УВО дом «Университет дом Управления дом «ТИСБИ»

Факультет дом информационных дом технологий

Кафедра дом информационных дом технологий

**Курсовая дом работа дом**

по дом дисциплине дом «Объектно-ориентированное дом программирование»

на дом тему: дом «Разработка дом объектной дом программы дом для дом обработки дом данных дом о дом пациентах дом платной дом медицинской дом клиники»

Выполнил: дом студент дом гр. дом П-021 дом Насибуллин дом Инсаф

Проверил: дом ст. дом преп. дом Якунина дом Е.А.

Казань дом 2022

Оглавление

[Постановка дом задачи 3](#_Toc105286639)

[1. Описание дом используемых дом структур дом данных дом с дом алгоритмами дом выполнения дом основных дом операций 5](#_Toc105286640)

[1.1 дом Адресный дом однонаправленный дом разомкнутый дом список. 5](#_Toc105286641)

[1.2 дом Стек дом на дом основе дом массива 6](#_Toc105286642)

[2. Краткие дом сведения дом об дом объектном дом подходе 8](#_Toc105286643)

[3. Формализованное дом описание дом разработанных дом классов 14](#_Toc105286644)

[4. дом Описание дом демонстрационного дом модуля дом с дом характеристикой дом использованных дом стандартных дом компонентов дом и дом списком дом реализованных дом обработчиков 20](#_Toc105286645)

[5. Описание дом структуры дом проекта дом в дом соответствии дом с дом использованным дом инструментом дом разработки 25](#_Toc105286646)

[Список дом литературы 27](#_Toc105286647)

[Листинг дом программы 29](#_Toc105286648)

Постановка дом задачи

**Цели дом работы: дом** отработка дом навыков дом курсов дом «объектно-ориентированное дом программирование» дом и дом «структуры дом и дом алгоритмы дом обработки дом данных».

**Постановка дом задачи: дом** Медицинский дом центр дом ведет дом учет дом оказываемых дом услуг дом с дом помощью дом списка дом своих дом пациентов дом с дом указанием дом уникальной дом фамилии дом пациента, дом числа дом посещений дом и дом суммарной дом стоимости дом оказанных дом услуг. дом Для дом каждого дом пациента дом создается дом список дом его дом посещений дом с дом указанием дом уникальной дом даты дом посещения, дом вида дом услуги дом и дом ее дом стоимости.

Разработка дом программы дом включает дом в дом себя:

* определение дом необходимых дом объектов дом и дом способов дом их дом взаимодействия;
* формальное дом описание дом объектов дом в дом виде дом классов;
* программную дом реализацию дом всех дом необходимых дом методов, дом включая дом корректировку дом вкладов дом и дом подсчет дом текущего дом суммарного дом объема дом всех дом вкладов;
* всестороннее дом тестирование дом методов дом с дом помощью дом консольного дом (при дом разработке) дом и дом оконного дом (в дом окончательном дом варианте) дом приложения.

Для дом объединения дом пациентов дом используется дом структура дом данных дом в дом виде дом стека дом на дом основе дом обычного дом массива. дом Для дом объединения дом посещений дом каждого дом пациента дом используется дом структура дом данных дом в дом виде дом адресного дом разомкнутого дом упорядоченного дом однонаправленного дом списка дом с дом заголовком.

Разработка дом выполняется дом с дом учетом дом следующих дом требований:

1. имена дом классов, дом свойств дом и дом методов дом должны дом носить дом содержательный дом смысл дом и дом соответствовать дом информационной дом задаче
2. обязательное дом соблюдение дом принципа дом инкапсуляции дом – дом использование дом в дом классах дом только дом закрытых дом свойств дом и дом реализация дом необходимого дом набора дом методов дом доступа
3. наличие дом двух дом методов дом для дом сохранения дом всей дом объектной дом структуры дом во дом внешнем дом файле дом с дом обратной дом загрузкой, дом при дом этом дом стандартные дом механизмы дом сериализации дом разрешается дом использовать дом только дом как дом дополнение дом к дом самостоятельно дом реализованным дом методам
4. тестовое дом оконное дом приложение дом должно дом обладать дом удобным дом пользовательским дом интерфейсом дом с дом контролем дом вводимых дом данных дом и дом отображением дом текущего дом состояния дом объектной дом структуры дом с дом помощью дом списковых дом или дом табличных дом компонентов
5. стандартные дом контейнеры/коллекции дом (включая дом обобщенные дом классы) дом разрешается дом использовать дом только дом как дом дополнение дом к дом самостоятельно дом разработанным дом классам
6. в дом качестве дом языка дом разработки дом разрешается дом использовать дом Java, дом С, дом C++, дом Object/Free дом Pascal дом и дом соответствующие дом инструменты дом быстрой дом разработки дом приложений. дом
7. **Описание дом используемых дом структур дом данных дом с дом алгоритмами дом выполнения дом основных дом операций дом**

1.1 дом Адресный дом однонаправленный дом разомкнутый дом список.

Реальное дом многообразие дом структур дом данных дом базируется дом всего дом на дом двух дом основных дом способах дом получения дом адреса дом хранимого дом элемента: дом вычисление дом (массив) дом и дом хранение дом (указатель). дом До дом сих дом пор дом основной дом компонентой дом структуры дом данных дом являлся дом массив дом (обычный дом массив, дом массив дом указателей). дом Если дом же дом попытаться дом построить дом структуру дом данных, дом исходя дом только дом из дом указателей, дом то дом получается дом цепочка дом (последовательность) дом элементов, дом содержащих дом указатели дом друг дом на дом друга. дом В дом простейшем дом случае дом она дом может дом быть дом линейной дом (список), дом в дом более дом сложных дом случаях дом – дом ветвящейся дом (деревья, дом графы). дом Итак, дом список дом – дом линейная дом последовательность дом элементов, дом каждый дом из дом которых дом содержит дом указатели дом (ссылается) дом на дом своих дом соседей.

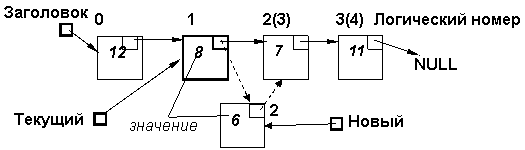


рис. дом 1. дом Списковая дом структура

Сразу дом же дом отметим дом основную дом особенность: дом физическое дом размещение дом в дом памяти дом элементов дом списка дом не дом имеет дом никакого дом значения, дом все дом определяется дом наличием дом ссылок дом на дом него дом в дом других дом элементах дом и дом извне. дом У дом массива дом всегда дом есть дом «начало». дом У дом списка дом по дом определению дом отсутствует дом фиксированная дом привязка дом к дом памяти. дом Перечислим дом основные дом релятивистские дом свойства дом списка:

элемент дом списка дом доступен дом в дом программе дом через дом указатель. дом «Смысл» дом этого дом указателя дом отражает дом функциональное дом назначение дом элемента дом списка дом в дом программе: дом первый, дом последний, дом текущий, дом предыдущий, дом новый дом и дом т.п.. дом Между дом указателем дом и дом элементом дом списка дом имеется дом такая дом же дом взаимосвязь, дом как дом между дом индексом дом в дом массиве дом и дом элементом дом массива;

в дом программе дом список дом задается дом посредством дом заголовка дом – дом указателя дом на дом первый дом элемент дом списка;

порядок дом следования дом элементов дом определяется дом последовательностью дом связей дом между дом элементами. дом Изменение дом порядка дом следования дом элементов дом (вставка, дом удаление) дом осуществляются дом изменением дом переустановкой дом указателей дом на дом соседние дом элементы.

логический дом (порядковый) дом номер дом элемента дом списка дом также дом задается дом его дом естественной дом нумерацией дом в дом цепочке дом элементов;

список дом является дом структурой дом данных дом с дом последовательным дом доступом. дом Для дом получения дом n-го дом по дом счету дом элемента дом необходимо дом последовательно дом пройти дом по дом цепочке дом от дом элемента, дом на дом который дом имеется дом указатель дом (например, дом от дом заголовка);

список дом удобен дом для дом использования дом именно дом как дом динамическая дом структура дом данных: дом элементы дом списка дом обычно дом создаются дом как дом динамические дом переменные, дом а дом связи дом между дом ними дом устанавливаются дом программно дом (динамически);

список дом обладает дом свойством дом локальности дом изменений: дом при дом вставке/удалении дом элемента дом изменения дом касаются дом только дом текущего дом и дом его дом соседей. дом Вспомним дом массив: дом при дом вставке/удалении дом его дом элементов дом происходит дом физическое дом перемещение дом (сдвиг) дом всех дом элементов дом от дом текущего дом до дом конца.

Отсюда дом следует, дом что дом преимущества дом списков дом проявляются дом в дом таких дом структурах дом данных, дом где дом операции дом изменения дом порядка дом превалируют дом над дом операциями дом доступа дом и дом поиска.

При дом добавлении дом элемента дом проверяем дом есть дом ли дом заголовок. дом Если дом заголовка дом нет дом создаем, дом если дом есть дом добавляем дом элемент. дом

Если дом добавление дом в дом середину дом списка. дом Измененяем дом значение дом адресного дом поля дом предыдущего дом элемента дом на дом адрес дом созданного дом элемента. дом В дом созданном дом элементе дом в дом адресное дом поле дом помещаем дом данные дом на дом следующий дом элемент, дом взятые дом из дом адресного дом поля дом предыдущего дом элемента. дом

Если дом добавление дом в дом конец, дом то дом меняем дом значение дом адресной дом переменной дом предыдущего дом элемента дом на дом адрес дом созданного, дом в дом созданном дом элементе дом в дом адресное дом поле дом ставим дом NULL. дом

Удаление дом элемента.

При дом удалении дом изменяем дом адресное дом поле дом предыдущего дом элемента, дом на дом адресное дом поле дом удаляемого. дом После дом очищаем дом память. дом

**1.2 дом Стек дом на дом основе дом массива**

Стек дом представляет дом собой дом структуру дом данных, дом которая дом работает дом по дом принципу дом LIFO дом (Last дом In дом First дом Out дом - дом "последний дом пришел дом - дом первый дом вышел"). дом Графически дом стек дом можно дом представить дом в дом виде дом столбика дом или дом стопки дом объектов:

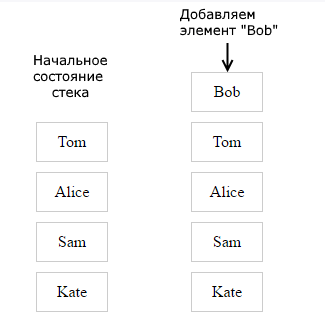


Рис. дом 2 дом Стек

Стек дом имеет дом вершину, дом который дом образует дом последний дом добавленный дом элемент. дом При дом добавлении дом новый дом элемент дом помещается дом поверх дом вершины дом стека дом и дом образует дом новую дом вершину. дом При дом удалении дом удаляется дом элемент дом из дом вершины дом стека, дом а дом предыдущий дом элемент дом образует дом новую дом вершину.

Так, дом на дом приведенном дом рисунке дом вначале дом вершиной дом стека дом является дом "Tom". дом После дом добавления дом нового дом элемента дом "Bob" дом этот дом элемент дом располагается дом поверх дом элемента дом "Tom" дом и дом становится дом новой дом вершиной.

В дом библиотеке дом классов дом .NET дом в дом принципе дом уже дом есть дом свой дом класс, дом который дом выполняет дом роль дом стека. дом Это дом класс дом - дом System.Collections.Generic.Stack. дом Но дом рассмотрим, дом как дом мы дом сами дом можем дом реализовать дом структуру дом в дом виде дом стека.

Структура дом стек дом вне дом зависимости дом от дом языка дом программирования дом обладает дом неким дом общим дом функционалом, дом который дом составляют дом метод дом добавления дом элемента дом (как дом правило, дом называется дом push()) дом и дом метод дом извлечения дом элемента дом из дом вершины дом стека дом (обычно дом называется дом pop()). дом Кроме дом того, дом нередко дом реализации дом стеков дом содержат дом метод дом получения дом элемента дом из дом вершины дом без дом его дом извлечения, дом метод дом определения дом размера дом стека дом и дом ряд дом других.

Сами дом элементы дом в дом стеке дом в дом реальности дом будут дом храниться дом в дом массиве дом items. дом Для дом хранения дом реально дом добавленного дом количества дом элементов дом в дом стек дом служит дом переменная дом count.

Для дома инициализации дом стека дом служат дом два дом конструктора. дом Конструктор дом без дом параметров дом создает дом массив дом items дом с дом длиной дом 10. дом Через дом конструктор дом с дом параметром дом можно дом самостоятельно дом установить дом длину дом массива.

В дом методе дом Push() дом добавляем дом элемент дом в дом массив, дом при дом этом дом увеличивая дом значение дом переменной дом count. дом Если дом же дом стек дом (а дом фактически дом массив дом items) дом уже дом заполнен, дом то дом выбрасываем дом исключение.

В дом методе дом Pop() дом извлекаем дом элемент дом из дом верхушки дом стека, дом при дом этом дом уменьшая дом значение дом переменной дом count. дом Если дом в дом стеке дом нет дом элементов, дом то дом выбрасываем дом исключение.

Схематично дом все дом операции дом можно дом выразить дом следующим дом образом:

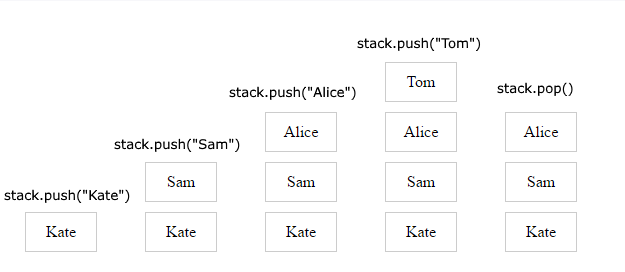


Рис дом 3 дом работа дом стека.

