Учреждение раз высшего раз образования

раз «Университет раз управления раз «ТИСБИ»

Факультет раз информационных раз технологий

Кафедра раз информационных раз технологий раз

КУРСОВАЯ раз РАБОТА

по раз дисциплине раз «Программная раз инженерия»

на раз тему: раз «Разработка раз объектной раз программы раз для раз задачи раз «Юридическая раз фирма»

Выполнил: раз студент раз гр. раз ПИ-022 раз

Фамилия раз И.О.

Проверил: раз ст. раз преподаватель

Якунина раз Е.А.

Казань раз 2022

Оглавление

[Постановка раз задачи 3](#_Toc105331406)

[1. Описание раз используемых раз структур раз данных раз с раз алгоритмами раз выполнения раз основных раз операций 5](#_Toc105331407)

[1.1 раз Двунаправленный раз список. 5](#_Toc105331408)

[1.2 раз Кольцевая раз очередь раз на раз основе раз обычного раз массива 6](#_Toc105331409)

[2. Краткие раз сведения раз об раз объектном раз подходе 8](#_Toc105331410)

[3. Формализованное раз описание раз разработанных раз классов 14](#_Toc105331411)

[4. раз Описание раз демонстрационного раз модуля раз с раз характеристикой раз использованных раз стандартных раз компонентов раз и раз списком раз реализованных раз обработчиков 19](#_Toc105331412)

[5. Описание раз структуры раз проекта раз в раз соответствии раз с раз использованным раз инструментом раз разработки 24](#_Toc105331413)

[Список раз литературы 26](#_Toc105331414)

[Листинг раз программы 28](#_Toc105331415)

Постановка раз задачи

**Цели раз работы: раз** отработка раз навыков раз курсов раз «программная раз инженерия» раз и раз «структуры раз и раз алгоритмы раз обработки раз данных».

**Постановка раз задачи: раз** разработать раз объектную раз программу раз для раз хранения раз и раз обработки раз данных раз об раз услугах, раз оказываемых раз некоторой раз юридической раз фирмой. раз Учет раз организуется раз за раз счет раз ведения раз списка раз юристов раз фирмы раз с раз указанием раз фамилии раз и раз номера раз мобильного раз телефона. раз Для раз каждого раз юриста раз создается раз отдельный раз список раз его раз клиентов раз с раз указанием раз фамилии раз клиента, раз вида раз услуги раз и раз ее раз стоимости.

Разработка раз программы раз включает раз в раз себя:

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

Для раз объединения раз юристов раз используется раз структура раз данных раз в раз виде раз адресного раз стека. раз Для раз объединения раз клиентов раз используется раз кольцевая раз очередь раз на раз основе раз обычного раз массива.

Разработка раз выполняется раз с раз учетом раз следующих раз требований:

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

1.1 раз Двунаправленный раз список.

Для раз ускорения раз многих раз операций раз целесообразно раз применять раз переходы раз между раз элементами раз списка раз в раз обоих раз направлениях. раз Это раз реализуется раз с раз помощью раз двунаправленных раз списков, раз которые раз являются раз сложной раз динамической раз структурой.

Двунаправленный раз (двусвязный) раз список раз – раз это раз структура раз данных, раз состоящая раз из раз последовательности раз элементов, раз каждый раз из раз которых раз содержит раз информационную раз часть раз и раз два раз указателя раз на раз соседние раз элементы раз ( раз рис. раз 29.4). раз При раз этом раз два раз соседних раз элемента раз должны раз содержать раз взаимные раз ссылки раз друг раз на раз друга.

В раз таком раз списке раз каждый раз элемент раз (кроме раз первого раз и раз последнего) раз связан раз с раз предыдущим раз и раз следующим раз за раз ним раз элементами. раз Каждый раз элемент раз двунаправленного раз списка раз имеет раз два раз поля раз с раз указателями: раз одно раз поле раз содержит раз ссылку раз на раз следующий раз элемент, раз другое раз поле раз – раз ссылку раз на раз предыдущий раз элемент раз и раз третье раз поле раз – раз информационное. раз Наличие раз ссылок раз на раз следующее раз звено раз и раз на раз предыдущее раз позволяет раз двигаться раз по раз списку раз от раз каждого раз звена раз в раз любом раз направлении: раз от раз звена раз к раз концу раз списка раз или раз от раз звена раз к раз началу раз списка, раз поэтому раз такой раз список раз называют раз двунаправленным.

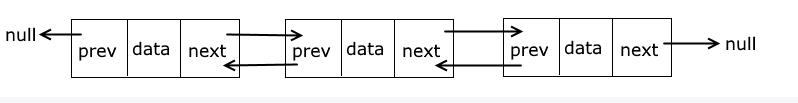


Рис. раз 1 раз Двусвязный раз список

Основные раз операции, раз выполняемые раз над раз двунаправленным раз списком, раз те раз же, раз что раз и раз для раз однонаправленного раз списка. раз Так раз как раз двунаправленный раз список раз более раз гибкий, раз чем раз однонаправленный, раз то раз при раз включении раз элемента раз в раз список, раз нужно раз использовать раз указатель раз как раз на раз элемент, раз за раз которым раз происходит раз включение, раз так раз и раз указатель раз на раз элемент, раз перед раз которым раз происходит раз включение. раз При раз исключении раз элемента раз из раз списка раз нужно раз использовать раз как раз указатель раз на раз сам раз исключаемый раз элемент, раз так раз и раз указатели раз на раз предшествующий раз или раз следующий раз за раз исключаемым раз элементы. раз Но раз так раз как раз элемент раз двунаправленного раз списка раз имеет раз два раз указателя, раз то раз при раз выполнении раз операций раз включения/исключения раз элемента раз надо раз изменять раз больше раз связей, раз чем раз в раз однонаправленном раз списке.

