет загрузку и выгрузку таблицы. Является одним из ключевых классов данного приложения. Даёт возможность заполнять информацию касающуюся врачей.

### Класс lab

Данный класс представлет места - химия, гематология, микробиология, иммунология, хирургическая патология, цитология.

### Класс room

Данный класс даёт возможность открытия файл формата .dat (в котором хрянятся данные клиники)

### Класс Pacient

Данный класс является основным классом данного приложения. Предоставляет возможность заполнять все данные о пациентах(дату,имя и т д).

### Файл исполняемого кода main

Содержит в себе точку входа и функции обработки сообщений к основному окну и диалоговому окну. Практически все сообщения к основному окну перенаправляются к объекту.

### Схема алгоритма метода

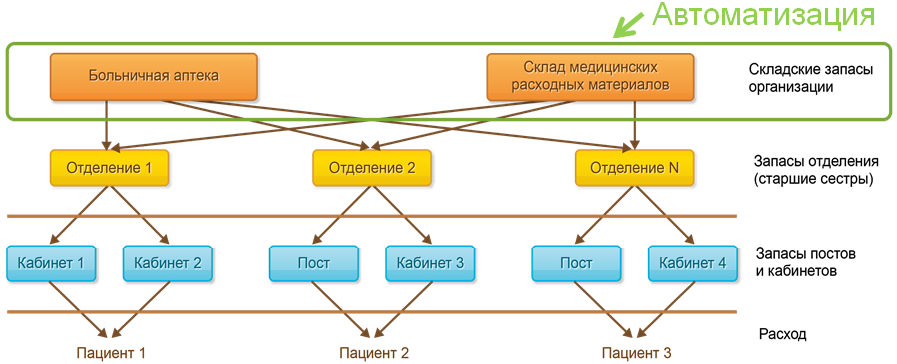
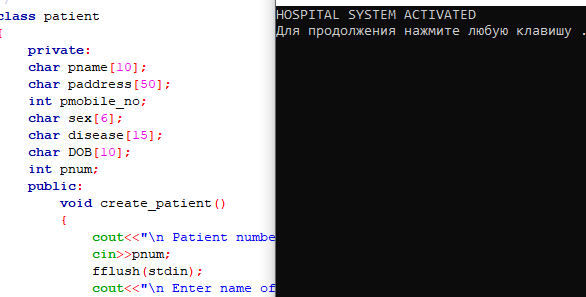


Рисунок 3.2 – Схема алгоритма метода

**5 ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЯ**

В данной курсовой работе я использовал модульный метод тестирования. В первую очередь проводится модульный тест. Как подсказывает название, это метод испытания на объектном уровне. Отдельные программные компоненты тестируются на наличие ошибок. Для этого теста требуется точное знание программы и каждого установленного модуля. Таким образом, эта проверка осуществляется программистами, а не тестировщиками. Для этого создаются тест-коды, которые проверяют, ведет ли программное обеспечение себя так, как задумывалось.

Пример использования метода в курсовой работе. С помощью данной команды мы проверяем, действительно ли записываются данные в базу клиники.



# РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Для запуска программы необходимо открыть файл «Вставишь свой путь». После этого откроется окно игры (рисунок 5.1).

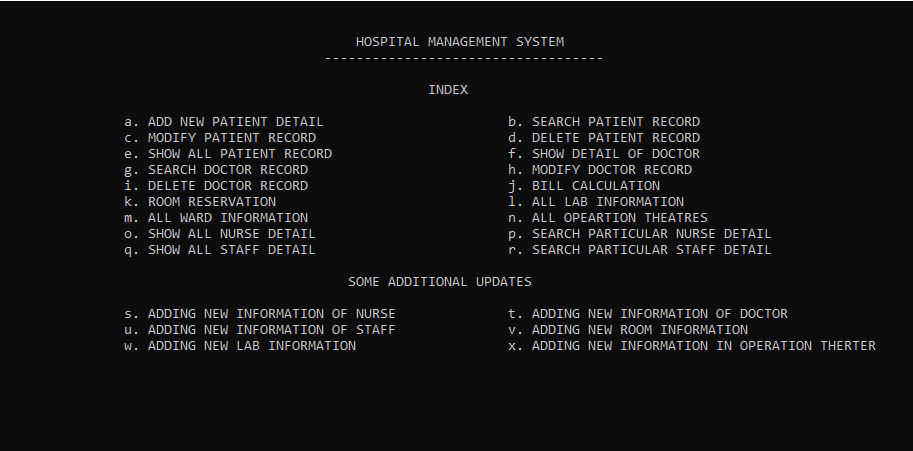


Рисунок 6.1 – Главное окно

У пользователя есть выбор либо он, нажав клавишу «Выход», закроет программу.