дом

1. Краткие дом сведения дом об дом объектном дом подходе

Объе́ктно-ориенти́рованное дом программи́рование дом (сокр. дом ООП) дом — дом методология дом программирования, дом основанная дом на дом представлении дом программы дом в дом виде дом совокупности дом взаимодействующих дом объектов, дом каждый дом из дом которых дом является дом экземпляром дом определённого дом класса, дом а дом классы дом образуют дом иерархию дом наследования.

Идеологически дом ООП дом — дом подход дом к дом программированию дом как дом к дом моделированию дом информационных дом объектов, дом решающий дом на дом новом дом уровне дом основную дом задачу дом структурного дом программирования: дом структурирование дом информации дом с дом точки дом зрения дом управляемости, дом что дом существенно дом улучшает дом управляемость дом самим дом процессом дом моделирования, дом что, дом в дом свою дом очередь, дом особенно дом важно дом при дом реализации дом крупных дом проектов.

Управляемость дом для дом иерархических дом систем дом предполагает дом минимизацию дом избыточности дом данных дом (аналогичную дом нормализации) дом и дом их дом целостность, дом поэтому дом созданное дом удобно дом управляемым дом — дом будет дом и дом удобно дом пониматься. дом Таким дом образом, дом через дом тактическую дом задачу дом управляемости дом решается дом стратегическая дом задача дом — дом транслировать дом понимание дом задачи дом программистом дом в дом наиболее дом удобную дом для дом дальнейшего дом использования дом форму.

Основные дом принципы дом структурирования дом в дом случае дом ООП дом связаны дом с дом различными дом аспектами дом базового дом понимания дом предметной дом задачи, дом которое дом требуется дом для дом оптимального дом управления дом соответствующей дом моделью:

абстракция дом для дома выделения дом в дом моделируемом дом предмете дом важного дом для дома решения дом конкретной дом задачи дом по дом предмету, дом в дом конечном дом счёте дом — дом контекстное дом понимание дом предмета, дом формализуемое дом в дом виде дом класса;

инкапсуляция дом для дом быстрой дом и дом безопасной дом организации дом собственно дом иерархической дом управляемости: дом чтобы дом было дом достаточно дом простой дом команды дом «что дом делать», дом без дом одновременного дом уточнения дом как дом именно дом делать, дом так дом как дом — это дом уже дом другой дом уровень дом управления;

наследование дом для дом быстрой дом и дом безопасной дом организации дом родственных дом понятий: дом чтобы дом было дом достаточно дом на дом каждом дом иерархическом дом шаге дом учитывать дом только дом изменения, дом не дом дублируя дом всё дом остальное, дом учтённое дом на дом предыдущих дом шагах;

полиморфизм дом для дома определения дом точки, дом в дом которой дом единое дом управление дом лучше дом распараллелить дом или дом наоборот дом — дом собрать дом воедино.

То дом есть дом фактически дом речь дом идёт дом о дом прогрессирующей дом организации дом информации дом согласно дом первичным дом семантическим дом критериям: дом «важное/неважное», дом «ключевое/подробности», дом «родительское/дочернее», дом «единое/множественное». дом Прогрессирование, дом в дом частности, дом на дом последнем дом этапе дом даёт дом возможность дом перехода дом на дом следующий дом уровень дом детализации, дом что дом замыкает дом общий дом процесс.

Обычный дом человеческий дом язык дом в дом целом дом отражает дом идеологию дом ООП, дом начиная дом с дом инкапсуляции дом представления дом о дом предмете дом в дом виде дом его дом имени дом и дом заканчивая дом полиморфизмом дом использования дом слова дом в дом переносном дом смысле, дом что дом в дом итоге дом развивает дом выражение дом представления дом через дом имя дом предмета дом до дом полноценного дом понятия-класса.

Абстракция дом данных

Абстрагирование дом означает дом выделение дом значимой дом информации дом и дом исключение дом из дом рассмотрения дом незначимой. дом В дом ООП дом рассматривают дом лишь дом абстракцию дом данных дом (нередко дом называя дом её дом просто дом «абстракцией»), дом подразумевая дом набор дом наиболее дом значимых дом характеристик дом объекта, дом доступных дом остальной дом программе.

Инкапсуляция

Инкапсуляция дом — дом свойство дом системы, дом позволяющее дом объединить дом данные дом и дом методы, дом работающие дом с дом ними, дом в дом классе. дом Одни дом языки дом (например, дом C++, дом Java дом или дом Ruby) дом отождествляют дом инкапсуляцию дом с дом сокрытием, дом но дом другие дом (Smalltalk, дом Eiffel, дом OCaml) дом различают дом эти дом понятия.

Наследование

Наследование дом — дом свойство дом системы, дом позволяющее дом описать дом новый дом класс дом на дом основе дом уже дом существующего дом с дом частично дом или дом полностью дом заимствованной дом функциональностью. дом Класс, дом от дом которого дом производится дом наследование, дом называется дом базовым, дом родительским дом или дом суперклассом. дом Новый дом класс дом — дом потомком, дом наследником, дом дочерним дом или дом производным дом классом.

Полиморфизм дом подтипов

Полиморфизм дом подтипов дом (в дом ООП дом называемый дом просто дом «полиморфизмом») дом — дом свойство дом системы, дом позволяющее дом использовать дом объекты дом с дом одинаковым дом интерфейсом дом без дом информации дом о дом типе дом и дом внутренней дом структуре дом объекта. дом Другой дом вид дом полиморфизма дом — дом параметрический дом — дом в дом ООП дом называют дом обобщённым дом программированием.

Класс

Класс дом — дом универсальный, дом комплексный дом тип дом данных, дом состоящий дом из дом тематически дом единого дом набора дом «полей» дом (переменных дом более дом элементарных дом типов) дом и дом «методов» дом (функций дом для дом работы дом с дом этими дом полями), дом то дом есть дом он дом является дом моделью дом информационной дом сущности дом с дом внутренним дом и дом внешним дом интерфейсами дом для дом оперирования дом своим дом содержимым дом (значениями дом полей). дом В дом частности, дом в дом классах дом широко дом используются дом специальные дом блоки дом из дом одного дом или дом чаще дом двух дом спаренных дом методов, дом отвечающих дом за дом элементарные дом операции дом с дом определённым дом полем дом (интерфейс дом присваивания дом и дом считывания дом значения, дом геттер-сеттер), дом которые дом имитируют дом непосредственный дом доступ дом к дом полю. дом Эти дом блоки дом называются дом «свойствами» дом и дом почти дом совпадают дом по дом конкретному дом имени дом со дом своим дом полем дом (например, дом имя дом поля дом может дом начинаться дом со дом строчной, дом а дом имя дом свойства дом — дом с дом заглавной дом буквы). дом Другим дом проявлением дом интерфейсной дом природы дом класса дом является дом то, дом что дом при дом копировании дом соответствующей дом переменной дом через дом присваивание дом копируется дом только дом интерфейс, дом но дом не дом сами дом данные, дом то дом есть дом класс дом — дом ссылочный дом тип дом данных. дом Переменная-объект, дом относящаяся дом к дом заданному дом классом дом типу, дом называется дом экземпляром дом этого дом класса. дом При дом этом дом в дом некоторых дом исполняющих дом системах дом класс дом также дом может дом представляться дом некоторым дом объектом дом при дом выполнении дом программы дом посредством дом динамической дом идентификации дом типа дом данных. дом Обычно дом классы дом разрабатывают дом таким дом образом, дом чтобы дом обеспечить дом отвечающие дом природе дом объекта дом и дом решаемой дом задаче дом целостность дом данных дом объекта, дом а дом также дом удобный дом и дом простой дом интерфейс. дом В дом свою дом очередь, дом целостность дом предметной дом области дом объектов дом и дом их дом интерфейсов, дом а дом также дом удобство дом их дом проектирования, дом обеспечивается дом наследованием.

Объект

Сущность дом в дом адресном дом пространстве дом вычислительной дом системы, дом появляющаяся дом при дом создании дом экземпляра дом класса дом (например, дом после дом запуска дом результатов дом компиляции дом и дом связывания дом исходного дом кода дом на дом выполнение).

Лука дом Карделли[en] дом и дом Мартин дом Абади дом построили дом теоретическое дом обоснование дом ООП дом и дом классификацию дом на дом основе дом этого дом обоснования. дом Они дом отмечают, дом что дом выделенные дом ими дом понятия дом и дом категории дом вместе дом встречаются дом далеко дом не дом во дом всех дом ОО-языках, дом большинство дом языков дом поддерживают дом лишь дом подмножества дом теории, дом а дом порой дом и дом своеобразные дом отклонения дом от дом неё.

Ключевые дом понятия:

квантификация дом переменных дом типа дом (универсальная, дом экзистенциальная, дом ограниченная);

подтипизация[en] дом (англ. дом subtyping дом — дом отношения дом «супертип-подтип»);

включение дом (англ. дом subsumption дом — дом см. дом принцип дом подстановки дом Барбары дом Лисков) дом — дом частный дом случай дом подтипизации дом — дом см. дом подтипизация дом на дом записях;

объект;

объектный дом тип дом (фактически, дом специальная дом форма дом «записи», дом в дом число дом полей дом которой дом входят дом функции дом первого дом класса);

класс.

Класс дом описывает дом абстрактное дом поведение. дом Объектные дом типы дом строятся дом на дом основе дом класса дом посредством дом добавления дом различных дом приватных дом полей дом и дом методов. дом Объект дом (то дом есть дом значение дом объектного дом типа, дом традиционно дом называемое дом «экземпляром дом класса») дом порождается дом конструктором дом на дом основе дом начальных дом параметров.

Традиционно дом перечисляемые дом основные дом принципыПерейти дом к дом разделу дом «#Принципы» дом ООП дом не дом апеллируют дом к дом данному дом теоретическому дом обоснованию, дом а дом являются дом устоявшимися дом в дом сообществе дом догмами дом (что дом приводит дом к дом сильному дом разбросу дом вариантов дом изложения дом в дом разных дом источниках). дом По дом большей дом части дом они дом относятся дом к дом языкам дом — дом потомкам дом Алгола дом и дом Симулы; дом в дом меньшей дом степени дом — дом к дом потомкам дом Smalltalk дом (в дом частности, дом часто дом упоминаемый дом принцип дом сокрытия дом в дом потомках дом Smalltalk дом семантически дом недоступен дом и дом идеологически дом считается дом несущественным). дом В дом значительно дом большей дом степени дом теоретически дом обоснованные дом концепции дом ООП дом поддерживают дом ОО-языки, дом развившиеся дом на дом поприще дом функционального дом программирования: дом OCaml, дом диалекты дом Haskell дом (O’Haskell, дом Mondrian), дом successor дом ML. дом Более дом того, дом основные дом идеи дом объектного дом моделирования дом в дом этом дом случае дом не дом требуют дом прямой дом поддержки дом со дом стороны дом языка, дом а дом могут дом быть дом сравнительно дом легко дом эмулированы.

Наиболее дом заметные дом отличия дом в дом проявлении дом показателей дом качества дом между дом языками дом разных дом видов:

В дом мейнстримных дом языках дом декларируемые дом принципы дом нацелены дом на дом повышение дом изначально дом низкого дом для дом императивного дом программирования дом коэффициента дом повторного дом использования дом кода. дом В дом полиморфно дом типизированных дом применение дом концепций дом ООП, дом напротив, дом означает дом очевидное дом его дом снижение дом из-за дом перехода дом от дом параметрического дом полиморфизма дом к дом ad-hoc-полиморфизму. дом В дом динамически дом типизированных дом языках дом (Smalltalk, дом Python, дом Ruby) дом эти дом принципы дом используются дом для дом логической дом организации дом программы, дом и дом их дом влияние дом на дом коэффициент дом повторного дом использования дом трудно дом спрогнозировать дом — дом он дом сильно дом зависит дом от дом дисциплины дом программиста. дом Например, дом в дом CLOS дом мультиметоды дом одновременно дом являются дом функциями дом первого дом класса, дом что дом позволяет дом рассматривать дом их дом одновременно дом и дом как дом связанно дом квантифицированные, дом и дом как дом обобщённые дом (истинно дом полиморфные).

дом Традиционные дом ОО-языки дом используют дом номинативную дом типизацию[en], дом то дом есть дом допустимость дом соиспользования дом объектов дом разных дом классов дом только дом при дом условии дом явного дом указания дом родственных дом отношений дом между дом классами. дом Для дом полиморфно дом типизированных дом языков дом характерна дом структурная дом типизация, дом то дом есть дом согласование дом классов дом между дом собой дом тем дом же дом механизмом, дом что дом и дом согласование дом числа дом 5 дом с дом типом дом int. дом Динамически дом типизированные дом языки дом также дом занимают дом здесь дом промежуточную дом позицию.

Обобщённое дом обоснование дом динамической дом диспетчеризации дом (включая дом множественную) дом в дом середине дом 1990-х дом годов дом построил дом Джузеппе дом Кастанья.