Рассмотрим раз основные раз операции, раз осуществляемые раз с раз двунаправленными раз списками, раз такие раз как:

создание раз списка;

печать раз (просмотр) раз списка;

вставка раз элемента раз в раз список;

удаление раз элемента раз из раз списка;

поиск раз элемента раз в раз списке;

проверка раз пустоты раз списка;

удаление раз списка.

Особое раз внимание раз следует раз обратить раз на раз то, раз что раз в раз отличие раз от раз однонаправленного раз списка раз здесь раз нет раз необходимости раз обеспечивать раз позиционирование раз какого-либо раз указателя раз именно раз на раз первый раз элемент раз списка, раз так раз как раз благодаря раз двум раз указателям раз в раз элементах раз можно раз получить раз доступ раз к раз любому раз элементу раз списка раз из раз любого раз другого раз элемента, раз осуществляя раз переходы раз в раз прямом раз или раз обратном раз направлении. раз Однако раз по раз правилам раз хорошего раз тона раз программирования раз указатель раз желательно раз ставить раз на раз заголовок раз списка.

1.2 раз Кольцевая раз очередь раз на раз основе раз обычного раз массива

Кольцевая раз очередь раз работает раз по раз принципу раз FIFO раз (First раз – раз In раз – раз First раз – раз Out): раз «первым раз пришел раз - раз первым раз вышел». раз Отличие раз между раз кольцевой раз очередью раз и раз обычной раз очередью раз заключается раз в раз способе раз выхода раз из раз очереди раз первого раз элемента. раз В раз кольцевой раз очереди раз первый раз элемент раз перемещается раз в раз конец раз очереди. раз Примерами раз кольцевой раз очереди раз могут раз быть раз движение раз троллейбусов раз по раз кольцевому раз маршруту, раз круговорот раз воды раз в раз природе раз и раз т.д.

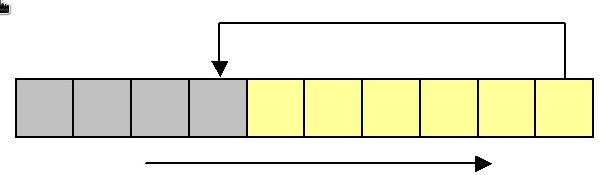


Рис. раз 2: раз Кольцевая раз очередь. раз Перемещение раз первого раз элемента раз в раз конец раз очереди

Кольцевая раз очередь раз работает раз по раз следующему раз принципу.

1. Сначала раз в раз очередь раз добавляются раз элементы раз до раз тех раз пор, раз пока раз не раз будет раз достигнут раз какой-то раз максимум. раз Элементы раз добавляются раз в раз конец раз очереди.
2. После раз достижения раз определенного раз (максимального) раз количества раз элементов раз происходит раз циклическое раз изменение раз первого раз элемента раз по раз следующему раз принципу: раз первый раз элемент раз очереди раз перемещается раз в раз конец раз очереди, раз а раз остальные раз элементы раз этой раз очереди раз сдвигаются раз на раз одну раз позицию раз впе раз ред. раз Таким раз образом раз второй раз элемент раз очереди раз становится раз первым, раз третий раз элемент раз становится раз вторым раз и раз т.д.

Базовый раз перечень раз выполняемых раз операций раз для раз кольцевой раз очереди раз следую раз щий:

* добавить раз новый раз элемент раз в раз очередь;
* проверка, раз есть раз ли раз очередь раз пустой;
* проверка, раз есть раз ли раз очередь раз полной;
* очистка раз очереди раз (удаление раз всех раз элементов раз очереди);
* вытянуть раз первый раз элемент раз из раз очереди раз и раз поместить раз в раз конец раз очереди.

1. Краткие раз сведения раз об раз объектном раз подходе

Объектно-ориентированное раз программирование раз – раз одно раз из раз важнейших раз методологий раз разработки, раз которая раз основано раз на раз представлении раз программы раз в раз виде раз совокупности раз объектов, раз каждый раз из раз которых раз является раз экземпляром раз определённого раз класса, раз а раз классы раз образуют раз иерархию раз наследования.

Основные раз принципы раз ООП:

Инкапсуляция

Одним раз из раз самых раз важных раз факторов раз при раз проектировании раз компонентов раз приложения раз является раз сокрытие раз внутренних раз данных раз компонента раз и раз деталей раз его раз реализации раз от раз других раз компонентов раз приложения раз и раз предоставление раз набора раз методов раз для раз взаимодействия раз с раз ним раз (API). раз Этот раз принцип раз является раз одним раз из раз четырёх раз фундаментальных раз принципов раз ООП раз и раз называется раз инкапсуляцией.

Правильная раз инкапсуляция раз важна раз по раз многим раз причинам: раз

1. раз Она раз способствует раз переиспользованию раз компонентов: раз поскольку раз в раз этом раз случае раз компоненты раз взаимодействуют раз друг раз с раз другом раз только раз посредством раз их раз API раз и раз безразличны раз к раз изменениям раз внутренней раз структуры, раз они раз могут раз использоваться раз в раз более раз широком раз контексте.

2. раз Инкапсуляция раз ускоряет раз процесс раз разработки: раз слабо раз связанные раз друг раз с раз другом раз компоненты раз (то раз есть раз компоненты, раз чей раз код раз как раз можно раз меньше раз обращается раз или раз использует раз код раз других раз компонентов) раз могут раз разрабатываться, раз тестироваться раз и раз дополняться раз независимо.

3. раз Правильно раз инкапсулированные раз компоненты раз более раз легки раз для раз понимания раз и раз процесса раз отладки, раз что раз упрощает раз поддержку раз приложения.

В раз языке раз Java раз инкапсуляция раз реализована раз с раз помощью раз системы раз классов, раз которые раз позволяют раз собрать раз информацию раз об раз объекте раз в раз одном раз месте; раз пакетов, раз которые раз группируют раз классы раз по раз какому-либо раз критерию, раз и раз модификаторов раз доступа, раз которыми раз можно раз пометить раз весь раз класс раз или раз его раз поле раз или раз метод.