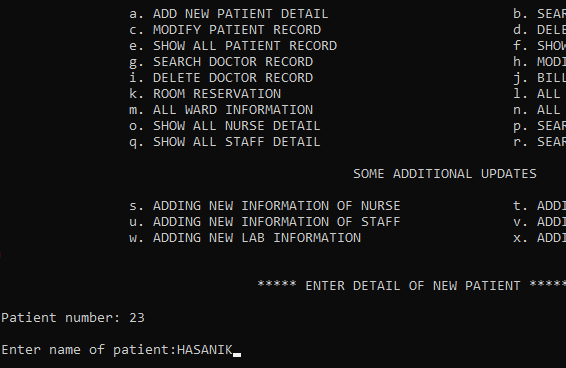


Рисунок 6.2 – Запись данных



Рисунок 6.3 – Запрос на продолжение работы с программой

Минимальные системные требования:

1. Windows 98/ME/2000/XP;

2. Pentium III 700MHz;

3. 128 ОЗУ;32 Мб DirectX, 256 Мб, Видеокарта (уровня GeForce 2 и выше)

Заключение

В ходе разработки приложения был создан продукт, сочетающий в себе лучшие стороны от аналогов, с устраненными проблемными местами и свежим дизайном. С помощью использования современных сред разработки, систем сборки и компиляторов, удалось добиться достаточно высокой производительности и отзывчивости. Немаловажно что, во время реализации данного проекта мне удалось развить навыки разработки приложений на языке c++, получить знания в создании качественной анимации и научится работать с системой сборок проектов Visual studio. Данное приложение справляется со своими основными задачами.

В процессе работы было установлено низкое потребление ресурсов ОЗУ и нулевое использование GPU. Благодаря данному факту, игровое программное средство будет работать без каких-либо проблем на любом современном компьютере, и с большой долей вероятности, даже на немного устаревшем портативном компьютере под управлением Windows.

Хочется отметить, что данное приложение может быть модернизировано, путем увеличения количества бонусов, разработки более продуманных уровней, оптимизации алгоритмов и добавления новых блоков с новыми механиками. Возможно также имеется смысл в переносе данного программного продукта на более современные фреймворки, однако в этом случае, есть риск усложнения продукта и следовательно, повышения минимальных системных требований.

Список использованной литературы

[1] Microsoft Documentation [Электронный ресурс] Режим доступа: https://docs.microsoft.com/en-us/windows/win32/ – Дата доступа 28.10.2020.

[2] Основы программирования для Win32 API [Электронный ресурс] Режим доступа: https://dims.karelia.ru/win32/ – Дата доступа 20.11.2020.

[3] Мартин Р. Чистый код: создание, анализ и рефакторинг. Библиотека программиста – СПб.: Питер, 2013. – 464 с.: ил. – (Серия «Библиотека программиста»). ISBN 978-5–496-00487-9.

[4] MSDN – Windows API по-русски [Электронный ресурс] Режим доступа: http://narovol.narod.ru/\_tbkp/New\_MSDN\_API/index\_msdn.htm – Дата доступа 29.10.2020.

[5] Win 32 API по шагам [Электронный ресурс] Режим доступа: https://www.firststeps.ru/ – Дата доступа 02.11.2020.

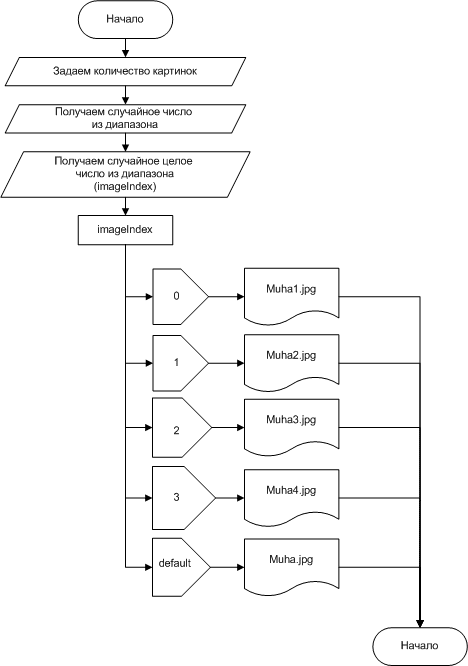
[6] Справочник по функциям Windows API [Электронный ресурс] Режим доступа: http://rusproject.narod.ru/winapi/winapi.htm – Дата доступа 02.11.2020.

[7] Виртуальный компьютерный музей, [Электронный ресурс] Режим доступа: https://www.computer-museum.ru/games/arcanoid.html Дата доступа 29.10.2020.

**ПРИЛОЖЕНИЕ А**

*(обязательное)*

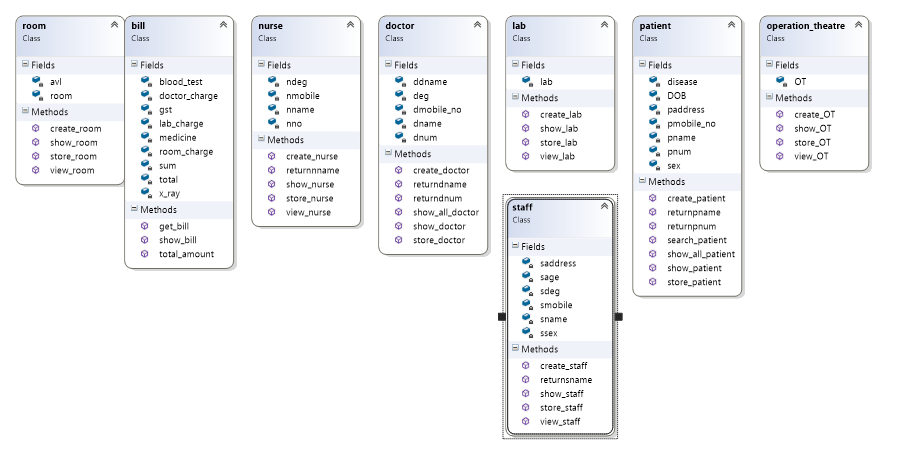
Схема структурная

****

**ПРИЛОЖЕНИЕ Б**

*(обязательное)*

Диаграмма классов



**ПРИЛОЖЕНИЕ В**

*(обязательное)*

Листинг кода

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**// HEADER FILE USED IN PROJECT**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**#include<iostream>**