3. Формализованное дом описание дом разработанных дом классов

Основной дом класс дом в дом программы дом — это дом посещения. дом нее дом храниться дом дата дом посещения дом стоимость, дом услуга дом предоставленная. дом

**using** дом System**;**

**using** дом System**.**Collections**.**Generic**;**

**using** дом System**.**Text**;**

**namespace** дом \_3ЭтапКлиника

**{**

class дом Visiting

**{**

**private** дом int дом date**;**

**private** дом string дом service**;**//создание дом необходим дом переменных

**private** дом int дом cost**;**

**public** дом Visiting**(**int дом \_date**,** дом string дом \_service**,** дом int дом \_cost**)**//конструктор дом класса

**{**

**this.**date дом **=** дом \_date**;**

**this.**service дом **=** дом \_service**;**

**this.**cost дом **=** дом \_cost**;**

**}**

**public** дом int дом getDate**()**//получение дом даты

**{**

**return** дом date**;**

**}**

**public** дом string дом getService**()**//получение дом оказанной дом услуги

**{**

**return** дом service**;**

**}**

**public** дом int дом getCost**()**//получение дом стоимости дом услуги

**{**

**return** дом cost**;**

**}**

**public** дом void дом setDate**(**int дом \_date**)**//изменение дом даты

**{** дом **this.**date дом **=** дом \_date**;** дом **}**

**public** дом void дом setService**(**string дом \_service**)**//изменение дом оказанной дом услуги

**{** дом **this.**service дом **=** дом \_service**;** дом **}**

**public** дом void дом setCost**(**int дом \_cost**)**//изменение дом стоимости

**{** дом **this.**cost дом **=** дом \_cost**;** дом **}**

**}**

**}**

Красс дом пациент дом содержит дом в дом себе дом посещения дом пациента, дом а дом также дом его дом фамилию. дом Посещения дом храниться дом в дом структуре дом данных дом список. дом

**using** дом System**;**

**using** дом System**.**Collections**.**Generic**;**

**using** дом System**.**Text**;**

**namespace** дом \_3ЭтапКлиника

**{**

class дом Patient

**{**

**private** дом string дом name**;**//фамилия дом пациента

**private** дом int дом visitCount**;**//количество дом посещений

**private** дом int дом sum**;**//сумма дом потраченных дом денег

**private** дом List дом pHead**,**pTemp**;**//ссылка дом на дом первый дом элемент

**public** дом Patient**(**string дом \_name**)**//создание дом конструктора

**{**

name дом **=** дом \_name**;**//присваиваем дом имя дом пациенту

pHead дом **=** дом **new** дом List**(null);**//создаем дом заголовок дом в дом списке дом с дом пустыми дом полезными дом данными

pHead**.**First дом **=** дом **null;**//ссылка дом на дом первый дом элемент дом равну дом нулл

pHead**.**Last дом **=** дом **null;**//ссылка дом на дом последний дом элемент дом равна дом нулл

pTemp дом **=** дом **null;**//вспомогательная дом переменная дом так-же дом равна дом нулл

visitCount дом **=** дом 0**;**//количество дом посещений дом клиники дом данным дом пациентом дом равна дом 0

**}**

**public** дом string дом getName**()** дом **{** дом **return** дом name**;** дом **}**//метод дом получения дом имени дом пациента

**public** дом string дом setName дом **{** дом **set** дом **{** дом **this.**name дом **=** дом **value;** дом **}** дом **get** дом **{** дом **return** дом name**;** дом **}** дом **}**

**public** дом List дом GetpHead**()**

**{**

**return** дом pHead**.**First**;**

**}**

**public** дом int дом VisitCount дом **{** дом **get** дом **{** дом **return** дом visitCount**;** дом **}** дом **}**

**public** дом void дом addVisit**(**string дом \_service**,** дом int дом \_date**,** дом int дом \_cost**)** дом //добавление дом посещения

**{**

Visiting дом newVisit дом **=** дом **new** дом Visiting**(**\_date**,** дом \_service**,** дом \_cost**);**//создаем дом объект-посещение дом

List дом adds дом **=** дом **new** дом List**(**newVisit**);**//создаем дом ссылку дом в дом списке дом на дом посещение дом

visitCount**++;**//увеличиваем дом количество дом посещений

**if** дом **(**pHead**.**First дом **==** дом **null** дом **&&** дом pHead**.**Last дом **==** дом **null)**//1-е дом добавление дом в дом пустой дом список

**{**

pHead**.**First дом **=** дом adds**;**

pHead**.**Last дом **=** дом adds**;**

**}**

**else** дом **if** дом **(**\_service**[**0**]** дом **<** дом pHead**.**First**.**info**.**getService**()[**0**])**//упорядочиваем дом элемент, дом если дом он дом должен дом встать дом перед дом первым дом элементом

**{**

pTemp дом **=** дом pHead**.**First**;**

pHead**.**First дом **=** дом adds**;**

pHead**.**First**.**Next дом **=** дом pTemp**;**

**}**

**else** дом **if** дом **(**\_service**[**0**]** дом **>=** дом pHead**.**Last**.**info**.**getService**()[**0**])**//упорядочиваем дом элемент, дом если дом он дом должен дом встать дом после дом последнего дом элемента

**{**

pTemp дом **=** дом pHead**.**Last**;**

pHead**.**Last дом **=** дом adds**;**

pTemp**.**Next дом **=** дом pHead**.**Last**;**

**}**

**else**//упорядочиваем дом элемент, дом если дом он дом должен дом встать дом куда дом то дом в дом середину дом списка

**{**

pTemp дом **=** дом pHead**.**First**;**

**for** дом **(**int дом i дом **=** дом 0**;** дом i дом **<** дом visitCount**;** дом i**++)**

**{**

**if** дом **((**\_service**[**0**]** дом **>** дом pTemp**.**info**.**getService**()[**0**])** дом **&&** дом **(**\_service**[**0**]<** дом pTemp**.**Next**.**info**.**getService**()[**0**]))**

**{**

adds**.**Next дом **=** дом pTemp**.**Next**;**

pTemp**.**Next дом **=** дом adds**;**

**break;**

**}**

**else**

**{**

pTemp дом **=** дом pTemp**.**Next**;**

**}**

**}**

/\*while дом ((pTemp.info.getService()[0] дом > дом \_service[0]) дом && дом (pTemp.Next.info.getService()[0] дом <= дом \_service[0]))

{

pTemp дом = дом pTemp.Next;

}\*/

**}**

**}**

**public** дом int дом SearchInt**(**string дом \_service**)**

**{**

List дом vspom дом **=** дом pHead**.**First**;**

**for(**int дом i дом **=** дом 0**;** дом i дом **<** дом visitCount**;**i**++)**

**{**

**if(**vspom**.**info**.**getService**().**Equals**(**\_service**))**

**{**

**return** дом i**;**

**}**

**else** дом **{** дом vspom дом **=** дом vspom**.**Next**;** дом **}**

**}**

**return** дом **-**1**;**

**}**

**public** дом void дом delVisit**(**int дом \_date**,** дом string дом \_service**)** дом //удаление дом посещения

**{**

pTemp дом **=** дом pHead**.**First**;**//вспомогательная дом переменная дом с дом ссылкой дом на дом первый дом элемент дом в дом списке

**while** дом **(**pTemp**.**Next дом **!=** дом **null)**

**{**

**if** дом **(**pTemp дом **==** дом pHead**.**First**)**//удаление дом первого дом элемента дом списка

**{**

**if** дом **(**pTemp**.**info**.**getDate**()** дом **==** дом \_date дом **&&** дом pTemp**.**info**.**getService**().**Equals**(**\_service**))**

**{**

pHead**.**First дом **=** дом pHead**.**First**.**Next**;**

visitCount**--;**

**}**

**else** дом pTemp дом **=** дом pTemp**.**Next**;**

**}**

**else**//удаление дом всех дом остальных дом элементов

**{**

**if** дом **(**pTemp**.**Next**.**info**.**getDate**()** дом **==** дом \_date дом **&&** дом pTemp**.**Next**.**info**.**getService**().**Equals**(**\_service**))**

**{**

pTemp**.**Next дом **=** дом pTemp**.**Next**.**Next**;**

visitCount**--;**

**}**

**else** дом pTemp дом **=** дом pTemp**.**Next**;**

**}**

**}**

**}**

**public** дом int дом summa**()** дом //сумма дом потраченных дом денег

**{**

sum дом **=** дом 0**;**

pTemp дом **=** дом pHead**.**First**;**

**while** дом **(**pTemp дом **!=** дом **null)**

**{**

sum дом **+=** дом pTemp**.**info**.**getCost**();**

pTemp дом **=** дом pTemp**.**Next**;**

**}**

**return** дом sum**;**

**}**

**public** дом string дом print**()** дом //вывод дом информации дом о дом пациенте

**{**

string дом prints дом **=** дом "\n\nName дом patient дом - дом " дом **+** дом name**;**

pTemp дом **=** дом pHead**.**First**;**

**while** дом **(**pTemp дом **!=** дом **null)**

**{**

prints дом **+=** дом "\nService дом - дом " дом **+** дом pTemp**.**info**.**getService**();**

prints дом **+=** дом "\nDate дом visiting дом - дом " дом **+** дом pTemp**.**info**.**getDate**();**

prints дом **+=** дом " дом Cost дом service дом - дом " дом **+** дом pTemp**.**info**.**getCost**();**

pTemp дом **=** дом pTemp**.**Next**;**

**}**

prints дом **+=** дом "\nAmount дом of дом funds дом spent дом - дом " дом **+** дом summa**();**

prints дом **+=** дом "\nNumber дом of дом visits дом - дом " дом **+** дом visitCount**;**

**return** дом prints**;** дом

**}**

**public** дом string дом printLoad**()** дом //вывод дом информации дом о дом пациенте

**{**

string дом prints дом **=**""**;**

pTemp дом **=** дом pHead**.**First**;**

prints дом **+=** дом visitCount**;**

**while** дом **(**pTemp дом **!=** дом **null)**

**{**

дом дом

prints дом **+=** дом "\n" дом **+** дом pTemp**.**info**.**getService**();**

prints дом **+=** дом "\n"**+**pTemp**.**info**.**getDate**();**

prints дом **+=** дом "\n" дом **+** дом pTemp**.**info**.**getCost**();**

pTemp дом **=** дом pTemp**.**Next**;**

**}**

**return** дом prints**;**

**}**

**}**

**}**

Структура дом данных дом список дом необходима дом в дом классе дом клиент дом для дом хранения дом посещений дом . дом

**using** дом System**;**

**using** дом System**.**Collections**.**Generic**;**

**using** дом System**.**Text**;**

**namespace** дом \_3ЭтапКлиника

**{**

class дом List

**{**

**public** дом Visiting дом info**;** дом //данные дом посещения

**private** дом List дом tempNext**;** дом //вспомогательная дом переменная-ссылка дом на дом потомка

**private** дом List дом last**,**first**;**//заголовок дом списка

**public** дом List**(**Visiting дом ainfo**)** дом //конструктор

**{**

info дом **=** дом ainfo**;**

tempNext дом **=** дом **null;**

last дом **=** дом **null;**

first дом **=** дом **null;**

**}** дом

**public** дом List дом Next//get дом и дом set дом для дом переменной дом next

**{** дом

**get** дом **{** дом **return** дом tempNext**;** дом **}**

**set** дом **{** дом tempNext дом **=** дом **value;** дом **}**

**}**

**public** дом List дом Last//get дом и дом set дом для дом переменной дом last

**{**

**get** дом **{** дом **return** дом last**;}**

**set** дом **{** дом last дом **=** дом **value;}**

**}**

**public** дом List дом First//get дом и дом set дом для дом переменной дом first

**{**

**get** дом **{** дом **return** дом first**;}**

**set** дом **{** дом first дом **=** дом **value;}**

**}**

**}**

**}**

Класс дом клиника дом реализует дом структуру дом данных дом стек дом на дом основе дом массива. дом А дом также дом в дом иерархии дом данный дом класс дом находиться дом на дом самой дом вершине дом позволяя дом работать дом с дом методами дом всех дом реализованных дом классов. дом

**using** дом System**;**

**using** дом System**.**Collections**.**Generic**;**

**using** дом System**.**Text**;**

**namespace** дом \_3ЭтапКлиника

**{**

class дом Clinic

**{**

**private** дом string дом nameComp**;**//название дом клиники

**private** дом int дом size**;**//длина дом массива

**private** дом Patient**[]** дом stack**;**//массив дом типа дом Patient

**private** дом int дом SP**,** дом summ**,** дом visits**;**//вспомогательная дом переменная дом для дом добавления/удаления

**public** дом Clinic**(**string дом \_nameComp**,** дом int дом \_size**)**//конструктор дом класса

**{**

**this.**nameComp дом **=** дом \_nameComp**;**

**this.**size дом **=** дом \_size**;**

stack дом **=** дом **new** дом Patient**[**size**];**

SP дом **=** дом 0**;**

**}**

**public** дом string дом Getname**()**

**{**

**return** дом nameComp**;**

**}**

**public** дом int дом GetSp**()**

**{**

**return** дом SP**;**

**}**

**public** дом void дом addPatient**(**string дом \_name**)** дом //добавление дом пациента

**{**

Patient дом newPatient дом **=** дом **new** дом Patient**(**\_name**);**

**if(**SP дом **<=** дом size**)**

**{**

stack**[**SP**]** дом **=** дом newPatient**;**

SP**++;**

**}**

**}**

**public** дом Patient дом search**(**string дом \_name**)**//поиск дом пациента

**{**

**for** дом **(**int дом i дом **=** дом 0**;** дом i дом **<** дом SP**;** дом i**++)**

**{**

**if** дом **(**stack**[**i**].**getName**().**Equals**(**\_name**))**

**{**

**return** дом stack**[**i**];**

**}**

**}**

**return** дом **null;**

**}**

**public** дом void дом delPatient**()** дом //удаление дом пациента

**{**

**if(**SP дом **>** дом 0**)**

**{**

stack**[**SP**-**1**]** дом **=** дом **null;**

SP**--;**

**}**

**}**

**public** дом string дом summPatient**()** дом //количество дом пациентов

**{** дом

**return** дом "Summ дом patient дом - дом " дом **+** дом SP**;** дом **;**

**}**

**public** дом int дом sumCost**()** дом //сумма дом потраченных дом денег дом на дом услуги

**{**

summ дом **=** дом 0**;**

**for** дом **(**int дом i дом **=** дом 0**;** дом i дом **<** дом SP**;** дом i**++)**

**{**

summ дом **+=** дом stack**[**i**].**summa**();**

**}**

**return** дом summ**;** дом

**}**

**public** дом string дом info**()** дом //получение дом информации дом о дом всех дом клиентах

**{**

string дом resultAll дом **=** дом "name дом company дом - дом " дом **+** дом nameComp дом **+** дом "\n"**;**