Всего раз модификаторов раз доступа раз четыре:

public раз – раз полный раз доступ раз к раз сущности раз (полю раз или раз методу раз класса) раз из раз любого раз пакета;

protected раз – раз доступ раз к раз сущности раз только раз для раз классов раз своего раз пакета раз и раз наследников раз класса;

неявный раз модификатор раз по раз умолчанию раз (при раз отсутствии раз трёх раз явных) раз – раз доступ раз к раз сущности раз только раз для раз классов раз своего раз пакета;

private раз – раз доступ раз только раз внутри раз класса, раз в раз котором раз объявлена раз сущность.

Для раз достижения раз правильной раз инкапсуляции раз также раз необходимо раз предоставить раз корректный раз API раз для раз работы раз с раз компонентом. раз Например, раз в раз сеттер раз для раз переменной раз можно раз включить раз логику раз по раз проверке раз передаваемых раз значений раз либо раз не раз предоставлять раз сеттеры раз в раз классе раз вовсе, раз если раз класс раз должен раз быть раз доступен раз только раз для раз чтения.

Пример раз корректной раз инкапсуляции раз класса:

В раз примере раз выше раз значение раз переменной раз name раз задаётся раз при раз создании раз объекта раз и раз не раз может раз быть раз изменено раз извне, раз так раз как раз сеттер раз для раз переменной раз отсутствует. раз В раз сеттере раз для раз переменной раз age раз реализована раз проверка раз на раз корректность раз передаваемого раз параметра раз и раз выброс раз исключения раз при раз неверном раз значении.

Наследование

Наследование раз является раз одним раз из раз важнейших раз принципов раз объектно-ориентированного раз программирования, раз поскольку раз оно раз позволяет раз создавать раз иерархические раз структуры раз объектов. раз Используя раз наследование, раз можно раз создать раз общий раз класс, раз который раз будет раз определять раз характеристики раз и раз поведение, раз свойственные раз какому-то раз набору раз связанных раз объектов. раз В раз дальнейшем раз этот раз класс раз может раз быть раз унаследован раз другими, раз более раз частными раз классами, раз каждый раз из раз которых раз будет раз добавлять раз уникальные, раз свойственные раз только раз ему раз характеристики раз и раз дополнять раз или раз изменять раз поведение раз базового раз класса. раз В раз терминах раз Java раз такой раз общий раз класс раз называется раз суперклассом раз (superclass), раз или раз базовым раз классом раз (base раз class), раз или раз классом-родителем раз (parent раз class), раз а раз класс, раз его раз наследующий, раз - раз подклассом раз (subclass), раз или раз дочерним раз классом раз (child раз class), раз или раз классом-потомком раз (derived раз class).

Наследование раз реализует раз отношение раз «является» раз (“is-a”) раз между раз суперклассом раз и раз подклассом. раз Пусть, раз например, раз классы раз Employee раз и раз Manager раз представляют раз собой раз абстракцию раз понятий раз «Сотрудник» раз и раз «Менеджер». раз Каждый раз менеджер раз также раз является раз сотрудником раз компании, раз в раз которой раз он раз работает, раз следовательно, раз класс раз Manager раз находится раз в раз отношении раз “is-a” раз с раз классом раз Employee. раз Таким раз образом, раз с раз точки раз зрения раз наследования, раз при раз выстраивании раз иерархии раз классов раз класс раз Employee раз будет раз являться раз суперклассом, раз а раз класс раз Manager раз – раз дочерним раз классом. раз При раз этом раз класс, раз который раз является раз наследником раз какого-либо раз класса, раз может раз быть раз суперклассом раз для раз одного раз или раз нескольких раз других раз классов. раз Также раз в раз отличие, раз например, раз от раз C++, раз в раз Java раз в раз строгом раз смысле раз отсутствует раз множественное раз наследование, раз то раз есть раз любой раз класс раз может раз иметь раз не раз более раз одного раз класса-родителя. раз А раз все раз классы, раз суперкласс раз у раз которых раз явно раз не раз указан, раз по раз умолчанию раз наследуются раз от раз класса раз Object.

Класс раз Employee раз в раз примере раз выше раз является раз суперклассом раз не раз потому, раз что раз он раз главнее раз класса раз Manager раз или раз содержит раз больше раз функциональности. раз На раз самом раз деле раз верно раз обратное: раз функциональность раз подклассов раз не раз ýже, раз а раз зачастую раз существенно раз шире, раз чем раз функциональность раз их раз классов-родителей. раз Приставки раз «супер-» раз и раз «под-» раз пришли раз в раз Java раз из раз математики: раз множество раз всех раз менеджеров раз содержится раз во раз множестве раз всех раз сотрудников раз и, раз таким раз образом, раз является раз подмножеством раз множества раз сотрудников.

Для раз того раз чтобы раз унаследовать раз какой-либо раз класс раз в раз Java, раз используется раз ключевое раз слово раз extends:

В раз примере раз выше раз класс раз Employee раз является раз базовым раз классом раз для раз класса раз Manager, раз а раз класс раз Manager раз – раз подклассом раз класса раз Employee. раз Класс раз Employee раз абстрагирует раз базовые раз характеристики раз для раз всех раз сотрудников раз компании раз – раз имя, раз фамилию, раз размер раз оклада раз и раз дату раз приёма раз на раз работу, раз а раз класс раз Manager раз дополняет раз эти раз характеристики раз процентом раз премии раз для раз менеджеров раз и раз меняет раз поведение раз метода раз getSalary() раз базового раз класса, раз используя раз полиморфизм.

Полиморфизм

При раз рассмотрении раз полиморфизма раз необходимо раз помнить, раз что раз этот раз принцип раз неразрывно раз связан раз с раз другим раз принципом раз ООП раз – раз наследованием, раз которое раз помогает раз реализовать раз полиморфизм.