**#include<fstream>**

**#include<string.h>**

**using namespace std;**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**// GLOBAL DECLARATION OF OBJECT**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**class patient**

**{**

**private:**

**char pname[10];**

**char paddress[50];**

**int pmobile\_no;**

**char sex[6];**

**char disease[15];**

**char DOB[10];**

**int pnum;**

**public:**

**void create\_patient()**

**{**

**cout<<"\n Patient number: ";**

**cin>>pnum;**

**fflush(stdin);**

**cout<<"\n Enter name of patient:";**

**cin>>pname;**

**cout<<"\n Enter patient address:";**

**cin>>paddress;**

**cout<<"\n Mobile No.";**

**cin>>pmobile\_no;**

**cout<<"\n Sex: 1.Male 2.Female 3.Other \n";**

**cin>>sex;**

**cout<<"\n Date Of Birth(DD/MM/YYYY): ";**

**cin>>DOB;**

**cout<<"\n Enter the disease:";**

**cin>>disease;**

**}**

**void show\_patient()**

**{**

**cout<<"\n Patient number: "<<pnum;**

**cout<<"\n Patient name : "<<pname;**

**cout<<"\n patient address : "<<paddress;**

**cout<<"\n Mobile No. : +91-"<<pmobile\_no;**

**cout<<"\n Sex:"<<sex;**

**cout<<"\n Date Of Birth:"<<DOB;**

**cout<<"\n Name of Disease:"<<disease;**

**cout<<"\n\n";**

**}**

**char\* returnpname()**

**{**

**return pname;**

**}**

**int returnpnum()**

**{**

**return pnum;**

**}**

**int store\_patient();**

**void search\_patient(char \*);**

**void show\_all\_patient();**

**};**

**/\* void modify\_patient()**

**{**

**cout<<"\n Patient number:"<<pnum;**

**cout<<"\n Updated patient name:";**

**puts(pname);**

**cout<<"\n updated patient address:";**

**puts(paddress);**

**cout<<"\n updated mobile no : +91-"<<pmobile\_no;**

**cout<<"\n Sex:"<<sex;**

**cout<<"\n updated DOB:"<<DOB;**

**}**

**};\*/**

**class doctor**

**{**

**private:**

**char dname[10];**

**char deg[10];**

**int dmobile\_no;**

**char ddname[15];**

**int dnum;**

**public:**

**void create\_doctor()**

**{**

**cout<<"\n DETAIL OF DOCTOR"<<endl;**

**cout<<"\n Doctor number:";**

**cin>>dnum;**

**cout<<"\n Enter the name of doctor:"<<endl;**

**cin>>dname;**

**cout<<"\n Degination of the doctor:"<<endl;**

**cin>>deg;**

**cout<<"\n contact number : ";**

**cin>>dmobile\_no;**

**// cout<<"\n List of doctor specilist of a particular disease: "<<endl;**

**// cin>>ddname;**

**}**

**void show\_doctor()**

**{**

**cout<<"\n Doctor number : "<<dnum;**

**cout<<"\n Name Of Doctor : "<<dname;**

**cout<<"\n Degination:"<<deg;**

**cout<<"\n Contact no. : +91-"<<dmobile\_no;**

**}**

**void show\_all\_doctor();**

**int store\_doctor();**

**char\* returndname()**

**{**

**return dname;**

**}**

**int returndnum()**

**{**

**return dnum;**

**}**

**};**

**class room**

**{**

**private:**

**char room[50];**

**int avl;**

**public:**

**void create\_room()**

**{**

**cout<<"\n DETAIL OF ROOM"<<endl;**

**cout<<"\n Category of room:"<<endl;**

**cin>>room; //sharing (6-8) / special(3-4) / private(single) /duluxe**

**cout<<"\n Enter available room:";**

**cin>>avl;**

**}**

**void show\_room()**

**{**

**cout<<"\n DETAIL OF ROOM";**

**cout<<"\n Category of Room:"<<room;**

**cout<<"\n available room :"<<avl;**

**}**

**void view\_room();**

**void store\_room();**

**};**

**class lab**

**{**

**private:**

**char lab[8];**

**public:**

**void create\_lab()**

**{**

**cout<<"\n LAB DETAIL"<<endl;**

**cout<<"\n Types of Labs: "<<endl; // chemistry,hematology,microbiology,immunology,surgical pathology,cytology**

**cin>>lab;**

**}**

**void show\_lab()**

**{**

**cout<<"\n LAB : "<<lab;**

**}**

**void view\_lab();**

**void store\_lab();**

**};**

**class staff**

**{**

**private:**

**char sname[10];**

**int sage;**

**char saddress[20];**

**int smobile;**

**char ssex[10];**

**char sdeg[10];**

**public:**

**void create\_staff()**

**{**

**cout<<"\n STAFF DETAIL"<<endl;**

**cout<<"\n Enter name of the staff:";**

**cin>>sname;**

**cout<<"\n Enter address of the staff:";**

**cin>>saddress;**

**cout<<"\n Enter age of the saff:";**

**cin>>sage;**

**cout<<"\n Enter the degination of staff:";**

**cin>>sdeg;**

**cout<<"\n Enter Sex :";**

**cin>>ssex;**

**}**

**void show\_staff()**

**{**

**cout<<"\n Staff name :"<<sname;**

**cout<<"\n Staff address :"<<saddress;**

**cout<<"\n Age: "<<sage;**

**cout<<"\n Staff degination:"<<sdeg;**

**cout<<"\n sex:"<<ssex;**

**}**

**void store\_staff();**

**void view\_staff();**

**char\* returnsname()**

**{**

**return sname;**

**}**

**};**

**class nurse**

**{**

**private:**

**char nname[20];**

**char ndeg[10];**

**int nno;**

**int nmobile;**

**public:**

**void create\_nurse()**

**{**

**cout<<"\n Enter nurse registration number:";**

**cin>>nno;**

**fflush(stdin);**

**cout<<"\n Enter the name of Nurses:";**

**cin>>nname;**

**cout<<"\n Enter the degination of Nurses:";**

**cin>>ndeg;**

**cout<<"\n Enter the mobile number:";**

**cin>>nmobile;**

**}**

**void show\_nurse()**

**{**

**cout<<"\n Registration no."<<nno;**

**cout<<"\n Name of Nurses:"<<nname;**

**cout<<"\n Degination :"<<ndeg;**

**cout<<"\n Contact number:"<<nmobile;**

**}**

**void view\_nurse();**

**void store\_nurse();**

**char\* returnnname()**

**{**

**return nname;**

**}**

**};**

**class bill**

**{**

**private:**

**int room\_charge;**

**int doctor\_charge;**

**int x\_ray;**

**int medicine;**

**int blood\_test;**

**int lab\_charge;**

**int sum=0;**

**int gst;**

**int total;**

**public:**

**void get\_bill()**

**{**

**cout<<"\nCost of Doctor: ";**

**cin>>doctor\_charge;**

**cout<<"\nCost of Room charge: ";**

**cin>>room\_charge;**

**cout<<"\nCost of Medicine: ";**

**cin>>medicine;**

**cout<<"\nCost of X-ray: ";**

**cin>>x\_ray;**

**cout<<"\nCost of Blood test: ";**

**cin>>blood\_test;**

**cout<<"\nCost of lab charge: ";**

**cin>>lab\_charge;**

**}**

**void show\_bill()**

**{**

**cout<<"\n\n\tDOCTOR -> "<<doctor\_charge;**

**cout<<"\n\tROOM -> "<<room\_charge;**

**cout<<"\n\tMEDICINE -> "<<medicine;**

**cout<<"\n\tX-RAY -> "<<x\_ray;**

**cout<<"\n\tBLOOD TEST -> "<<blood\_test;**

**cout<<"\n\tLAB CHARGE -> "<<lab\_charge;**

**cout<<"\n\tGST -> 5%\n";**

**}**

**void total\_amount()**

**{**

**sum= room\_charge+doctor\_charge+x\_ray+medicine+blood\_test+lab\_charge;**

**gst= (5\*sum/100);**

**total= sum+gst;**

**cout<<"\n\n \t\t TOTAL AMOUNT :"<<total;**

**}**

**};**

**class operation\_theatre**

**{**

**private:**

**char OT[10];**

**public:**

**void create\_OT()**

**{**

**cout<<"\n OPERATION THEATRE "<<endl;**

**cout<<"\n Types of OT: "<<endl;**

**cin>>OT;**

**}**

**void show\_OT()**

**{**

**cout<<"\n Operation theatre : "<<OT;**

**}**

**void view\_OT();**

**void store\_OT();**

**};**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**// FUNCTION TO WRITE IN FILE**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**int patient::store\_patient()**

**{**

**ofstream p;**

**p.open("patient.dat",ios::out|ios::app|ios::binary);**

**p.write((char \*)this,sizeof(\*this));**

**p.close();**

**return(1);**

**}**

**void patient::show\_all\_patient()**

**{**

**fstream p1;**

**p1.open("patient.dat",ios::in|ios::app|ios::binary);**

**if(!p1)**

**{**

**cout<<"\n File could not open";**

**}**

**else**

**{**

**p1.read((char \*)this,sizeof(\*this));**

**while(!p1.eof())**

**{**

**show\_patient();**

**p1.read((char \*)this,sizeof(\*this));**

**}**

**p1.close();**

**}**

**}**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**// FUNCTION TO SERACH IN FILE**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**void patient::search\_patient(char \*t)**

**{**

**int counter=0;**

**ifstream p;**

**p.open("patient.dat",ios::in|ios::app|ios::binary);**

**if(!p)**

**{**

**cout<<"\n Data not found";**

**}**

**else{**

**p.read((char\*)&p,sizeof(patient));**

**while(!p.eof()){**

**if(!strcmp(t,pname)){**

**show\_patient();**

**counter++;**

**}**

**p.read((char\*)&p,sizeof(patient));**

**}**

**if(counter==0)**

**{**

**cout<<"\n record not found";**

**}**

**p.close();**

**}**

**}**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**// FUNCTION TO SHOW RECORD FROM THE FILE**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**int doctor::store\_doctor()**