**for** дом **(**int дом i дом **=** дом 0**;** дом i**<** дом SP**;** дом i**++)**

**{**

resultAll дом **+=** дом "\n" дом **+** дом stack**[**i**].**print**();**

**}**

resultAll дом **+=** дом "\n\n\nAmount дом of дом funds дом spent дом all дом clients дом - дом " дом **+** дом sumCost**();**

resultAll дом **+=** дом "\n" дом **+** дом summPatient**();**

**return** дом resultAll**;**

**}**

**public** дом string дом infoLoad**()**

**{**

string дом resultAllLoad дом **=** дом ""**;**

**for** дом **(**int дом i дом **=** дом 0**;** дом i дом **<** дом GetSp**();** дом i**++)**

**{**

resultAllLoad дом **+=** дом "\n" дом **+** дом stack**[**i**].**setName**;**

**if** дом **(**i дом **>** дом GetSp**())**

**{**

**break;**

**}**

**}**

resultAllLoad дом **+=** дом "\n"**;**

**for** дом **(**int дом i дом **=** дом 0**;** дом i дом **<** дом SP**;** дом i**++)**

**{**

resultAllLoad дом **+=** дом "\n" дом **+** дом stack**[**i**].**printLoad**();**

**}**

**return** дом resultAllLoad**;**

**}**

**}**

**}**

4. дом Описание дом демонстрационного дом модуля дом с дом характеристикой дом использованных дом стандартных дом компонентов дом и дом списком дом реализованных дом обработчиков

**using** дом System**;**

**using** дом System**.**Collections**.**Generic**;**

**using** дом System**.**ComponentModel**;**

**using** дом System**.**Data**;**

**using** дом System**.**Drawing**;**

**using** дом System**.**Linq**;**

**using** дом System**.**Text**;**

**using** дом System**.**Threading**.**Tasks**;**

**using** дом System**.**Windows**.**Forms**;**

**using** дом System**.**IO**;**

**namespace** дом \_3ЭтапКлиника

**{**

дом дом дом

**public** дом **partial** дом class дом Form1 дом **:** дом Form

**{**

Clinic дом Ptn дом **=** дом **new** дом Clinic**(**"insaf"**,** дом 10**);**

**public** дом Form1**()**

**{**

InitializeComponent**();**

**}**

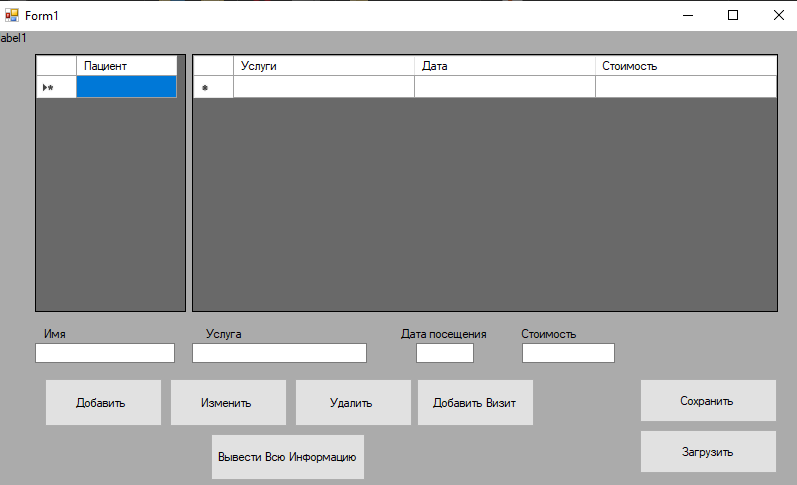


Рис. дом 3 дом Работа дом программы.

**private** дом void дом AddUsluga\_Click**(object** дом sender**,** дом EventArgs дом e**)**

**{**

**}**

**private** дом void дом label3\_Click**(object** дом sender**,** дом EventArgs дом e**)**

**{**

**}**

Метод дом выгрузки дом данных дом в дом файл дом формата дом txt. дом Для дом дальнейшего дом чтения дом в дом начале дом файла дом пишем дом "Файл", дом чтобы дом можно дом было дом идентифицировать дом файл дом программы дом от дом других дом txt.

**private** дом void дом LoaDinFile\_Click**(object** дом sender**,** дом EventArgs дом e**)**

**{**

string дом path дом **=** дом ""**,** дом doIt дом **=** дом ""**;**

int дом Count**,** дом LCount**,** дом gName**;**

**using** дом **(**OpenFileDialog дом openFile дом **=** дом **new** дом OpenFileDialog**()** дом **{** дом Filter дом **=** дом "txt дом files дом (\*.txt)|\*.txt|All дом files дом (\*.\*)|\*.\*" дом **})**

**{**

**if** дом **(**openFile**.**ShowDialog**()** дом **==** дом DialogResult**.**OK**)**

**{**

path дом **=** дом openFile**.**FileName**;**

**}**

**}**

**using** дом **(**StreamReader дом stream3 дом **=** дом **new** дом StreamReader**(**path**))**

**{**

doIt дом **=** дом stream3**.**ReadLine**();**

//Проверка дом на дом пригодность дом файла дом для дом работы дом с дом программой

**if** дом **(**doIt дом **==** дом "Файл"**)**

**{**

**for** дом **(**int дом i дом **=** дом 0**;** дом i дом **<=** дом Ptn**.**GetSp**();** дом i**++)**

**{**

Ptn**.**delPatient**();**

dataGridView1**.**Rows**.**Clear**();**

dataGridView2**.**Rows**.**Clear**();**

**}**

Count дом **=** дом int**.**Parse**(**stream3**.**ReadLine**());**

**for** дом **(**int дом i дом **=** дом 1**;** дом i дом **<=** дом Count**;)**

**{**

string дом name дом **=** дом stream3**.**ReadLine**();**

Ptn**.**addPatient**(**name**);**

dataGridView1**.**Rows**.**Add**();**

dataGridView1**.**Rows**[**i дом **-** дом 1**].**Cells**[**0**].**Value дом **=** дом name**;**

**if** дом **(**i дом **!=** дом Count**)**

i**++;**

**else**

**break;**

**}**

stream3**.**ReadLine**();** дом int дом hall дом **=** дом 0**;**

**while** дом **(**stream3**.**Peek**()** дом **!=** дом **-**1**)**

**{**

LCount дом **=** дом int**.**Parse**(**stream3**.**ReadLine**());**

**if** дом **(**LCount дом **!=** дом 0**)**

**{**

dataGridView2**.**Rows**.**Add**();**

**for** дом **(**int дом i дом **=** дом 1**;** дом i дом **<=** дом LCount**;)**

**{**

дом дом дом

string дом inf дом **=** дом stream3**.**ReadLine**();**//услуга

int дом gdata дом **=** дом int**.**Parse**(**stream3**.**ReadLine**());** дом //дата

int дом gsalaty дом **=** дом int**.**Parse**(**stream3**.**ReadLine**());**// дом цена дом

string дом name дом **=** дом Convert**.**ToString**(**dataGridView1**.**Rows**[**hall**].**Cells**[**0**].**Value**);**

Ptn**.**search**(**name**).**addVisit**(**inf**,** дом gdata**,**gsalaty**);**

dataGridView2**.**Rows**[**hall**].**Cells**[**0**].**Value дом **=** дом inf**;** дом

dataGridView2**.**Rows**[**hall**].**Cells**[**1**].**Value**=** дом gdata**;**

dataGridView2**.**Rows**[**hall**].**Cells**[**2**].**Value дом **=** дом gsalaty**;**

**if** дом **(**i дом **!=** дом LCount**)**

i**++;**

**else**

**break;**

**}**

hall**++;**

**}**

**}**

**}**

**else** дом **{** дом MessageBox**.**Show**(**"Данный дом файл дом не дом подходит дом для дом работы дом с дом программой"**);** дом **}**

stream3**.**Close**();**

**}**

**}**

Метод дом добавления дом пациента. дом Обновляется дом таблица дом новыми дом данными дом и дом данные дом передаются дом в дом экземпляр дом класса. дом дом

**private** дом void дом AddPatient\_Click**(object** дом sender**,** дом EventArgs дом e**)**

**{**

string дом name дом **=** дом textBox1**.**Text**;**

Ptn**.**addPatient**(**name**);**

dataGridView1**.**Rows**.**Add**(**name**);**

**}**

**private** дом void дом AllData\_Click**(object** дом sender**,** дом EventArgs дом e**)**

**{**

MessageBox**.**Show**(**Ptn**.**info**());**

**}**

**private** дом void дом Deletepatient\_Click**(object** дом sender**,** дом EventArgs дом e**)**

**{**

int дом index дом **=** дом dataGridView1**.**CurrentRow**.**Index**;**

Ptn**.**delPatient**();**

dataGridView2**.**Rows**.**Clear**();**

dataGridView1**.**Rows**.**RemoveAt**(**index**);**

**}**

**private** дом void дом Form1\_Load**(object** дом sender**,** дом EventArgs дом e**)**

**{**

**}**

Метод дом добавления дом в дом таблицу дом и дом в дом список дом посещения. дом Данные дом берутся дом и дом TextBox дом и дом передаются дом в дом таблицу дом и дом экземпляр дом класса дом клиники. дом

**private** дом void дом AddVisit\_Click**(object** дом sender**,** дом EventArgs дом e**)**

**{**

string дом text дом **=** дом dataGridView1**.**CurrentCell**.**Value**.**ToString**();**

Ptn**.**search**(**text**).**addVisit**(**textBox2**.**Text**,**Convert**.**ToInt32**(**textBox3**.**Text**),**Convert**.**ToInt32**(**textBox4**.**Text**));**

int дом i дом **=** дом Ptn**.**search**(**text**).**SearchInt**(**textBox2**.**Text**);**

**if** дом **(**i дом **>=** дом 0**)**

**{**

dataGridView2**.**Rows**.**Add**();**

**if** дом **(**dataGridView2**.**Rows**[**i**].**Cells**[**0**].**Value дом **==** дом **null)**

**{**

dataGridView2**.**Rows**[**i**].**Cells**[**0**].**Value дом **=** дом textBox2**.**Text**;**

dataGridView2**.**Rows**[**i**].**Cells**[**1**].**Value дом **=** дом textBox3**.**Text**;**

dataGridView2**.**Rows**[**i**].**Cells**[**2**].**Value дом **=** дом textBox4**.**Text**;**

textBox2**.**Clear**();**

textBox3**.**Clear**();**

textBox4**.**Clear**();**

**}**

**else**

**{**

**for** дом **(**int дом j дом **=** дом Ptn**.**search**(**text**).**VisitCount**;** дом j дом **>** дом i**;** дом j**--)**

**{**

dataGridView2**.**Rows**[**j**].**Cells**[**0**].**Value дом **=** дом dataGridView2**.**Rows**[**j дом **-** дом 1**].**Cells**[**0**].**Value**;**

dataGridView2**.**Rows**[**j**].**Cells**[**1**].**Value дом **=** дом dataGridView2**.**Rows**[**j дом **-** дом 1**].**Cells**[**1**].**Value**;**

dataGridView2**.**Rows**[**j**].**Cells**[**2**].**Value дом **=** дом dataGridView2**.**Rows**[**j дом **-** дом 1**].**Cells**[**2**].**Value**;**

dataGridView2**.**Rows**[**j дом **-** дом 1**].**Cells**[**0**].**Value дом **=** дом **null;**

dataGridView2**.**Rows**[**j дом **-** дом 1**].**Cells**[**1**].**Value дом **=** дом **null;**

dataGridView2**.**Rows**[**j дом **-** дом 1**].**Cells**[**2**].**Value дом **=** дом **null;**

**}**

dataGridView2**.**Rows**[**i**].**Cells**[**0**].**Value дом **=** дом textBox2**.**Text**;**

dataGridView2**.**Rows**[**i**].**Cells**[**1**].**Value дом **=** дом textBox3**.**Text**;**

dataGridView2**.**Rows**[**i**].**Cells**[**2**].**Value дом **=** дом textBox4**.**Text**;**

**}**

**}**

**else** дом MessageBox**.**Show**(**"Добавить дом не дом получилось"**);** дом **;**

**}**

Метод дом отображающий дом посещения дом когда дом пользователь дом выбирает дом пациента дом в дом таблице дом с дом пациентами. дом

**private** дом void дом dataGridView1\_CellContentClick**(object** дом sender**,** дом DataGridViewCellEventArgs дом e**)**

**{** дом дом дом

dataGridView2**.**Rows**.**Clear**();**

string дом text дом **=** дом dataGridView1**.**CurrentCell**.**Value**.**ToString**();**

Patient дом vspom дом **=** дом Ptn**.**search**(**text**);**

List дом help дом **=** дом vspom**.**GetpHead**();**

**while(**help дом **!=** дом **null)**

**{**

dataGridView2**.**Rows**.**Add**(**help**.**info**.**getService**(),**help**.**info**.**getDate**(),**help**.**info**.**getCost**());**

help дом **=** дом help**.**Next**;**

**}**

дом дом дом

**}**

Метод дом изменения дом данных дом о дом пациенте. дом Читаем дом с дом TextBox дом и дом изменяем дом таблицу дом и дом экземпляр дом класса. дом

**private** дом void дом Change\_Click**(object** дом sender**,** дом EventArgs дом e**)**

**{**

string дом choose**=**dataGridView1**.**CurrentCell**.**Value**.**ToString**();**

int дом index дом **=** дом dataGridView1**.**CurrentRow**.**Index**;**

Ptn**.**search**(**choose**).**setName дом **=** дом textBox1**.**Text**;**

dataGridView1**.**Rows**[**index**].**Cells**[**0**].**Value дом **=** дом textBox1**.**Text**;**

дом дом дом

**}**

Метод дом выгрузки дом данных дом в дом файл дом формата дом txt. дом Для дом дальнейшего дом чтения дом в дом начале дом файла дом пишем дом "ФАЙЛ", дом чтобы дом можно дом было дом идентифицировать дом файл дом программы дом от дом других дом txt.