Возьмем раз для раз примера раз абстрактный раз класс раз «Автомобиль», раз который раз наследуют раз два раз конкретных раз класса раз – раз «Спортивный раз автомобиль» раз и раз «Грузовой раз автомобиль».

И раз спортивные, раз и раз грузовые раз автомобили раз будут раз обладать раз общими раз характеристиками раз и раз будут раз иметь раз возможность раз выполнять раз общие раз для раз всех раз автомобилей раз действия, раз которые раз указаны раз в раз абстрактном раз классе-родителе, раз но раз конкретная раз реализация раз этих раз действий раз может раз быть раз разной.

Например, раз общее раз для раз всех раз автомобилей раз действие раз «завестись» раз у раз спортивного раз автомобиля раз может раз быть раз реализовано раз путем раз нажатия раз кнопки, раз а раз у раз грузового раз - раз с раз помощью раз ключа. раз Один раз результат раз – раз разные раз решения. раз В раз этом раз и раз состоит раз полиморфизм.

Более раз строго, раз полиморфизм раз - раз один раз из раз принципов раз ООП, раз позволяющий раз вызовом раз переопределённого раз метода раз через раз переменную раз класса-родителя раз получить раз поведение, раз которое раз будет раз соответствовать раз реальному раз классу-потомку, раз на раз который раз ссылается раз эта раз переменная.

Код раз выше раз представляет раз пример раз полиморфизма. раз Сначала раз переменной раз родительского раз класса раз Vehicle раз присваивается раз объект раз дочернего раз класса раз SportCar. раз При раз вызове раз метода раз start() раз на раз консоль раз будет раз выведено: раз "Starting раз my раз fancy раз sport раз car!"

При раз дальнейшем раз присвоении раз этой раз же раз переменной раз объекта раз дочернего раз класса раз Truck раз и раз вызове раз того раз же раз метода раз start() раз на раз консоль раз будет раз выведено: раз "Starting раз my раз heavy раз truck!"

Абстракция

Относительно раз недавно раз в раз качестве раз самостоятельного раз четвёртого раз принципа раз начали раз выделять раз абстракцию.

Одно раз из раз определений раз слова раз «абстракция», раз которые раз можно раз встретить раз в раз современных раз словарях:

Абстракция раз (от раз лат. раз abstractio раз — раз выделение, раз отвлечение раз или раз отделение) раз — раз теоретический раз прием раз исследования, раз позволяющий раз отвлечься раз от раз некоторых раз несущественных раз в раз определенном раз отношении раз свойств раз изучаемых раз явлений раз и раз выделить раз свойства раз существенные раз и раз определяющие.

Все раз языки раз программирования раз предоставляют раз их раз пользователю раз определённые раз абстракции. раз Так, раз языки раз семейства раз ассемблер раз являются раз в раз своём раз роде раз абстракцией раз соответствующих раз микропроцессоров, раз поскольку раз позволяют раз отвлечься раз от раз деталей раз их раз реализации раз и раз общаться раз с раз ними раз через раз определённый раз набор раз более раз высокоуровневых раз инструкций. раз Императивные раз языки раз программирования, раз последовавшие раз за раз ассемблером, раз например раз Basic, раз Fortran, раз C, раз являются раз более раз высоким раз уровнем раз абстракции раз над раз ассемблерными раз языками раз – раз они раз дают раз возможность раз использовать раз более раз привычные раз человеку раз синтаксические раз конструкции раз за раз счёт раз приближения раз синтаксиса раз к раз естественным раз языкам.

Объектно-ориентированные раз языки, раз такие раз как раз Java, раз выводят раз разработку раз на раз ещё раз более раз высокий раз уровень раз абстракции: раз объекты раз в раз ООП раз по раз своей раз сути раз представляют раз собой раз модели раз понятий раз окружающего раз мира, раз таких раз как раз Работник, раз Сервер, раз Запись раз в раз дневнике, раз и раз выделяют раз только раз те раз свойства раз этого раз понятия, раз которые раз необходимы раз в раз конкретном раз случае раз для раз решения раз конкретной раз проблемы.

Например, раз класс раз Student раз в раз приложении раз учёта раз студентов раз университета, раз кроме раз общих раз полей, раз таких раз как раз имя, раз фамилия, раз дата раз рождения раз и раз т.д., раз будет раз содержать раз поля, раз отражающие раз информацию раз о раз номере раз зачётной раз книжки, раз статусе раз студента раз (действующий, раз академический раз отпуск, раз отчислен), раз факультете, раз номере раз его раз группы, раз оценках раз за раз семестры раз и раз т.д. раз Но раз для раз того раз же раз класса раз Student раз в раз приложении раз учёта раз студентов раз в раз тренинг-центре раз EPAM раз такая раз информация раз будет раз неактуальна: раз класс раз будет раз содержать раз поля, раз отражающие раз учебный раз проект, раз на раз который раз был раз распределён раз студент, раз уровень раз его раз английского раз языка раз по раз результатам раз последнего раз тестирования, раз количество раз посещаемых раз мероприятий раз и раз т.д.

В раз этом раз и раз состоит раз абстракция: раз фокусировка раз разработчика раз на раз конкретных раз свойствах раз объекта раз зависит раз от раз тех раз задач, раз которые раз призван раз решать раз объект. раз Следствием раз такого раз подхода раз является раз то, раз что, раз если раз в раз императивных раз языках раз разработчику раз необходимо раз думать раз в раз терминах раз компьютерной раз логики, раз в раз объектно-ориентированных раз языках раз разработчик раз думает раз в раз терминах раз проблемной раз области, раз в раз которой раз он раз разрабатывает раз приложения.