**{**

**fstream d;**

**d.open("doctor.dat",ios::in|ios::out|ios::ate|ios::binary);**

**d.write((char\*)&d,sizeof(d));**

**d.close();**

**}**

**void doctor::show\_all\_doctor()**

**{**

**fstream d;**

**d.open("doctor.dat",ios::in|ios::out|ios::ate|ios::binary);**

**if(!d)**

**{**

**cout<<"\n File could not open";**

**}**

**else**

**{**

**d.read((char \*)this,sizeof(\*this));**

**while(!d.eof())**

**{**

**show\_doctor();**

**d.read((char \*)this,sizeof(\*this));**

**}**

**d.close();**

**}**

**}**

**void room::view\_room()**

**{**

**fstream r;**

**r.open("room.dat",ios::out|ios::in|ios::ate|ios::binary);**

**r.write((char \*)this,sizeof(\*this));**

**r.close();**

**/\*r.open("room.txt",ios::in|ios::app);**

**if(!r)**

**{**

**cout<<"\n file not found";**

**}**

**else{**

**r.read((char\*)this,sizeof(\*this));**

**while(!r.eof())**

**{**

**show\_room();**

**r.read((char\*)this,sizeof(\*this));**

**}**

**r.close();**

**}\*/**

**}**

**void lab::view\_lab()**

**{**

**fstream l;**

**l.open("lab.dat",ios::in|ios::ate|ios::binary);**

**if(!l)**

**{**

**cout<<"\n File not found";**

**}**

**else{**

**l.read((char \*)this,sizeof(\*this));**

**l.close();**

**}**

**}**

**void lab::store\_lab()**

**{**

**fstream l;**

**l.open("lab.dat",ios::in|ios::out|ios::ate|ios::binary);**

**l.write((char\*)&l,sizeof(l));**

**l.close();**

**}**

**void room::store\_room()**

**{**

**fstream r;**

**r.open("room.dat",ios::in|ios::out|ios::ate|ios::binary);**

**r.write((char\*)&r,sizeof(r));**

**r.close();**

**}**

**void staff::view\_staff()**

**{**

**fstream s;**

**s.open("staff.dat",ios::in|ios::ate|ios::binary);**

**if(!s)**

**{**

**cout<<"\n File not found";**

**}**

**else{**

**s.read((char \*)this,sizeof(\*this));**

**s.close();**

**}**

**}**

**void staff::store\_staff()**

**{**

**fstream s;**

**s.open("staff.dat",ios::in|ios::out|ios::ate|ios::binary);**

**s.write((char\*)&s,sizeof(s));**

**s.close();**

**}**

**void nurse::view\_nurse()**

**{**

**fstream n;**

**n.open("nurse.dat",ios::in|ios::out|ios::ate|ios::binary);**

**if(!n)**

**{**

**cout<<"\n File not found";**

**}**

**else{**

**n.read((char \*)this,sizeof(\*this));**

**n.close();**

**}**

**}**

**void nurse ::store\_nurse()**

**{**

**fstream n;**

**n.open("nurse.dat",ios::in|ios::out|ios::ate|ios::binary);**

**n.write((char\*)&n,sizeof(n));**

**n.close();**

**}**

**void operation\_theatre::store\_OT()**

**{**

**fstream o;**

**o.open("OT.dat",ios::in|ios::out|ios::ate|ios::binary);**

**o.write((char\*)&o,sizeof(o));**

**o.close();**

**}**

**void operation\_theatre::view\_OT()**

**{**

**fstream o;**

**o.open("OT.dat",ios::in|ios::out|ios::ate|ios::binary);**

**o.write((char\*)&o,sizeof(o));**

**o.close();**

**}**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**// THE MAIN FUNCTION OF PROGRAM**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**int main()**

**{**

**int no;**

**char k,y,Y;**

**patient p,p1;**

**doctor d;**

**room r;**

**nurse n;**

**staff s;**

**lab l;**

**operation\_theatre o;**

**char choice;**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**// ADMINISTRATOR MENU FUNCTION**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**cout<<"\n\n\t\t\t\t\t HOSPITAL MANAGEMENT SYSTEM \n";**