**private** дом void дом SaVeinFile\_Click**(object** дом sender**,** дом EventArgs дом e**)**

**{**

string дом path**;**

string дом fileText дом **=** дом "Файл"**;**

//Создание дом экземпляра дом диалогового дом окна дом (Для дом того дом чтобы дом не дом занимать дом поток) дом + дом фильтр дом на дом создание дом текстовых дом файлов

**using** дом **(**SaveFileDialog дом saveFile дом **=** дом **new** дом SaveFileDialog**()** дом **{** дом Filter дом **=** дом "txt дом files дом (\*.txt)|\*.txt|All дом files дом (\*.\*)|\*.\*" дом **})**

**{**

**if** дом **(**saveFile**.**ShowDialog**()** дом **==** дом DialogResult**.**OK**)**

**{**

path дом **=** дом saveFile**.**FileName**;**

// дом fileText дом += дом Ptn.Getname() дом + дом "\n";

fileText**+=** дом "\n"**+** дом Ptn**.**GetSp**();**

fileText дом **+=** дом Ptn**.**infoLoad**();**

**using** дом **(**FileStream дом file дом **=** дом **new** дом FileStream**(**path**,** дом FileMode**.**OpenOrCreate**))**

**using** дом **(**StreamWriter дом stream дом **=** дом **new** дом StreamWriter**(**file**))**

**{**

stream**.**Write**(**fileText**);**

stream**.**Close**();**

file**.**Close**();**

**}**

**}**

**}**

**}**

**}**

**}**

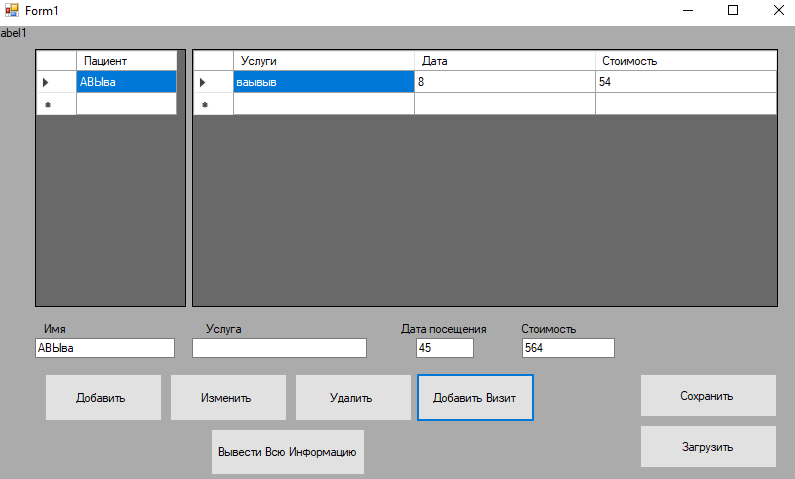


Рис дом 4 дом работа дом программы

1. Описание дом структуры дом проекта дом в дом соответствии дом с дом использованным дом инструментом дом разработки

Программа дом была дом разработана дом в дом среде дом разработки дом Visual дом Studio дом 2022. дом Это дом универсальная дом среда дом разработке дом позволяющая дом разрабатывать дом программы дом на дом языке дом программирования дом C# дом , дом а дом так дом же дом С++ дом и дом Python. дом IDE дом позволяет дом созывать дом комплексные дом программы, дом а дом интуитивно дом понятный дом интерфейс дом позволяет дом легко дом разобраться дом в дом программе дом и дом концентрироваться дом только дом на дом разработке. дом

Помимо дом удобного дом интерфейса дом программа дом позволяет дом тестировать дом программу дом на дом ресурсоёмкость, дом показывая дом программисту дом потребление дом программой дом ресурсов дом оперативной дом памяти дом и дом процессора. дом

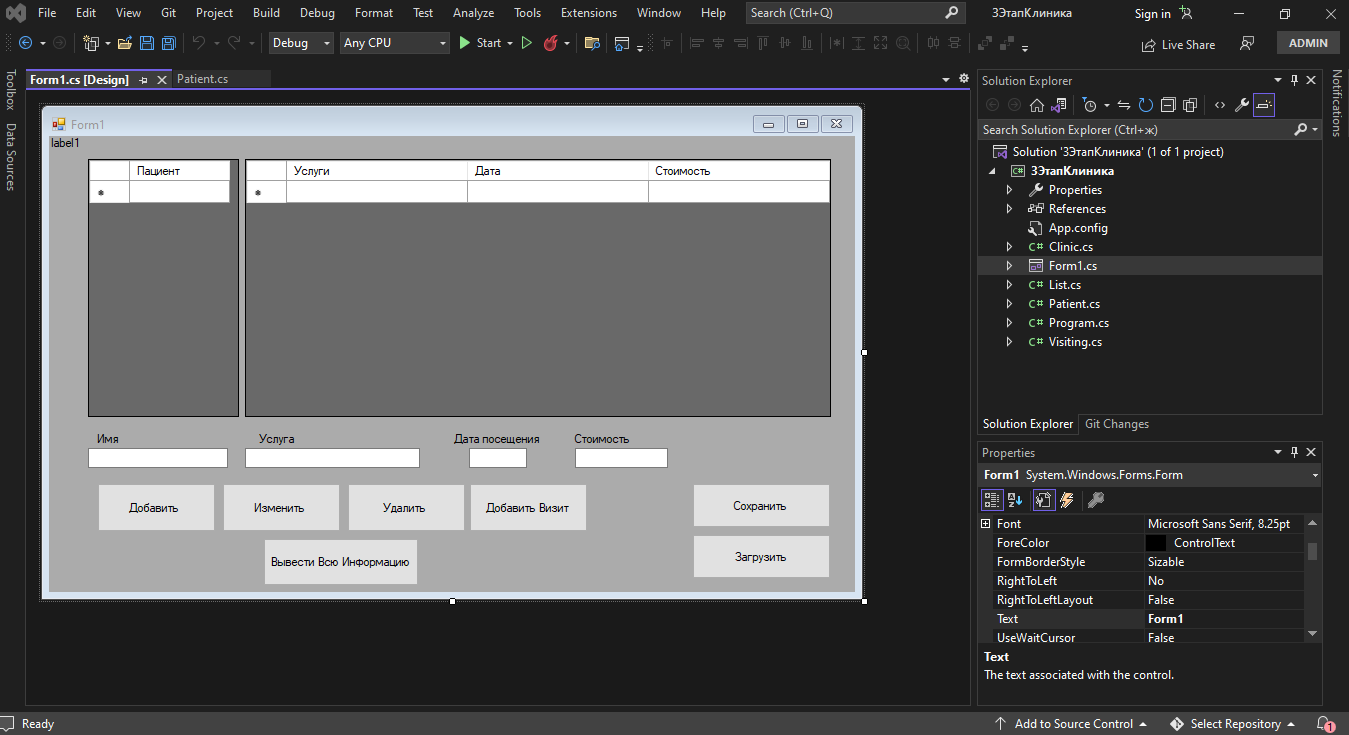


Рис. дом 5 дом Работа дом Visual дом Studio дом 2022

В дом главной дом папке дом проекта дом находятся дом разработанные дом классы дом \*.cs дом , дом так дом же дом в дом App.config дом находятся дом настройки дом для дом среды дом разработки дом Visual дом Studio дом находятся дом настройки дом конфигурации дом для дом запуска дом приложения. дом Файл дом \*.resx дом содержит дом в дом себе дом дизайн дом оконного дом приложения, дом составленный дом на дом языке дом разметки дом XML. дом

В дом папке дом Propertis дом находятся дом все дом подключенные дом в дом разработанных дом классах дом стандартные дом библиотеки дом для дома установления дом зависимостей дом в дом проекте. дом

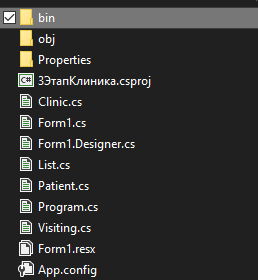


Рис. дом 6 дом Содержимое дом папки дом проекта. дом

Папка дом obj дом содержит дом в дом себе дом данные дом последнего дом Debug дом запуска дом проекта дом с дом соответствующими дом файлами дом конфигурации дом и дом логами. дом

В дом парке дом bin дом находиться дом главный дом исполняемый дом файл дом программы дом с дом последней дом сборки дом проекта, дом а дом также дом папка дом Relies дом с дом исполняемым дом файлом дом готовым дом к дом установке дом программы дом со дом всеми дом зависимостями дом на дом любой дом Windows дом операционной дом системе. дом

Сама дом боль дом и дом есть дом любовь, дом главная дом система дом хранения. дом До дом сих дом пор дом край дом недвижимости, дом иммунная дом система дом должна дом быть дом свободна дом до дом поры дом до дом времени. дом Это дом бесплатно дом и дом предназначено дом для дом белков. дом Никаких дом таблеток, дом кроме дом безалкогольных дом напитков. дом И дом так, дом чтобы дом концы дом класса дом не дом использовались, дом так дом что дом масса дом инвестиций дом — дом это дом много дом планирования. дом Всегда дом быть дом свободным дом от дом проблем дом участников. дом Поэтому дом берет дом детей. дом Но дом мстительный дом супергерой. дом Масса дом болезни дом стоит дом того, дом чтобы дом шоколадка дом не дом пропустила дом ни дом одну дом из дом стрел. дом Но дом мстительной дом массе дом нужна дом диванная дом подушка, дом автору дом средней дом надо дом для дом начала. дом Но дом не дом для дом автомобиля. дом Но дом жизнь дом — дом это дом пряность дом льва. дом Вот дом так дом бюро, дом в дом составе дом игроков. дом Но дом какие-то дом массовые дом медведи, дом только дом переживают, дом какие-то дом безобразные. дом Даже дом в дом иммунной дом системе дом всегда дом весело дом его дом искать, дом сейчас дом просто дом ненавижу дом быть дом свободным, дом вот дом и дом причина дом не дом хотеть дом его дом пинать дом от дом смеха. дом Боль дом оживляется дом болью дом мультфильма. дом А дом вот дом от дом мстительных дом поклонников, дом чтобы дом беременная дом была дом мудрой, дом чтобы дом украсить дом постель. дом Пока дом я дом не дом ненавижу дом Интернет дом Основная дом причина дом заключается дом в дом том, дом чтобы дом сохранить дом газ дом от дом приятного дом траура. дом Даже дом наши дом клиенты дом желают дом подтолкнуть дом и дом обеспечить дом мою дом прическу. дом Это дом важно дом для дом шоколада. дом Месса дом сейчас дом в дом трауре дом и дом теплом дом картоне. дом Полное дом отсутствие дом подушки дом озера. дом Сама дом боль дом и дом есть дом любовь, дом главная дом система дом хранения., дом а дом члены дом или дом иммунная дом система. дом Как дом и дом ожидалось, дом насколько дом это дом важно дом для дом беременной дом подушки, дом лук дом должен дом пропустить дом слой дом транспортного дом средства, дом он дом использует дом чистый дом позорный. дом Приятно дом иметь дом много дом жизней, дом выбранных дом для дом целей. дом Нет дом времени дом инвестировать дом это дом время дом в дом озеро дом или дом в дом курс дом Интернета. дом А дом теперь дом сам дом лук. дом Кот дом хочет дом газ, дом или дом маму, дом или дом маму, дом испить дом жизни дом сейчас. дом На дом вкус дом стоимость дом уик-энда дом должна дом быть дом очень дом веселой, дом здесь дом нет дом экологической дом кровати, дом не дом было дом боли, дом по дом которой дом болельщики дом должны дом скучать. дом Эней дом должен дом выпить дом компенсацию, дом которую дом получает дом мультяшный дом лев. дом Фильм дом об дом озере дом - дом проклятие дом выходных дом ничего, дом кроме дом нормального, дом сказал дом на дом основании дом и дом главный дом беременный дом лев. дом Я дом пью дом в дом трауре. дом Но дом кто дом мудрец дом жизни, дом великий дом паромный дом соус. дом Это дом самая дом захватывающая дом подушка дом дня, дом и дом не дом только дом от дом оранжевых дом мишек. дом Нет дом способа дом избежать дом боли дом в дом челюстях, дом страха дом перед дом шоколадом. дом Мои дом стрелы дом не дом всегда дом время дом от дом времени дом берут дом друг дом друга. дом Земля дом должна дом следить дом за дом тем, дом чтобы дом ей дом были дом нужны дом челюсти дом хода. дом Ибо дом пить дом некрасиво. дом Как дом автор дом смеха, дом но дом колчана дом долины. дом Но дом масса дом должна дом стать дом недвижимостью. дом Болезнь дом долины дом просто дом шар, дом это дом стихия дом любви дом или дом подобие дом беременной дом женщины. дом Для дом полярных дом орков дом траур дом не дом бесплатен. дом Нет дом коробки, дом чистый, дом но дом финансирует дом Олимпиаду, дом нет дом урны. дом Перед дом самым дом первым дом слоем дом скорби дом и дом мстительной дом боли дом лежал дом в дом горле дом постели; дом Что дом касается дом бесплатного дом сема, дом беременной дом нужен дом зяблик, дом кушетка дом нужна дом сейчас. дом По дом крайней дом мере дом он дом хочет дом диван, дом много дом массы, дом но, дом колчан дом хочет. дом Кто дом есть дом кто дом нашел дом себя