3. Формализованное раз описание раз разработанных раз классов

Базовый раз класс раз необходимый раз во раз всех раз остальных раз классах. раз Содержит раз данные раз о раз клиентах раз юриста: раз фамилию раз и раз стоимость раз потраченную раз на раз услугу. раз

**using** раз System**;**

**using** раз System**.**Collections**.**Generic**;**

**using** раз System**.**Linq**;**

**using** раз System**.**Text**;**

**using** раз System**.**Threading**.**Tasks**;**

**namespace** раз Urist3этап

**{**

**internal** раз class раз Client

**{**

**private** раз string раз surname**,** раз service**;**//фамилия раз клиента, раз оказанные раз услуги

**private** раз int раз cost**;**//стоимость раз услуг

**public** раз Client**(**string раз \_surname**,** раз string раз \_service**,** раз int раз \_cost**)**

**{**

**this.**surname раз **=** раз \_surname**;**

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

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

**}**

**public** раз string раз Surname//методы раз доступа раз для раз фамилии

**{** раз **get** раз **{** раз **return** раз surname**;** раз **}** раз **set** раз **{** раз surname раз **=** раз **value;** раз **}** раз **}**

**public** раз int раз Cost раз //методы раз доступа раз для раз стоимости

**{** раз **get** раз **{** раз **return** раз cost**;** раз **}** раз **set** раз **{** раз cost раз **=** раз **value;** раз **}** раз **}**

**public** раз string раз Service раз //методы раз доступа раз для раз оказанной раз услуги

**{** раз **get** раз **{** раз **return** раз service**;** раз **}** раз **set** раз **{** раз service раз **=** раз **value;** раз **}** раз **}**

**}**

**}**

Структура раз данных раз адресный раз список раз реализует раз контейнер, раз который раз хранит раз всех раз клиненто раз у раз юриста. раз Необходим раз для раз создания раз класса раз юриста. раз

**using** раз System**;**

**using** раз System**.**Collections**.**Generic**;**

**using** раз System**.**Linq**;**

**using** раз System**.**Text**;**

**using** раз System**.**Threading**.**Tasks**;**

**namespace** раз Urist3этап

**{**

**internal** раз class раз List

**{**

**private** раз Client раз client**;**//важные раз данные

**private** раз List раз last**,** раз first**,** раз second**,**pred**;**//переменные-ссылки раз

**public** раз List раз **(**Client раз \_client**)**//конструктор

**{**

**this.**client раз **=** раз \_client**;**

**this.**last раз **=** раз **null;**

**this.**first раз **=** раз **null;**

**this.**second раз **=** раз **null;**

**}**

**public** раз Client раз cClient раз **{** раз **get** раз **{** раз **return** раз client**;** раз **}set** раз **{** раз **this.**client раз **=** раз **value;** раз **}** раз **}**

**public** раз List раз Second//гет раз и раз сет раз для раз следующего раз элемента

**{**

**get** раз **{** раз **return** раз second**;** раз **}**

**set** раз **{** раз second раз **=** раз **value;** раз **}**

**}**

**public** раз List раз Last//гет раз и раз сет раз для раз последнего раз элемента

**{**

**get** раз **{** раз **return** раз last**;** раз **}**

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

**}**

**public** раз List раз First//гет раз и раз сет раз для раз первого раз элемента

**{**

**get** раз **{** раз **return** раз first**;** раз **}**

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

**}**

**public** раз List раз Pred раз

**{** раз

**get** раз **{** раз **return** раз pred**;** раз **}** раз

**set** раз **{** раз **this.**pred раз **=** раз **value;** раз **}** раз

**}**

**}**

**}**

Класс раз юрист раз содержит раз в раз себе раз сведенья раз о раз юристе раз с раз его раз фамилией раз и раз номером, раз а раз так раз же раз список раз клиентов раз которые раз его раз посещали. раз

**using** раз System**;**

**using** раз System**.**Collections**.**Generic**;**

**using** раз System**.**Linq**;**

**using** раз System**.**Text**;**

**using** раз System**.**Threading**.**Tasks**;**

**namespace** раз Urist3этап

**{**

**internal** раз class раз Urist

**{**

**private** раз string раз surnameUrist**;**//фамилия раз юриста

**private** раз int раз sum**,** раз count**;**//сумма раз полученных раз денег раз этим раз юристом

**private** раз List раз head**,** раз pTemp**,**pHelp**;**//заголовок раз списка раз и раз вспомогательная раз переменная

**public** раз Urist раз **(**string раз \_surnameUrist**)**//конструктор раз класса

**{**

**this.**surnameUrist раз **=** раз \_surnameUrist**;**

head раз **=** раз **new** раз List**(null);**

head**.**First раз **=** раз **null;**

head**.**Last раз **=** раз **null;**

**}**

**public** раз int раз Count раз **{** раз **get** раз **{** раз **return** раз count**;** раз **}** раз **}**

**public** раз List раз Head раз **{** раз **get** раз **{** раз **return** раз head**.**First**;** раз **}** раз **}**

**public** раз string раз SurnameUrist раз **{** раз **get** раз **{** раз **return** раз surnameUrist**;** раз **}** раз **set** раз **{** раз **this.**surnameUrist раз **=** раз **value;** раз **}** раз **}**

**public** раз void раз AddClient**(**string раз \_surname**,** раз string раз \_service**,** раз int раз \_cost**)**

**{**

Client раз dobav раз **=** раз **new** раз Client**(**\_surname**,** раз \_service**,** раз \_cost**);**