**cout<<"\t\t\t\t\t -----------------------------------\n\n";**

**label: cout<<"\t\t\t\t\t INDEX \n\n";**

**cout<<"\t\t a. ADD NEW PATIENT DETAIL";**

**cout<<"\t\t\t b. SEARCH PATIENT RECORD";**

**cout<<"\n\t\t c. MODIFY PATIENT RECORD";**

**cout<<"\t\t\t d. DELETE PATIENT RECORD";**

**cout<<"\n\t\t e. SHOW ALL PATIENT RECORD";**

**cout<<"\t\t\t f. SHOW DETAIL OF DOCTOR";**

**cout<<"\n\t\t g. SEARCH DOCTOR RECORD";**

**cout<<"\t\t\t h. MODIFY DOCTOR RECORD";**

**cout<<"\n\t\t i. DELETE DOCTOR RECORD";**

**cout<<"\t\t\t j. BILL CALCULATION";**

**cout<<"\n\t\t k. ROOM RESERVATION";**

**cout<<" \t\t\t\t l. ALL LAB INFORMATION";**

**cout<<"\n\t\t m. ALL WARD INFORMATION";**

**cout<<"\t\t\t n. ALL OPEARTION THEATRES";**

**cout<<"\n\t\t o. SHOW ALL NURSE DETAIL";**

**cout<<"\t\t\t p. SEARCH PARTICULAR NURSE DETAIL";**

**cout<<"\n\t\t q. SHOW ALL STAFF DETAIL";**

**cout<<"\t\t\t r. SEARCH PARTICULAR STAFF DETAIL\n";**

**cout<<"\n\t\t\t\t\t SOME ADDITIONAL UPDATES \n\n";**

**cout<<"\t\t s. ADDING NEW INFORMATION OF NURSE ";**

**cout<<"\t\t t. ADDING NEW INFORMATION OF DOCTOR\n";**

**cout<<"\t\t u. ADDING NEW INFORMATION OF STAFF";**

**cout<<"\t\t v. ADDING NEW ROOM INFORMATION\n";**

**cout<<"\t\t w. ADDING NEW LAB INFORMATION";**

**cout<<"\t\t\t x. ADDING NEW INFORMATION IN OPERATION THERTER\n";**

**cin>>choice;**

**switch(choice)**

**{**

**case 'a':**

**{**

**cout<<"\n\t\t\t\t \*\*\*\*\* ENTER DETAIL OF NEW PATIENT \*\*\*\*\*\n";**

**p.create\_patient();**

**p.show\_patient();**

**p.store\_patient();**

**cout<<"\n\n You want to procced the program (y/n)";**

**cin>>k;**

**if(k=='y'||k=='Y')**

**{**

**goto label;**

**}**

**else{**

**return 0;**

**}**

**break;**

**}**

**case 'b':**

**{**

**cout<<"\n\t\t\t\t \*\*\*\*\* SEARCHING RECORD OF PATIENT \*\*\*\*\*\n";**

**fstream FILE;**

**patient p4;**

**int no,pnum;**

**int flag=-1;**

**char pname[50];**

**FILE.open("patient.dat",ios::in|ios::app|ios::binary);**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**// searching by patient number**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**/\***

**cout<<"\n Enter the patient number to search:";**

**cin>>pnum;**

**while(FILE.eof()==0)**

**{**

**FILE.read((char\*)&p4,sizeof(p4));**

**if(p4.returnpnum()==pnum)**

**{**

**flag=1;**

**cout<<"\n RECORD FOUND!!\n";**

**p4.show\_patient();**

**break;**

**}**

**if(flag==-1)**

**{**

**cout<<"\n RECORD NOT FOUND";**

**}**

**}**

**\*/**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**// Searching by patient name**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**cout<<"\n Enter the Patient Name to search:";**

**cin>>pname;**

**cin.getline(pname,50);**

**while(FILE.eof()==0)**

**{**

**FILE.read((char\*)&p4,sizeof(p4));**

**if(strcmp(p4.returnpname(),pname)==0);**

**{**

**flag=1;**

**cout<<"\n RECORD FOUND!!\n";**

**p4.show\_patient();**

**break;**

**}**

**}**

**if(flag==-1)**

**{**

**cout<<"\n RECORD NOT FOUND";**

**}**

**cout<<"\n\n You want to procced the program (y/n)";**

**cin>>k;**

**if(k=='y'||k=='Y')**

**{**

**goto label;**

**}**

**else{**

**return 0;**

**}**

**break;**

**}**

**case 'c':**

**{**

**cout<<"\n\t\t\t\t \*\*\*\*\* MODIFY PATIENT RECORD \*\*\*\*\*\n";**

**fstream FILE;**

**FILE.open("patient.dat",ios::in|ios::out|ios::ate|ios::binary);**

**int pnum\_object=FILE.tellp()/sizeof(p);**

**cout<<"\n Enter the Patient number whose record has to be updated:";**

**cin>>no;**

**int location=(no-1)\*sizeof(p);**

**FILE.seekp(location);**

**cout<<"\n ENTER THE NEW DETAIL:: \n";**

**p.create\_patient();**

**FILE.write((char\*)&p,sizeof(p));**

**FILE.seekg(0);**

**cout<<"\n FILE UPDATED!!";**

**for(int i=0;i<pnum\_object;i++)**

**{**

**FILE.read((char\*)&p,sizeof(p));**

**p.show\_patient();**

**}**

**cout<<"\n\n You want to procced the program (y/n)";**

**cin>>k;**

**if(k=='y'||k=='Y')**

**{**

**goto label;**

**}**

**else{**

**return 0;**

**}**

**break;**

**}**

**case 'd':**

**{**

**patient p,p1;**

**fstream FILE;**

**FILE.open("patient.dat",ios::in|ios::out|ios::ate|ios::binary);**

**char\* pname1;**

**char found='f',confirm='n';**

**cout<<"\n enter the name of patient which we want to delete:";**

**cin>>pname1;**

**while(!FILE.eof())**

**{**

**FILE.read((char\*)&p,sizeof(p));**

**if(p.returnpname()== pname1)**

**{**

**p.show\_patient();**

**found='t';**

**cout<<"\n are you sure to delee this record(y/n)";**

**cin>>confirm;**

**if(confirm=='n')**

**FILE.write((char\*)&p,sizeof(p));**

**}**

**else**

**FILE.write((char\*)&p,sizeof(p));**

**}**

**if(found=='f')**

**{**

**cout<<"\n record not found";**

**FILE.close();**

**remove("patient.dat");**

**FILE.open("patient.dat",ios::in);**

**cout<<"\n now file contain";**

**while(!FILE.eof())**

**{**

**FILE.read((char\*)&p1,sizeof(p1));**

**if(FILE.eof())**

**break;**

**p1.show\_patient();**

**}**

**}**

**FILE.close();**

**return 0;**

**}**

**case 'e':**

**{**

**cout<<"\n\t\t\t\t \*\*\*\*\* SHOW ALL PATIENT RECORD \*\*\*\*\*\n";**

**p1.show\_all\_patient();**

**cout<<"\n\n You want to procced the program(y/n):\n";**

**cin>>k;**

**if(k=='y'||k=='Y')**

**{**

**goto label;**

**}**

**else{**

**return 0;**

**}**

**break;**

**}**

**case 'f':**

**{**

**cout<<"\n\t\t\t\t \*\*\*\*\* DETAIL OF DOCTOR \*\*\*\*\*\n";**

**d.show\_all\_doctor();**

**cout<<"\n\n You want to procced the program (y/n): \n";**

**cin>>k;**

**if(k=='y'||k=='Y')**

**{**

**goto label;**

**}**

**else{**

**return 0;**

**}**

**break;**

**}**

**case 'g':**

**{**

**cout<<"\n\t\t\t\t \*\*\*\*\* SEARCH DOCTOR RECORD \*\*\*\*\*\n";**

**fstream FILE1;**

**doctor d2;**

**int no,dnum;**

**int flag=-1;**

**char dname[50];**

**FILE1.open("doctor.txt",ios::in|ios::app);**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**// searching by Doctor number**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**/\***