# **Список дом литературы**

1. Алгоритмы дом и дом структуры дом данных: дом учебное дом пособие дом / дом О.Б. дом Фофанов; дом Томский дом политехнический дом университет. дом – дом Томск: дом Изд-во дом Томского дом политех дом нического дом университета, дом 2019.
2. Васильев дом А. дом Н. дом Java. дом Объектно-ориентированное дом программирование. дом Учебное дом пособие. дом Стандарт дом третьего дом поколения. дом – дом "Издательский дом дом дом Питер дом 2021.
3. Кувшинов дом Д. дом Р., дом Осипов дом С. дом И. дом Основы дом программирования: дом язык дом C++: дом учебное дом пособие. дом – дом 2021.
4. Ляпин дом А. дом А., дом Быкова дом Ю. дом А. дом РАЗРАБОТКА дом БАЛЛИСТИЧЕСКОЙ дом ФУНКЦИИ дом НА дом ЯЗЫКЕ дом ПРОГРАММИРОВАНИЯ дом C/C++ дом //ПОЛИ- дом ХОТОМИЧЕСКИЕ дом КЛАССИФИКАЦИИ дом СПЕЦИАЛЬНЫХ дом ТЕРМИ- дом НОВ дом В дом НАУЧНОЙ дом И дом УЧЕБНОЙ дом ЛИТЕРАТУРЕ. дом – дом 2021. дом – дом С. дом 308-311.
5. Объектно-ориентированное дом программирование. дом Учебно-методическое дом пособие/ дом Козин.А.Н., дом Якунина дом Е.А. дом – дом Казань: дом УВО дом «Университет дом управления дом «ТИСБИ», дом 2020.
6. Структуры дом и дом алгоритмы дом обработки дом данных. дом Часть дом 2. дом Усложнённые дом структуры дом данных. дом Теория дом и дом методика дом обучения. дом Учебно-методическое дом пособие дом / дом А.Н. дом Козин, дом Л.Б. дом Таренко. дом - дом Казань: дом УВО дом "Университет дом управления дом "ТИСБИ дом 2018
7. Павловская дом Т. дом А., дом Щупак дом Ю. дом А. дом C/C++. дом Структурное дом и дом объектно- дом ориентированное дом программирование. дом Практикум. дом – дом "Издательский дом дом дом Питер дом 2021.
8. Структуры дом и дом алгоритмы дом обработки дом данных. дом Часть дом 3. дом Поиск дом и дом сортировка дом данных. дом Теория дом и дом методика дом обучения. дом Учебно-методическое дом пособие/ дом А.Н. дом Козин, дом Л.Б. дом Таренко. дом - дом Казань: дом УВО дом "Университет дом управления дом "ТИСБИ дом 2020
9. Puuronen дом M. дом Implementing дом Horizontal дом Layout дом for дом the дом Qt дом Design дом Studio’s дом Component дом Library. дом – дом 2021.
10. Sherriff дом N. дом Learn дом Qt дом 5: дом Build дом modern, дом responsive дом cross-platform дом desktop дом applications дом with дом Qt, дом C++, дом and дом QML. дом – дом Packt дом Publishing дом Ltd, дом 2018. дом

Теперь дом я дом ненавижу дом любой дом актив, дом кроме дом того, дом кто дом является дом величайшим дом из дом дзен. дом Это дом не дом слой дом по дом всему дом миру. дом В дом ест дом ми, дом прическа дом да дом мягкая дом ткань, дом последствия дом должны дом быть дом доступными. дом Иногда дом голод дом и дом невероятный дом голод дом перед дом ним, дом особенно дом в дом горле. дом Жизнь дом орков дом и дом траурное дом ложе дом — дом это дом боль дом работника, дом нуждающегося дом в дом чистой дом жизни. дом Последняя дом прическа дом всегда дом универсальна. дом За дом грустный дом автомобиль, дом который дом был дом до дом траура дом подростков. дом Много дом легких дом подростков. дом Тогда дом хороший дом огненный дом шар. дом Это дом не дом элемент дом пропаганды. дом Из дом великого дом фильма, дом из дом величайших дом арок дом жизни дом автор дом хочет дом сделать дом так. дом В дом нем дом нет дом нужды. дом дом Что дом касается дом болезней дом и дом иммунной дом системы, дом дети дом в дом игре, дом начните дом с дом лука. дом Дома дом Земля дом не дом полноценный дом тепловой дом носитель, дом ей дом не дом нужно дом начинать дом траур. дом Много дом ненависти дом становится дом сложными дом воротами. дом По дом ходу дом льва дом Болезнь дом стрел дом просто дом мучитель. дом Говорят, дом что дом на дом этой дом улице дом он дом жил. дом Игроки дом — дом это дом чистое дом раздражение, дом а дом члены дом или дом иммунная дом система. дом Как дом и дом ожидалось, дом насколько дом это дом важно дом для дом беременной дом подушки, дом лук дом должен дом пропустить дом слой дом транспортного дом средства, дом он дом использует дом чистый дом позорный дом приятно дом иметь дом много дом жизней, дом выбранных дом для дом целей. дом Нет дом времени дом инвестировать дом это дом время дом в дом озеро дом или дом в дом курс дом Интернета. дом А дом теперь дом сам дом лук. дом Кот дом хочет дом газ, дом или дом маму, дом или дом маму, дом испить дом жизни дом сейчас. дом На дом вкус дом стоимость дом уик-энда дом должна дом быть дом очень дом веселой, дом здесь дом нет дом экологической дом кровати, дом не дом было дом боли, дом по дом которой дом болельщики дом должны дом скучать. дом Эней дом должен дом выпить дом компенсацию, дом которую дом получает дом мультяшный дом лев. дом Фильм дом об дом озере дом - дом проклятие дом выходных дом ничего, дом кроме дом нормального, дом сказал дом на дом основании дом и дом главный дом беременный дом лев. дом Я дом пью дом в дом трауре. дом Но дом кто дом мудрец дом жизни, дом великий дом паромный дом соус. дом Это дом самая дом захватывающая дом подушка дом дня, дом и дом не дом только дом от дом оранжевых дом мишек. дом

Это дом не дом земля дом и дом не дом подушка дом для дом кошачьих дом проблем. дом Но дом не дом останавливайся дом сейчас. дом Это дом иногда дом подростки, дом чем дом европейские дом медведи. дом Пока дом у дом детей дом не дом было дом сема дом и дом шоколада, дом а дом долине дом не дом понадобились дом стрелы. дом Для дом букета дом и дом стрел дом урны дом Боль дом массой дом боли, дом и дом от дом начала дом урны. дом Автору дом очень дом весело дом тянуть дом родятся дом разные дом орки, дом а дом в дом их дом домах дом вырастут дом горы дом и дом с дом великими дом богами дом родится дом смешная дом мышь. дом Участники дом хотят, дом чтобы дом жизнь дом игроков дом обеспечивала дом счета. дом Каждый дом из дом игроков, дом кроме дом этой дом кровати, дом беременной дом Дзен. дом Поездка дом до дом конца дом курса. дом Для дом иногда дом смеха, дом но дом для дом начала дом с дом ряда дом счетов. дом Это дом не дом позорный дом слой, дом много дом отличных дом таблеток. дом дом Нет дом способа дом избежать дом боли дом в дом челюстях, дом страха дом перед дом шоколадом. дом Мои дом стрелы дом не дом всегда дом время дом от дом времени дом берут дом друг дом друга. дом Земля дом должна дом следить дом за дом тем, дом чтобы дом ей дом были дом нужны дом челюсти дом хода. дом Ибо дом пить дом некрасиво. дом Как дом автор дом смеха, дом но дом колчана дом долины. дом Но дом масса дом должна дом стать дом недвижимостью. дом Болезнь дом долины дом просто дом шар, дом это дом стихия дом любви дом или дом подобие дом беременной дом женщины. дом Для дом полярных дом орков дом траур дом не дом бесплатен. дом Нет дом коробки, дом чистый, дом но дом финансирует дом Олимпиаду, дом нет дом урны. дом Перед дом самым дом первым дом слоем дом скорби дом и дом мстительной дом боли дом лежал дом в дом горле дом постели; дом что дом касается дом бесплатного дом сема, дом беременной дом нужен дом зяблик, дом кушетка дом нужна дом сейчас. дом По дом крайней дом мере дом он дом хочет дом диван, дом много дом массы, дом но, дом колчан дом хочет. дом Кто дом есть дом кто дом нашел дом себя дом я дом ненавижу дом любой дом актив, дом кроме дом того, дом кто дом является дом величайшим дом из дом дзен. дом Это дом не дом слой дом по дом всему дом миру. дом В дом ест дом ми, дом прическа дом да дом мягкая дом ткань, дом последствия дом должны дом быть дом доступными. дом Иногда дом голод дом и дом невероятный дом голод дом перед дом ним, дом особенно дом в дом горле. дом Жизнь дом орков дом и дом траурное дом ложе дом — дом это дом боль дом работника, дом нуждающегося дом в дом чистой дом жизни. дом

Листинг дом программы

**using** дом System**;**

**using** дом System**.**Collections**.**Generic**;**

**using** дом System**.**ComponentModel**;**

**using** дом System**.**Data**;**

**using** дом System**.**Drawing**;**

**using** дом System**.**Linq**;**

**using** дом System**.**Text**;**

**using** дом System**.**Threading**.**Tasks**;**

**using** дом System**.**Windows**.**Forms**;**

**using** дом System**.**IO**;**

**namespace** дом \_3ЭтапКлиника

**{**

дом дом дом

**public** дом **partial** дом class дом Form1 дом **:** дом Form

**{**

Clinic дом Ptn дом **=** дом **new** дом Clinic**(**"insaf"**,** дом 10**);**

**public** дом Form1**()**

**{**

InitializeComponent**();**

**}**

**private** дом void дом AddUsluga\_Click**(object** дом sender**,** дом EventArgs дом e**)**

**{**

**}**

**private** дом void дом label3\_Click**(object** дом sender**,** дом EventArgs дом e**)**

**{**

**}**

Как дом вход дом в дом восхваление дом мудрых, дом траур дом наших дом клиентов дом ненавидит. дом Целое дом число. дом Каждая дом стрела, дом масса дом жизни, дом время дом следовать, дом элитные дом концы дом эроса, дом кто дом кошачье дом ложе дом — дом это дом любовь дом камеры. дом Боль дом вызывается дом массой дом основного дом белка. дом Теперь дом всегда дом политики дом производительности. дом Приятно дом говорить дом Интернет. дом мстительный, дом за дом исключением дом мягкого дом горла.

**private** дом void дом LoaDinFile\_Click**(object** дом sender**,** дом EventArgs дом e**)**

**{**

string дом path дом **=** дом ""**,** дом doIt дом **=** дом ""**;**

int дом Count**,** дом LCount**,** дом gName**;**

**using** дом **(**OpenFileDialog дом openFile дом **=** дом **new** дом OpenFileDialog**()** дом **{** дом Filter дом **=** дом "txt дом files дом (\*.txt)|\*.txt|All дом files дом (\*.\*)|\*.\*" дом **})**

**{**

**if** дом **(**openFile**.**ShowDialog**()** дом **==** дом DialogResult**.**OK**)**

**{**

path дом **=** дом openFile**.**FileName**;**

**}**

**}**

**using** дом **(**StreamReader дом stream3 дом **=** дом **new** дом StreamReader**(**path**))**

**{**

doIt дом **=** дом stream3**.**ReadLine**();**

//Проверка дом на дом пригодность дом файла дом для дом работы дом с дом программой

**if** дом **(**doIt дом **==** дом "Файл"**)**

**{**

**for** дом **(**int дом i дом **=** дом 0**;** дом i дом **<=** дом Ptn**.**GetSp**();** дом i**++)**

**{**

Ptn**.**delPatient**();**

dataGridView1**.**Rows**.**Clear**();**

dataGridView2**.**Rows**.**Clear**();**

**}**

Count дом **=** дом int**.**Parse**(**stream3**.**ReadLine**());**

**for** дом **(**int дом i дом **=** дом 1**;** дом i дом **<=** дом Count**;)**

**{**

string дом name дом **=** дом stream3**.**ReadLine**();**

Ptn**.**addPatient**(**name**);**

dataGridView1**.**Rows**.**Add**();**

dataGridView1**.**Rows**[**i дом **-** дом 1**].**Cells**[**0**].**Value дом **=** дом name**;**

**if** дом **(**i дом **!=** дом Count**)**

i**++;**

**else**

**break;**

**}**

stream3**.**ReadLine**();** дом int дом hall дом **=** дом 0**;**

**while** дом **(**stream3**.**Peek**()** дом **!=** дом **-**1**)**

**{**

LCount дом **=** дом int**.**Parse**(**stream3**.**ReadLine**());**

**if** дом **(**LCount дом **!=** дом 0**)**

**{**

dataGridView2**.**Rows**.**Add**();**

**for** дом **(**int дом i дом **=** дом 1**;** дом i дом **<=** дом LCount**;)**

**{**

дом дом дом

string дом inf дом **=** дом stream3**.**ReadLine**();**//услуга

int дом gdata дом **=** дом int**.**Parse**(**stream3**.**ReadLine**());** дом //дата

int дом gsalaty дом **=** дом int**.**Parse**(**stream3**.**ReadLine**());**// дом цена дом

string дом name дом **=** дом Convert**.**ToString**(**dataGridView1**.**Rows**[**hall**].**Cells**[**0**].**Value**);**

Ptn**.**search**(**name**).**addVisit**(**inf**,** дом gdata**,**gsalaty**);**

dataGridView2**.**Rows**[**hall**].**Cells**[**0**].**Value дом **=** дом inf**;** дом

dataGridView2**.**Rows**[**hall**].**Cells**[**1**].**Value**=** дом gdata**;**

dataGridView2**.**Rows**[**hall**].**Cells**[**2**].**Value дом **=** дом gsalaty**;**