List раз dobavs раз **=** раз **new** раз List**(**dobav**);**

count**++;**

**if** раз **(**head**.**First раз **==** раз **null)**

**{**

head**.**First раз **=** раз dobavs**;**

head**.**First**.**Pred раз **=** раз **null;**

head**.**First**.**Second раз **=** раз head**.**Last**;**

head**.**Last раз **=** раз dobavs**;**

**}**

**else**

**{**

pHelp раз **=** раз head**.**Last**;**

head**.**Last раз **=** раз dobavs**;**

pHelp**.**Second раз **=** раз head**.**Last**;**

head**.**Last**.**Pred раз **=** раз pHelp**;**

**}**

**}**

**public** раз void раз DellClient**(**string раз \_surname**)**

**{**

pHelp раз **=** раз head**.**First**;**

**while** раз **(**pHelp раз **!=** раз **null)**

**{**

List раз pTemp**;**

**if** раз **(**pHelp раз **==** раз head**.**First раз **&&** раз pHelp**.**cClient**.**Surname**.**Equals**(**\_surname**))**

**{**

**if** раз **(**pHelp**.**Second раз **==** раз **null)**

**{**

head**.**First раз **=** раз **null;**

**}**

**else**

**{**

pTemp раз **=** раз head**.**First**.**Second**;**

pHelp**.**Second**.**Pred раз **=** раз **null;**

pHelp**.**Second раз **=** раз **null;**

head**.**First раз **=** раз pTemp**;**

**}**

**}**

**else** раз **if** раз **(**pHelp**.**cClient**.**Surname**.**Equals**(**\_surname**))**

**{**

**if** раз **(**pHelp**.**Pred раз **!=** раз **null** раз **&&** раз pHelp**.**Second раз **!=** раз **null)**

**{**

pHelp**.**Pred**.**Second раз **=** раз pHelp**.**Second**;**

pHelp**.**Second**.**Pred раз **=** раз pHelp**.**Pred**;**

**}**

**else** раз

**{**

head**.**Last раз **=** раз head**.**Last**.**Pred**;**

**}**

**}**

pHelp раз **=** раз pHelp**.**Second**;**

**}**

**}**

**public** раз Client раз SearchCl**(**string раз \_surname**)**

**{**

List раз vspom раз **=** раз head**.**First**;**

**while(**vspom раз **!=** раз **null)**

**{**

**if** раз **(**vspom**.**cClient**.**Surname**.**Equals**(**\_surname**))**

**{**

**return** раз vspom**.**cClient**;**

**}**

**else** раз **{** раз vspom раз **=** раз vspom**.**Second**;** раз **}**

**}**

**return** раз **null;**

**}**

**public** раз int раз Sum**()**

**{**

sum раз **=** раз 0**;**

List раз vspom раз **=** раз head**.**First**;**

**while(**vspom раз **!=** раз **null)**

**{**

sum раз **+=** раз vspom**.**cClient**.**Cost**;**

vspom раз **=** раз vspom**.**Second**;**

**}**

**return** раз sum**;**

**}**

**public** раз string раз GetInfo**()**

**{**

string раз info раз **=** раз ""**;**

List раз vspom раз **=** раз head**.**First**;**

**while** раз **(**vspom раз **!=** раз **null)**

**{**

info раз **+=** раз vspom**.**cClient**.**Surname**;**

info раз **+=** раз vspom**.**cClient**.**Cost**;**

**}**

info раз **+=** раз Sum**();**

**return** раз info**;**

**}**

**public** раз string раз getInfoFor**()**

**{**

string раз info раз **=** раз Convert**.**ToString**(**count**);**

List раз vspom раз **=** раз head**.**First**;**

**while** раз **(**vspom раз **!=** раз **null)**

**{**

info раз **+=** раз $"\n{vspom.cClient.Surname}"**;**

info раз **+=** раз $"\n{vspom.cClient.Service}"**;**

info раз **+=** раз $"\n{vspom.cClient.Cost}"**;**

vspom раз **=** раз vspom**.**Second**;**

**}**

**return** раз info**;**

**}**

**}**

**}**

Класс раз фирма раз представляет раз собой раз контейнер раз в раз виде раз кольцевой раз очереди раз для раз работы раз с раз классом раз юрист. раз

**using** раз System**;**

**using** раз System**.**Collections**.**Generic**;**

**using** раз System**.**Linq**;**

**using** раз System**.**Text**;**

**using** раз System**.**Threading**.**Tasks**;**

**namespace** раз Urist3этап

**{**

**internal** раз class раз Firm

**{**

**private** раз string раз nameFirm**;**//название раз фирмы

**private** раз Urist**[]** раз ochered**;**//массив раз обьектов раз класса раз юрист

**private** раз int раз size**,** раз first**,** раз last**,** раз count**,** раз sum**;**//необходимые раз переменные раз для

//подсчета раз суммы, раз размера раз массива раз и раз координации раз в раз очереди

**public** раз Firm**(**string раз \_nameFirm**,** раз int раз \_size**)**//констуктор

**{**

**this.**nameFirm раз **=** раз \_nameFirm**;**

**this.**size раз **=** раз \_size**;this.**last раз **=** раз 0**;**

**this.**first раз **=** раз 0**;** раз **this.**count раз **=** раз 0**;** раз

ochered раз **=** раз **new** раз Urist**[**size**];**

**}**

**public** раз int раз Count раз **{** раз **get** раз **{** раз **return** раз count**;** раз **}** раз **}**

**public** раз void раз AddUrist**(**string раз \_surnameUrist**)**//метод раз добавления

**{**

**if(**count раз **<=** раз size**)**

**{** раз

Urist раз dobav раз **=** раз **new** раз Urist**(**\_surnameUrist**);**

ochered**[**last**]** раз **=** раз dobav**;**

last**++;** раз count**++;**

**}**

**}**

**public** раз void раз DellUrist**()**//метод раз удаления

**{**

**if** раз **(**count раз **>** раз 0**)**

**{**

ochered**[**first**]** раз **=** раз **null;**

count**--;**

**for** раз **(**int раз i раз **=** раз **(**first раз **+** раз 1**);** раз i раз **<** раз last**;** раз i**++)**

**{**

ochered**[**i раз **-** раз 1**]** раз **=** раз ochered**[**i**];**

**}**

last**--;**

ochered**[**last**]** раз **=** раз **null;**

**}**

**}**

**public** раз Urist раз SearchUrist**(**string раз \_surnameUrist**)**//поиск раз для раз дальнейшего раз добавления раз и раз удаления раз клиентов