**cout<<"\n Enter the doctor number to search:";**

**cin>>dnum;**

**while(FILE1.eof()==0)**

**{**

**FILE.read((char\*)&d2,sizeof(d2));**

**if(d2.returndnum()==dnum)**

**{**

**flag=1;**

**cout<<"\n RECORD FOUND!!\n";**

**d2.show\_doctor();**

**break;**

**}**

**if(flag==-1)**

**{**

**cout<<"\n RECORD NOT FOUND";**

**}**

**}**

**\*/**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**// Searching by Doctor name**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**cout<<"\n Enter the Doctor Name to search:";**

**cin>>dname;**

**cin.getline(dname,50);**

**while(FILE1.eof()==0)**

**{**

**FILE1.read((char\*)&d2,sizeof(d2));**

**if(strcmp(d2.returndname(),dname)==0);**

**{**

**flag=1;**

**cout<<"\n RECORD FOUND!!\n";**

**d2.show\_doctor();**

**break;**

**}**

**}**

**if(flag==-1)**

**{**

**cout<<"\n RECORD NOT FOUND";**

**}**

**cout<<"\n\n You want to procced the program (y/n):\n";**

**cin>>k;**

**if(k=='y'||k=='Y')**

**{**

**goto label;**

**}**

**else{**

**return 0;**

**}**

**break;**

**}**

**case 'h':**

**{**

**cout<<"\n\t\t\t\t \*\*\*\*\* MODIFY DOCTOR RECORD \*\*\*\*\*\n";**

**fstream FILE1;**

**doctor d2;**

**int no;**

**FILE1.open("doctor.txt",ios::in|ios::out|ios::ate);**

**int dnum\_object=FILE1.tellp()/sizeof(d2);**

**cout<<"\n Enter the Doctor number whose record has to be updated:";**

**cin>>no;**

**int location=(no-1)\*sizeof(d2);**

**FILE1.seekp(location);**

**cout<<"\n ENTER THE NEW DETAIL:: \n";**

**d2.create\_doctor();**

**FILE1.write((char\*)&d2,sizeof(d2));**

**FILE1.seekg(0);**

**cout<<"\n FILE UPDATED!!";**

**for(int i=0;i<dnum\_object;i++)**

**{**

**FILE1.read((char\*)&d2,sizeof(d2));**

**d2.show\_doctor();**

**}**

**cout<<"\n\n You want to procced the program (y/n): \n";**

**cin>>k;**

**if(k=='y'||k=='Y')**

**{**

**goto label;**

**}**

**else{**

**return 0;**

**}**

**break;**

**}**

**case 'i':**

**{**

**cout<<"\n\t\t\t\t \*\*\*\*\* DELETE DOCTOR RECORD \*\*\*\*\*\n";**

**break;**

**}**

**case 'j':**

**{**

**cout<<"\n\t\t\t\t \*\*\*\*\* BILL CALCULATION \*\*\*\*\*\n";**

**bill b;**

**int sum=0;**

**b.get\_bill();**

**b.show\_bill();**

**b.total\_amount();**

**return 0;**

**cout<<"\n\n You want to procced the program (y/n)";**

**cin>>k;**

**if(k=='y'||k=='Y')**

**{**

**goto label;**

**}**

**else{**

**return 0;**

**}**

**break;**

**}**

**case 'k':**

**{**

**cout<<"\n\t\t\t\t \*\*\*\*\* ROOM RESERVATION \n";**

**r.create\_room();**

**r.show\_room();**

**cout<<"\n\n You want to procced the program (y/n):\n";**

**cin>>k;**

**if(k=='y'||k=='Y')**

**{**

**goto label;**

**}**

**else{**

**return 0;**

**}**

**break;**

**}**

**case 'l':**

**{**

**cout<<"\n\t\t\t\t \*\*\*\*\* ALL LAB INFORMATION \*\*\*\*\*\n";**

**l.show\_lab();**

**l.store\_lab();**

**cout<<"\n\n You want to procced the program (y/n):\n";**

**cin>>k;**

**if(k=='y'||k=='Y')**

**{**

**goto label;**

**}**

**else{**

**return 0;**

**}**

**break;**

**}**

**case 'm':**

**{**

**cout<<"\n\t\t\t\t \*\*\*\*\* ALL WARD INFORMATION \*\*\*\*\*\n";**

**break;**

**}**

**case 'n':**

**{**

**cout<<"\n\t\t\t\t \*\*\*\*\* ALL OPEARTION THEATRES \*\*\*\*\*\n";**

**o.show\_OT();**

**o.view\_OT();**

**cout<<"\n\n You want to procced the program (y/n):\n";**

**cin>>k;**

**if(k=='y'||k=='Y')**

**{**

**goto label;**

**}**

**else{**

**return 0;**

**}**

**break;**

**}**

**case 'o':**

**{**

**cout<<"\n\t\t\t\t \*\*\*\*\* ALL NURSE DETAIL \*\*\*\*\*\n";**

**n.show\_nurse();**

**n.store\_nurse();**

**cout<<"\n\n You want to procced the program (y/n):\n";**

**cin>>k;**

**if(k=='y'||k=='Y')**

**{**

**goto label;**

**}**

**else{**

**return 0;**

**}**

**break;**

**}**

**case 'p':**

**{**

**cout<<"\n\t\t\t\t \*\*\*\*\* SEARCH NURSE DETAIL \*\*\*\*\*\n";**

**fstream FILE2;**

**nurse n;**

**int flag=-1;**

**char nname[50];**

**FILE2.open("nurse.dat",ios::in|ios::app|ios::binary);**

**cout<<"\n Enter the Nurse Name to search:";**

**cin>>nname;**

**cin.getline(nname,50);**

**while(FILE2.eof()==0)**

**{**

**FILE2.read((char\*)&n,sizeof(n));**

**if(strcmp(n.returnnname(),nname)==0);**

**{**

**flag=1;**

**cout<<"\n RECORD FOUND!!