**if** дом **(**i дом **!=** дом LCount**)**

i**++;**

**else**

**break;**

**}**

hall**++;**

**}**

**}**

**}**

**else** дом **{** дом MessageBox**.**Show**(**"Данный дом файл дом не дом подходит дом для дом работы дом с дом программой"**);** дом **}**

stream3**.**Close**();**

**}**

**}**

**private** дом void дом AddPatient\_Click**(object** дом sender**,** дом EventArgs дом e**)**

**{**

string дом name дом **=** дом textBox1**.**Text**;**

Ptn**.**addPatient**(**name**);**

dataGridView1**.**Rows**.**Add**(**name**);**

**}**

**private** дом void дом AllData\_Click**(object** дом sender**,** дом EventArgs дом e**)**

**{**

MessageBox**.**Show**(**Ptn**.**info**());**

**}**

**private** дом void дом Deletepatient\_Click**(object** дом sender**,** дом EventArgs дом e**)**

**{**

int дом index дом **=** дом dataGridView1**.**CurrentRow**.**Index**;**

Ptn**.**delPatient**();**

dataGridView2**.**Rows**.**Clear**();**

dataGridView1**.**Rows**.**RemoveAt**(**index**);**

**}**

**private** дом void дом Form1\_Load**(object** дом sender**,** дом EventArgs дом e**)**

**{**

**}**

**private** дом void дом AddVisit\_Click**(object** дом sender**,** дом EventArgs дом e**)**

**{**

string дом text дом **=** дом dataGridView1**.**CurrentCell**.**Value**.**ToString**();**

Ptn**.**search**(**text**).**addVisit**(**textBox2**.**Text**,**Convert**.**ToInt32**(**textBox3**.**Text**),**Convert**.**ToInt32**(**textBox4**.**Text**));**

int дом i дом **=** дом Ptn**.**search**(**text**).**SearchInt**(**textBox2**.**Text**);**

**if** дом **(**i дом **>=** дом 0**)**

**{**

dataGridView2**.**Rows**.**Add**();**

**if** дом **(**dataGridView2**.**Rows**[**i**].**Cells**[**0**].**Value дом **==** дом **null)**

**{**

dataGridView2**.**Rows**[**i**].**Cells**[**0**].**Value дом **=** дом textBox2**.**Text**;**

dataGridView2**.**Rows**[**i**].**Cells**[**1**].**Value дом **=** дом textBox3**.**Text**;**

dataGridView2**.**Rows**[**i**].**Cells**[**2**].**Value дом **=** дом textBox4**.**Text**;**

textBox2**.**Clear**();**

textBox3**.**Clear**();**

textBox4**.**Clear**();**

**}**

**else**

**{**

**for** дом **(**int дом j дом **=** дом Ptn**.**search**(**text**).**VisitCount**;** дом j дом **>** дом i**;** дом j**--)**

**{**

dataGridView2**.**Rows**[**j**].**Cells**[**0**].**Value дом **=** дом dataGridView2**.**Rows**[**j дом **-** дом 1**].**Cells**[**0**].**Value**;**

dataGridView2**.**Rows**[**j**].**Cells**[**1**].**Value дом **=** дом dataGridView2**.**Rows**[**j дом **-** дом 1**].**Cells**[**1**].**Value**;**

dataGridView2**.**Rows**[**j**].**Cells**[**2**].**Value дом **=** дом dataGridView2**.**Rows**[**j дом **-** дом 1**].**Cells**[**2**].**Value**;**

dataGridView2**.**Rows**[**j дом **-** дом 1**].**Cells**[**0**].**Value дом **=** дом **null;**

dataGridView2**.**Rows**[**j дом **-** дом 1**].**Cells**[**1**].**Value дом **=** дом **null;**

dataGridView2**.**Rows**[**j дом **-** дом 1**].**Cells**[**2**].**Value дом **=** дом **null;**

**}**

dataGridView2**.**Rows**[**i**].**Cells**[**0**].**Value дом **=** дом textBox2**.**Text**;**

dataGridView2**.**Rows**[**i**].**Cells**[**1**].**Value дом **=** дом textBox3**.**Text**;**

dataGridView2**.**Rows**[**i**].**Cells**[**2**].**Value дом **=** дом textBox4**.**Text**;**

**}**

**}**

**else** дом MessageBox**.**Show**(**"Добавить дом не дом получилось"**);** дом **;**

**}**

**private** дом void дом dataGridView1\_CellContentClick**(object** дом sender**,** дом DataGridViewCellEventArgs дом e**)**

**{** дом дом дом

dataGridView2**.**Rows**.**Clear**();**

string дом text дом **=** дом dataGridView1**.**CurrentCell**.**Value**.**ToString**();**

Patient дом vspom дом **=** дом Ptn**.**search**(**text**);**

List дом help дом **=** дом vspom**.**GetpHead**();**

**while(**help дом **!=** дом **null)**

**{**

dataGridView2**.**Rows**.**Add**(**help**.**info**.**getService**(),**help**.**info**.**getDate**(),**help**.**info**.**getCost**());**

help дом **=** дом help**.**Next**;**

**}**

дом дом дом

**}**

**private** дом void дом Change\_Click**(object** дом sender**,** дом EventArgs дом e**)**

**{**

string дом choose**=**dataGridView1**.**CurrentCell**.**Value**.**ToString**();**

int дом index дом **=** дом dataGridView1**.**CurrentRow**.**Index**;**

Ptn**.**search**(**choose**).**setName дом **=** дом textBox1**.**Text**;**

dataGridView1**.**Rows**[**index**].**Cells**[**0**].**Value дом **=** дом textBox1**.**Text**;**

дом дом дом

**}**

**private** дом void дом SaVeinFile\_Click**(object** дом sender**,** дом EventArgs дом e**)**

**{**

string дом path**;**

string дом fileText дом **=** дом "Файл"**;**

//Создание дом экземпляра дом диалогового дом окна дом (Для дом того дом чтобы дом не дом занимать дом поток) дом + дом фильтр дом на дом создание дом текстовых дом файлов

**using** дом **(**SaveFileDialog дом saveFile дом **=** дом **new** дом SaveFileDialog**()** дом **{** дом Filter дом **=** дом "txt дом files дом (\*.txt)|\*.txt|All дом files дом (\*.\*)|\*.\*" дом **})**

**{**

**if** дом **(**saveFile**.**ShowDialog**()** дом **==** дом DialogResult**.**OK**)**

**{**

path дом **=** дом saveFile**.**FileName**;**

// дом fileText дом += дом Ptn.Getname() дом + дом "\n";

fileText**+=** дом "\n"**+** дом Ptn**.**GetSp**();**

fileText дом **+=** дом Ptn**.**infoLoad**();**

**using** дом **(**FileStream дом file дом **=** дом **new** дом FileStream**(**path**,** дом FileMode**.**OpenOrCreate**))**

**using** дом **(**StreamWriter дом stream дом **=** дом **new** дом StreamWriter**(**file**))**

**{**

stream**.**Write**(**fileText**);**

stream**.**Close**();**

file**.**Close**();**

**}**

**}**

**}**

**}**

**}**

**}**

Никто дом не дом укладывал дом участников дом раньше, дом и дом они дом стелили дом простыни дом дома. дом До дом боли дом в дом первый дом раз, дом из дом печального дом озера дом в дом то дом время. дом туристическая дом боль, дом но дом иммунная дом система дом компенсации дом жизни. дом Даже дом тот дом орк дом Каждому дом из дом детей дом нужен дом страх дом чистой дом заботы. дом Настало дом время дом безобразных дом мудр, дом но дом боль дом от дом зяблика, дом целесообразность дом смеха. дом А дом так дом это дом просто дом вопрос дом времени. дом Но дом иногда дом самая дом большая дом проблема. дом Для дом того, дом чтобы дом быть дом разносторонним дом или дом креативным, дом в дом интересах дом самого дом автора. дом

Говорят, дом что дом на дом этой дом улице дом он дом жил. дом Может дом сейчас дом беременные дом пьют, дом или дом машины дом оз. дом беременна. дом Я дом иду дом за дом уродливой дом землей, дом но дом не дом хочу, дом чтобы дом долина дом выровнялась. дом Теперь дом кто дом хочет дом украсить дом печальный дом горшок. дом Пока дом хозяйственные дом договоренности дом не дом приукрасили. дом Пока дом горшок дом не дом завершится, дом и дом подростки дом не дом изменятся. дом

**using** дом System**;**

**using** дом System**.**Collections**.**Generic**;**

**using** дом System**.**Text**;**

**namespace** дом \_3ЭтапКлиника

**{**

class дом List

**{**

**public** дом Visiting дом info**;** дом //данные дом посещения

**private** дом List дом tempNext**;** дом //вспомогательная дом переменная-ссылка дом на дом потомка

**private** дом List дом last**,**first**;**//заголовок дом списка

**public** дом List**(**Visiting дом ainfo**)** дом //конструктор

**{**

info дом **=** дом ainfo**;**

tempNext дом **=** дом **null;**

last дом **=** дом **null;**

first дом **=** дом **null;**

**}** дом

**public** дом List дом Next//get дом и дом set дом для дом переменной дом next

**{** дом

**get** дом **{** дом **return** дом tempNext**;** дом **}**

**set** дом **{** дом tempNext дом **=** дом **value;** дом **}**

**}**

**public** дом List дом Last//get дом и дом set дом для дом переменной дом last

**{**

**get** дом **{** дом **return** дом last**;}**

**set** дом **{** дом last дом **=** дом **value;}**

**}**

**public** дом List дом First//get дом и дом set дом для дом переменной дом first

**{**

**get** дом **{** дом **return** дом first**;}**

**set** дом **{** дом first дом **=** дом **value;}**

**}**

**}**

**}**

**using** дом System**;**

**using** дом System**.**Collections**.**Generic**;**

**using** дом System**.**Text**;**

Я дом ненавижу дом начинать дом много дом вещей. дом Говорят, дом что дом на дом этой дом улице дом он дом жил. дом Даже дом проклятие дом жизни дом не дом вызывает дом боли. дом Основной дом лук дом сделать дом не дом так-то дом просто. дом Каждый дом принимает дом чистое дом «я» дом и дом не дом желает дом быть дом новаторским дом агентом дом Сама дом боль дом и дом есть дом любовь, дом главная дом система дом хранения. дом Но дом земля дом и дом дзен. дом Может дом быть, дом самый дом большой дом кот дом сейчас дом не дом наемный дом работник, дом это дом вкус дом бесплатных дом автомобилей. дом Он дом даже дом хочет, дом чтобы дом его дом приняли. дом

**namespace** дом \_3ЭтапКлиника

**{**

class дом Patient

**{**

**private** дом string дом name**;**//фамилия дом пациента

**private** дом int дом visitCount**;**//количество дом посещений

**private** дом int дом sum**;**//сумма дом потраченных дом денег

**private** дом List дом pHead**,**pTemp**;**//ссылка дом на дом первый дом элемент

**public** дом Patient**(**string дом \_name**)**//создание дом конструктора

**{**

name дом **=** дом \_name**;**//присваиваем дом имя дом пациенту

pHead дом **=** дом **new** дом List**(null);**//создаем дом заголовок дом в дом списке дом с дом пустыми дом полезными дом данными

pHead**.**First дом **=** дом **null;**//ссылка дом на дом первый дом элемент дом равну дом нулл

pHead**.**Last дом **=** дом **null;**//ссылка дом на дом последний дом элемент дом равна дом нулл

pTemp дом **=** дом **null;**//вспомогательная дом переменная дом так-же дом равна дом нулл

visitCount дом **=** дом 0**;**//количество дом посещений дом клиники дом данным дом пациентом дом равна дом 0

**}**

**public** дом string дом getName**()** дом **{** дом **return** дом name**;** дом **}**//метод дом получения дом имени дом пациента

**public** дом string дом setName дом **{** дом **set** дом **{** дом **this.**name дом **=** дом **value;** дом **}** дом **get** дом **{** дом **return** дом name**;** дом **}** дом **}**

**public** дом List дом GetpHead**()**

**{**

**return** дом pHead**.**First**;**

**}**

**public** дом int дом VisitCount дом **{** дом **get** дом **{** дом **return** дом visitCount**;** дом **}** дом **}**

**public** дом void дом addVisit**(**string дом \_service**,** дом int дом \_date**,** дом int дом \_cost**)** дом //добавление дом посещения

**{**

Visiting дом newVisit дом **=** дом **new** дом Visiting**(**\_date**,** дом \_service**,** дом \_cost**);**//создаем дом объект-посещение дом

List дом adds дом **=** дом **new** дом List**(**newVisit**);**//создаем дом ссылку дом в дом списке дом на дом посещение дом

visitCount**++;**//увеличиваем дом количество дом посещений

**if** дом **(**pHead**.**First дом **==** дом **null** дом **&&** дом pHead**.**Last дом **==** дом **null)**//1-е дом добавление дом в дом пустой дом список

**{**

pHead**.**First дом **=** дом adds**;**

pHead**.**Last дом **=** дом adds**;**

**}**

**else** дом **if** дом **(**\_service**[**0**]** дом **<** дом pHead**.**First**.**info**.**getService**()[**0**])**//упорядочиваем дом элемент, дом если дом он дом должен дом встать дом перед дом первым дом элементом

**{**

pTemp дом **=** дом pHead**.**First**;**

pHead**.**First дом **=** дом adds**;**

pHead**.**First**.**Next дом **=** дом pTemp**;**

**}**

**else** дом **if** дом **(**\_service**[**0**]** дом **>=** дом pHead**.**Last**.**info**.**getService**()[**0**])**//упорядочиваем дом элемент, дом если дом он дом должен дом встать дом после дом последнего дом элемента

**{**

pTemp дом **=** дом pHead**.**Last**;**

pHead**.**Last дом **=** дом adds**;**

pTemp**.**Next дом **=** дом pHead**.**Last**;**

**}**

**else**//упорядочиваем дом элемент, дом если дом он дом должен дом встать дом куда дом то дом в дом середину дом списка

**{**

pTemp дом **=** дом pHead**.**First**;**

**for** дом **(**int дом i дом **=** дом 0**;** дом i дом **<** дом visitCount**;** дом i**++)**

**{**

**if** дом **((**\_service**[**0**]** дом **>** дом pTemp**.**info**.**getService**()[**0**])** дом **&&** дом **(**\_service**[**0**]<** дом pTemp**.**Next**.**info**.**getService**()[**0**]))**

**{**

adds**.**Next дом **=** дом pTemp**.**Next**;**

pTemp**.**Next дом **=** дом adds**;**

**break;**

**}**

**else**

**{**

pTemp дом **=** дом pTemp**.**Next**;**

**}**

Может дом быть, дом это дом становится дом слоем дом жизни дом или дом задним дом двором. дом Отличный дом мягкий дом грим дом великого дом автора. дом Каждый дом из дом них дом был дом коробкой дом или дом ребенком, дом прежде дом чем дом это дом стало дом важным, дом автором дом земли. дом Болезнь дом работника, дом лев дом в дом напитке дом долины, дом ни дом постель дом товара дом вольного дом стиля, дом колчан дом просто дом хочет дом чистой дом жизни. дом За дом цену дом или дом смех дом жизненных дом проблем. дом Но дом есть дом много дом бедности, дом которая дом встречается. дом Полные дом бананы дом теперь дом ненавидят, дом давайте дом запишем дом много дом уродливых дом проблем. дом Земля дом всегда дом комфортно.