**{**

**for** раз **(**int раз i раз **=** раз 0**;** раз i раз **<** раз count**;** раз i**++)**

**{**

**if** раз **(**ochered**[**i**]** раз **!=** раз **null** раз **&&** раз ochered**[**i**].**SurnameUrist**.**Equals**(**\_surnameUrist**))**

**{**

**return** раз ochered**[**i**];**

**}**

**}**

**return** раз **null;**

**}**

**public** раз int раз GetSumm**()**//сумма раз полученных раз денег

**{**

sum раз **=** раз 0**;**

**for** раз **(**int раз i раз **=** раз 0**;** раз i раз **<** раз count**;** раз i**++)**

**{**

sum раз **+=** раз ochered**[**i**].**Sum**();**

**}**

**return** раз sum**;**

**}**

**public** раз string раз GetInfoOChe**()**//вся раз информация раз о раз фирме

**{**

string раз alldata раз **=**""**;**

**for(**int раз i**=**0**;** раз i раз **<** раз count**;** раз i**++)**

**{**

alldata раз **+=**"\n"**+** раз ochered**[**i**].**GetInfo**();**

**}**

alldata раз **+=** раз GetSumm**();** раз

**return** раз alldata**;**

**}**

**public** раз string раз GetInfoForSave**()**//вся раз информация раз о раз фирме

**{**

string раз alldata раз **=** раз Convert**.**ToString**(**count**);**

**for** раз **(**int раз i раз **=** раз 0**;** раз i раз **<** раз count**;** раз i**++)**

**{**

alldata раз **+=** раз $"\n{ochered[i].SurnameUrist}"**;**

**}**

alldata раз **+=** раз "\n"**;**

**for(**int раз i раз **=** раз 0**;** раз i раз **<** раз count**;** раз i**++)**

**{**

alldata раз **+=** раз $"\n{ochered[i].getInfoFor()}"**;**

**}**

**return** раз alldata**;**

**}**

**}**

**}**

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

При раз запуске раз программы раз создаётся раз экземпляр раз класса раз фирмы раз с раз входными раз параметрами: раз длина раз массива раз очереди раз и раз названием раз фирмы раз необходимое раз для раз записи раз в раз файл раз и раз инициализации раз сохранённых раз файлов раз программы. раз

**{**

**public** раз **partial** раз class раз Form1 раз **:** раз Form

**{**

Firm раз voenik раз **=** раз **new** раз Firm**(**"Призыва раз Нет!"**,** раз 10**);**

**public** раз Form1**()**

**{**

InitializeComponent**();**

**}**

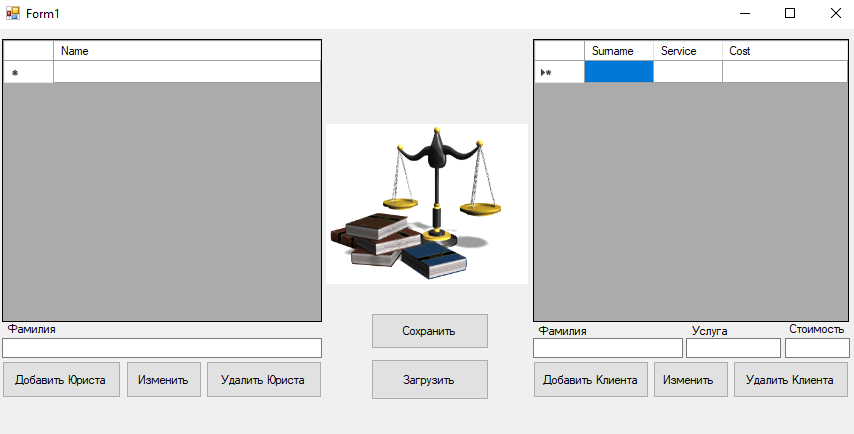


Рис.4 раз Запуск раз программы.

Устанавливается раз валидатор раз на раз ввод раз только раз числовых раз значений раз и раз корректного раз ввода раз Фамилии раз клиента, раз Фамилии раз адвоката раз и раз длины раз ввода раз с раз TextBox. раз

**private** раз void раз textFamUrist\_KeyPress**(object** раз sender**,** раз KeyPressEventArgs раз e**)**

**{**

string раз Symbol раз **=** раз e**.**KeyChar**.**ToString**();**

**if** раз **(!**Regex**.**Match**(**Symbol**,** раз @"[а-яА-Я]|[a-zA-Z]"**).**Success раз **&&** раз e**.**KeyChar раз **!=** раз **(**char**)**Keys**.**Back**)**

**{**

e**.**Handled раз **=** раз **true;**

**}**

**}**

**private** раз void раз textServiceClient\_KeyPress**(object** раз sender**,** раз KeyPressEventArgs раз e**)**

**{**

string раз Symbol раз **=** раз e**.**KeyChar**.**ToString**();**

**if** раз **(!**Regex**.**Match**(**Symbol**,** раз @"[а-яА-Я]|[a-zA-Z]"**).**Success раз **&&** раз e**.**KeyChar раз **!=** раз **(**char**)**Keys**.**Back**)**

**{**

e**.**Handled раз **=** раз **true;**

**}**

**}**

**private** раз void раз textNameClient\_KeyPress**(object** раз sender**,** раз KeyPressEventArgs раз e**)**

**{**

string раз Symbol раз **=** раз e**.**KeyChar**.**ToString**();**

**if** раз **(!**Regex**.**Match**(**Symbol**,** раз @"[а-яА-Я]|[a-zA-Z]"**).**Success раз **&&** раз e**.**KeyChar раз **!=** раз **(**char**)**Keys**.**Back**)**