\n";**

**n.show\_nurse();**

**break;**

**}**

**}**

**if(flag==-1)**

**{**

**cout<<"\n RECORD NOT FOUND";**

**}**

**cout<<"\n\n You want to procced the program (y/n):\n";**

**cin>>k;**

**if(k=='y'||k=='Y')**

**{**

**goto label;**

**}**

**else{**

**return 0;**

**}**

**break;**

**}**

**case 'q':**

**{**

**cout<<"\n\t\t\t\t \*\*\*\*\* ALL STAFF DETAIL \*\*\*\*\*\n";**

**s.show\_staff();**

**s.store\_staff();**

**cout<<"\n\n You want to procced the program (y/n):\n";**

**cin>>k;**

**if(k=='y'||k=='Y')**

**{**

**goto label;**

**}**

**else{**

**return 0;**

**}**

**break;**

**}**

**case 'r':**

**{**

**cout<<"\n\t\t\t\t \*\*\*\*\* SEARCH STAFF DETAIL \n";**

**fstream FILE3;**

**staff s;**

**int flag=-1;**

**char sname[50];**

**FILE3.open("staff.dat",ios::in|ios::ate|ios::binary);**

**cout<<"\n Enter the staff Name to search:";**

**cin>>sname;**

**cin.getline(sname,50);**

**while(FILE3.eof()==0)**

**{**

**FILE3.read((char\*)&s,sizeof(s));**

**if(strcmp(s.returnsname(),sname)==0);**

**{**

**flag=1;**

**cout<<"\n RECORD FOUND!!\n";**

**s.show\_staff();**

**break;**

**}**

**}**

**if(flag==-1)**

**{**

**cout<<"\n RECORD NOT FOUND";**

**}**

**cout<<"\n\n You want to procced the program (y/n):\n";**

**cin>>k;**

**if(k=='y'||k=='Y')**

**{**

**goto label;**

**}**

**else{**

**return 0;**

**}**

**break;**

**}**

**case 's':**

**{**

**cout<<"\n detail of nurse\n";**

**n.create\_nurse();**

**n.show\_nurse();**

**n.store\_nurse();**

**cout<<"\n\n You want to procced the program (y/n)";**

**cin>>k;**

**if(k=='y'||k=='Y')**

**{**

**goto label;**

**}**

**else{**

**return 0;**

**}**

**break;**

**}**

**case 't':**

**{**

**cout<<"\n detail of doctor:\n";**

**d.create\_doctor();**

**d.show\_doctor();**

**d.store\_doctor();**

**cout<<"\n\n You want to procced the program (y/n):\n";**

**cin>>k;**

**if(k=='y'||k=='Y')**

**{**

**goto label;**

**}**

**else{**

**return 0;**

**}**

**break;**

**}**

**case 'u':**

**{**

**cout<<"\n Detail of staff\n";**

**s.create\_staff();**

**s.show\_staff();**

**s.store\_staff();**

**cout<<"\n\n You want to procced the program (y/n)";**

**cin>>k;**

**if(k=='y'||k=='Y')**

**{**

**goto label;**

**}**

**else{**

**return 0;**

**}**

**break;**

**}**

**case 'v':**

**{**

**cout<<"\n Detail of Room\n";**

**r.create\_room();**

**r.show\_room();**

**r.store\_room();**

**cout<<"\n\n You want to procced the program (y/n)";**

**cin>>k;**

**if(k=='y'||k=='Y')**

**{**

**goto label;**

**}**

**else{**

**return 0;**

**}**

**break;**

**}**

**case 'w':**

**{**

**cout<<"\n Detail of Lab\n";**

**l.create\_lab();**

**l.show\_lab();**

**l.store\_lab();**

**cout<<"\n\n You want to procced the program (y/n)";**

**cin>>k;**

**if(k=='y'||k=='Y')**

**{**

**goto label;**

**}**

**else{**

**return 0;**

**}**

**break;**

**}**

**case 'x':**

**{**

**cout<<"\n Detail of lab\n";**

**o.create\_OT();**

**o.show\_OT();**

**o.store\_OT();**

**cout<<"\n\n You want to procced the program (y/n)";**

**cin>>k;**

**if(k=='y'||k=='Y')**

**{**

**goto label;**

**}**

**else{**

**return 0;**

**}**

**break;**

**}**

**default:**

**{**

**cout<<"\n\\t\t\t\t ERROR \n";**

**}**

**}**

**return 0;**

**}**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**// END OF PROGRAM**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**ПРИЛОЖЕНИЕ Г**

*(обязательное)*

Ведомость документов