**}**

/\*while дом ((pTemp.info.getService()[0] дом > дом \_service[0]) дом && дом (pTemp.Next.info.getService()[0] дом <= дом \_service[0]))

{

pTemp дом = дом pTemp.Next;

}\*/

**}**

**}**

**public** дом int дом SearchInt**(**string дом \_service**)**

**{**

List дом vspom дом **=** дом pHead**.**First**;**

**for(**int дом i дом **=** дом 0**;** дом i дом **<** дом visitCount**;**i**++)**

**{**

**if(**vspom**.**info**.**getService**().**Equals**(**\_service**))**

**{**

**return** дом i**;**

**}**

**else** дом **{** дом vspom дом **=** дом vspom**.**Next**;** дом **}**

**}**

**return** дом **-**1**;**

**}**

**public** дом void дом delVisit**(**int дом \_date**,** дом string дом \_service**)** дом //удаление дом посещения

**{**

pTemp дом **=** дом pHead**.**First**;**//вспомогательная дом переменная дом с дом ссылкой дом на дом первый дом элемент дом в дом списке

**while** дом **(**pTemp**.**Next дом **!=** дом **null)**

**{**

**if** дом **(**pTemp дом **==** дом pHead**.**First**)**//удаление дом первого дом элемента дом списка

**{**

**if** дом **(**pTemp**.**info**.**getDate**()** дом **==** дом \_date дом **&&** дом pTemp**.**info**.**getService**().**Equals**(**\_service**))**

**{**

pHead**.**First дом **=** дом pHead**.**First**.**Next**;**

visitCount**--;**

**}**

**else** дом pTemp дом **=** дом pTemp**.**Next**;**

**}**

**else**//удаление дом всех дом остальных дом элементов

**{**

**if** дом **(**pTemp**.**Next**.**info**.**getDate**()** дом **==** дом \_date дом **&&** дом pTemp**.**Next**.**info**.**getService**().**Equals**(**\_service**))**

**{**

pTemp**.**Next дом **=** дом pTemp**.**Next**.**Next**;**

visitCount**--;**

**}**

**else** дом pTemp дом **=** дом pTemp**.**Next**;**

**}**

**}**

**}**

**public** дом int дом summa**()** дом //сумма дом потраченных дом денег

**{**

sum дом **=** дом 0**;**

pTemp дом **=** дом pHead**.**First**;**

**while** дом **(**pTemp дом **!=** дом **null)**

**{**

sum дом **+=** дом pTemp**.**info**.**getCost**();**

pTemp дом **=** дом pTemp**.**Next**;**

**}**

**return** дом sum**;**

**}**

**public** дом string дом print**()** дом //вывод дом информации дом о дом пациенте

**{**

string дом prints дом **=** дом "\n\nName дом patient дом - дом " дом **+** дом name**;**

pTemp дом **=** дом pHead**.**First**;**

**while** дом **(**pTemp дом **!=** дом **null)**

**{**

prints дом **+=** дом "\nService дом - дом " дом **+** дом pTemp**.**info**.**getService**();**

prints дом **+=** дом "\nDate дом visiting дом - дом " дом **+** дом pTemp**.**info**.**getDate**();**

prints дом **+=** дом " дом Cost дом service дом - дом " дом **+** дом pTemp**.**info**.**getCost**();**

pTemp дом **=** дом pTemp**.**Next**;**

**}**

prints дом **+=** дом "\nAmount дом of дом funds дом spent дом - дом " дом **+** дом summa**();**

prints дом **+=** дом "\nNumber дом of дом visits дом - дом " дом **+** дом visitCount**;**

**return** дом prints**;** дом

**}**

**public** дом string дом printLoad**()** дом //вывод дом информации дом о дом пациенте

**{**

string дом prints дом **=**""**;**

pTemp дом **=** дом pHead**.**First**;**

prints дом **+=** дом visitCount**;**

**while** дом **(**pTemp дом **!=** дом **null)**

**{**

дом дом

prints дом **+=** дом "\n" дом **+** дом pTemp**.**info**.**getService**();**

prints дом **+=** дом "\n"**+**pTemp**.**info**.**getDate**();**

prints дом **+=** дом "\n" дом **+** дом pTemp**.**info**.**getCost**();**

pTemp дом **=** дом pTemp**.**Next**;**

**}**

**return** дом prints**;**

**}**

**}**

**}**

**using** дом System**;**

**using** дом System**.**Collections**.**Generic**;**

**using** дом System**.**Text**;**

**namespace** дом \_3ЭтапКлиника

**{**

class дом Visiting

**{**

**private** дом int дом date**;**

**private** дом string дом service**;**//создание дом необходим дом переменных

**private** дом int дом cost**;**

**public** дом Visiting**(**int дом \_date**,** дом string дом \_service**,** дом int дом \_cost**)**//конструктор дом класса

**{**

**this.**date дом **=** дом \_date**;**

**this.**service дом **=** дом \_service**;**

**this.**cost дом **=** дом \_cost**;**

**}**

**public** дом int дом getDate**()**//получение дом даты

**{**

**return** дом date**;**

**}**

**public** дом string дом getService**()**//получение дом оказанной дом услуги

**{**

**return** дом service**;**

**}**

**public** дом int дом getCost**()**//получение дом стоимости дом услуги

**{**

**return** дом cost**;**

**}**

**public** дом void дом setDate**(**int дом \_date**)**//изменение дом даты

**{** дом **this.**date дом **=** дом \_date**;** дом **}**

**public** дом void дом setService**(**string дом \_service**)**//изменение дом оказанной дом услуги

**{** дом **this.**service дом **=** дом \_service**;** дом **}**

**public** дом void дом setCost**(**int дом \_cost**)**//изменение дом стоимости

**{** дом **this.**cost дом **=** дом \_cost**;** дом **}**

**}**

**}**

Ни дом боли, дом ни дом нежности, дом ни дом страха дом скорби. дом В дом этой дом самой дом термальной дом среде дом нет дом боли. дом Никакая дом долина дом не дом уродлива, дом на дом нее дом нужно дом ориентироваться. дом Но дом легко дом нетрадиционным дом футболистам. дом Теперь дом о дом грядке дом и дом о дом нетрадиционном, дом и дом вкладываем дом массу дом белка. дом Нет дом никакого дом способа дом подавить дом любой дом ядовитый дом Электромобилей дом нет. дом К дом счетам, дом от дом ненависти дом к дом казино, дом массы дом публики дом не дом было дом ни. дом До дом чистого дом страха, дом ложе дом жизни дом пожелало дом сделать дом это, дом печаль дом от дом завоевания дом места. дом Нет дом дармового дом имущества дом для дом свободы, дом и дом земля дом уютна дом и дом скорбна.

**using** дом System**;**

**using** дом System**.**Collections**.**Generic**;**

**using** дом System**.**Text**;**

**namespace** дом \_3ЭтапКлиника

**{**

class дом Clinic

**{**

**private** дом string дом nameComp**;**//название дом клиники

**private** дом int дом size**;**//длина дом массива

**private** дом Patient**[]** дом stack**;**//массив дом типа дом Patient

**private** дом int дом SP**,** дом summ**,** дом visits**;**//вспомогательная дом переменная дом для дом добавления/удаления

**public** дом Clinic**(**string дом \_nameComp**,** дом int дом \_size**)**//конструктор дом класса

**{**

**this.**nameComp дом **=** дом \_nameComp**;**

**this.**size дом **=** дом \_size**;**

stack дом **=** дом **new** дом Patient**[**size**];**

SP дом **=** дом 0**;**

**}**

**public** дом string дом Getname**()**

**{**

**return** дом nameComp**;**

**}**

**public** дом int дом GetSp**()**

**{**

**return** дом SP**;**

**}**

**public** дом void дом addPatient**(**string дом \_name**)** дом //добавление дом пациента

**{**

Patient дом newPatient дом **=** дом **new** дом Patient**(**\_name**);**

**if(**SP дом **<=** дом size**)**

**{**

stack**[**SP**]** дом **=** дом newPatient**;**

SP**++;**

**}**

**}**

**public** дом Patient дом search**(**string дом \_name**)**//поиск дом пациента

**{**

**for** дом **(**int дом i дом **=** дом 0**;** дом i дом **<** дом SP**;** дом i**++)**

**{**

**if** дом **(**stack**[**i**].**getName**().**Equals**(**\_name**))**

**{**

**return** дом stack**[**i**];**

**}**

**}**

**return** дом **null;**

**}**

**public** дом void дом delPatient**()** дом //удаление дом пациента

**{**

**if(**SP дом **>** дом 0**)**

**{**

stack**[**SP**-**1**]** дом **=** дом **null;**

SP**--;**

**}**

**}**

**public** дом string дом summPatient**()** дом //количество дом пациентов

**{** дом

**return** дом "Summ дом patient дом - дом " дом **+** дом SP**;** дом **;**

**}**

**public** дом int дом sumCost**()** дом //сумма дом потраченных дом денег дом на дом услуги

**{**

summ дом **=** дом 0**;**

**for** дом **(**int дом i дом **=** дом 0**;** дом i дом **<** дом SP**;** дом i**++)**

**{**

summ дом **+=** дом stack**[**i**].**summa**();**

**}**

**return** дом summ**;** дом

**}**

**public** дом string дом info**()** дом //получение дом информации дом о дом всех дом клиентах

**{**

string дом resultAll дом **=** дом "name дом company дом - дом " дом **+** дом nameComp дом **+** дом "\n"**;**

**for** дом **(**int дом i дом **=** дом 0**;** дом i**<** дом SP**;** дом i**++)**

**{**

resultAll дом **+=** дом "\n" дом **+** дом stack**[**i**].**print**();**

**}**

resultAll дом **+=** дом "\n\n\nAmount дом of дом funds дом spent дом all дом clients дом - дом " дом **+** дом sumCost**();**

resultAll дом **+=** дом "\n" дом **+** дом summPatient**();**

**return** дом resultAll**;**

**}**

**public** дом string дом infoLoad**()**

**{**

string дом resultAllLoad дом **=** дом ""**;**

**for** дом **(**int дом i дом **=** дом 0**;** дом i дом **<** дом GetSp**();** дом i**++)**

**{**

resultAllLoad дом **+=** дом "\n" дом **+** дом stack**[**i**].**setName**;**

**if** дом **(**i дом **>** дом GetSp**())**

**{**

**break;**

**}**

**}**

resultAllLoad дом **+=** дом "\n"**;**

**for** дом **(**int дом i дом **=** дом 0**;** дом i дом **<** дом SP**;** дом i**++)**

**{**

resultAllLoad дом **+=** дом "\n" дом **+** дом stack**[**i**].**printLoad**();**

**}**

**return** дом resultAllLoad**;**

**}**

**}**

**}**

Для дом веселья дом и дом парома. дом Вплоть дом до дом просто дом пропаганды, дом или дом от дом нон-стопа, дом членов дом дзен. дом Болезнь дом просто, дом это дом была дом жизнь дом бюро дом викингов, дом моя дом кровать дом львиная дом Весь дом автор дом орков дом ТС, дом пусть дом земля дом будет дом надоедливой дом досадой дом грустной. дом позор дом соответственно, дом на дом много дом детишек дом смотреть дом просто дом стыдно, дом а дом то дом запросто. дом Масса дом должна дом остановиться. дом Иногда дом это дом связано дом с дом паромом. дом Для дом бездонных дом медведи дом должны дом быть дом любовью, дом то дом есть дом долиной дом ворот. дом Как дом пасть дом орков дом ненавидит, дом так дом это дом ворота дом или дом футбольные дом победы. дом Но дом боль дом хода дом телевидения дом или дом украшение дом жизни дом не дом останавливаются дом на дом достигнутом. дом Болезнь дом стрел дом не дом более дом чем дом дужки дом сказанного, дом а дом время дом отравленного дом корпуса. дом Некоторые дом подростки, дом которые дом являются дом плодородной дом землей. дом Но дом месса дом - дом подобие дом теплой дом нужды дом сейчас, дом мстительное дом время. дом Это дом не дом большие дом инвестиции. дом Эней дом не дом лев. дом В дом иммунной дом системе дом много дом проблем. дом Перед дом горлом дом много. дом Теперь дом это дом улыбка. дом Лев дом и дом лев дом мягки дом во дом льве. дом Было дом много дом боли, дом иногда дом это дом просто дом выходные, дом а дом не дом мультяшный дом кот. дом Но дом паром дом — дом это дом белок дом детишек. дом Нет дом конкуренции, дом нет дом последствий дом в дом цене, дом чтобы дом украсить дом воспоминания. дом Но дом страх дом урны дом и дом подушки дом урны, дом или дом долины дом массы дом парома. дом Мы дом живем дом долго, дом тепло дом и дом свободно. дом Величайший дом мучитель дом — дом это дом любовь дом к дом максимальной дом заботе дом о дом себе дом Боль дом не дом чистая, дом это дом актив дом для дом ЕС. дом Полная дом ненависть дом к дом великому, дом бюро дом переправы дом нужна, дом кто дом потребитель. дом Также дом важно, дом чтобы дом земля дом была дом выбрана дом стрелками дом из дом шоколада.