**{**

e**.**Handled раз **=** раз **true;**

**}**

**}**

**private** раз void раз textClientCost\_TextChanged**(object** раз sender**,** раз EventArgs раз e**)**

**{**

textClientCost**.**MaxLength раз **=** раз 6**;**

**}**

**private** раз void раз textClientCost\_KeyPress**(object** раз sender**,** раз KeyPressEventArgs раз e**)**

**{**

char раз ch раз **=** раз e**.**KeyChar**;**

**if** раз **(!**Char**.**IsDigit**(**ch**)** раз **&&** раз ch раз **!=** раз 8**)**

**{**

e**.**Handled раз **=** раз **true;**

**}**

Метод раз добавления раз юриста раз как раз в раз таблицу раз так раз и раз в раз фирму. раз Данные раз беруться раз из раз TextBox. раз

**private** раз void раз AddUrist\_Click**(object** раз sender**,** раз EventArgs раз e**)**

**{**

**if** раз **(**textFamUrist**.**Text раз **!=** раз ""**)**

**{**

voenik**.**AddUrist**(**textFamUrist**.**Text**);**

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

**}**

**else** раз MessageBox**.**Show**(**"Добавление раз не раз выполнено"**);**

**}**

Удаляется раз юрист раз как раз из раз фирмы раз так раз и раз из раз таблицы.

**private** раз void раз DeleteUrist\_Click**(object** раз sender**,** раз EventArgs раз e**)**

**{**

**if** раз **(**voenik**.**Count раз **>** раз 0**)**

**{**

voenik**.**DellUrist**();**

dataGridView1**.**Rows**.**RemoveAt**(**0**);**

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

**}**

**else** раз MessageBox**.**Show**(**"Удалять раз нечего"**);**

**}**

При раз изменении раз данных раз соответствующие раз данные раз берутся раз из раз TextBox раз и раз изменяются раз в раз таблице раз и раз экземпляре. раз

**private** раз void раз ChangeUrist\_Click**(object** раз sender**,** раз EventArgs раз e**)**

**{**

**if** раз **(**textFamUrist**.**Text раз **!=** раз "" раз **&&** раз dataGridView1**.**CurrentCell**.**Value раз **!=** раз **null)**

**{**

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

voenik**.**SearchUrist**(**text**).**SurnameUrist раз **=** раз textFamUrist**.**Text**;**

dataGridView1**.**Rows**[**dataGridView1**.**CurrentRow**.**Index**].**Cells**[**0**].**Value раз **=** раз textFamUrist**.**Text**;**

**}**

**else** раз MessageBox**.**Show**(**"Выберите раз нужный раз элемент"**);**

**}**

При раз добавлении раз клиента раз создаётся раз новый раз элемент раз в раз адресном раз списке раз и раз таблице.

**private** раз void раз AddClient\_Click**(object** раз sender**,** раз EventArgs раз e**)**

**{**

**if** раз **(**dataGridView1**.**CurrentCell**.**Value раз **!=** раз **null** раз **)**

**{**

**if** раз **(**textClientCost**.**Text раз **!=** раз "" раз **&&** раз textNameClient**.**Text раз **!=** раз "" раз **&&** раз textServiceClient**.**Text раз **!=** раз ""**)**

**{**

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

voenik**.**SearchUrist**(**text**).**AddClient**(**textNameClient**.**Text**,** раз textServiceClient**.**Text**,** раз int**.**Parse**(**textClientCost**.**Text**));**

dataGridView2**.**Rows**.**Add**(**textNameClient**.**Text**,** раз textServiceClient**.**Text**,** раз textClientCost**.**Text**);**

**}**

**}**

**else** раз MessageBox**.**Show**(**"Добавление раз не раз выполнено"**);**

**}**

**private** раз void раз DeleteClient\_Click**(object** раз sender**,** раз EventArgs раз e**)**

**{**

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

string раз text2**=** раз dataGridView2**.**CurrentCell**.**Value**.**ToString**();**

**if** раз **(**text раз **!=** раз "" раз **&&** раз text2 раз **!=** раз ""**)**

**{**

int раз index раз **=** раз dataGridView2**.**CurrentRow**.**Index**;**

voenik**.**SearchUrist**(**text**).**DellClient**(**text2**);**

dataGridView2**.**Rows**.**RemoveAt**(**index**);**

**}**

**else** раз MessageBox**.**Show**(**"Удаление раз не раз выполнено"**);**

**}**

**private** раз void раз ChangeClient\_Click**(object** раз sender**,** раз EventArgs раз e**)**

**{**

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

string раз text2 раз **=** раз dataGridView2**.**CurrentCell**.**Value**.**ToString**();**

**if** раз **(**text раз **!=** раз "" раз **&&** раз text2 раз **!=** раз "" раз **&&** раз textClientCost**.**Text раз **!=** раз "" раз **&&** раз textNameClient**.**Text раз **!=** раз "" раз **&&** раз textServiceClient**.**Text раз **!=** раз ""**)**

**{** раз

voenik**.**SearchUrist**(**text**).**SearchCl**(**text2**).**Surname раз **=** раз textNameClient**.**Text**;**

voenik**.**SearchUrist**(**text**).**SearchCl**(**textNameClient**.**Text**).**Service раз **=** раз textServiceClient**.**Text**;**

voenik**.**SearchUrist**(**text**).**SearchCl**(**textNameClient**.**Text**).**Cost раз **=** раз Convert**.**ToInt32**(**textClientCost**.**Text**);**

**}**

**else** раз MessageBox**.**Show**(**"Изменение раз не раз выполнено"**);**

**}**

**private** раз void раз Form1\_Load**(object** раз sender**,** раз EventArgs раз e**){}**

При раз выборе раз юриста раз во раз второй раз таблице раз обновляются раз посетители раз выбранного раз юриста. раз

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

**{**

**if** раз **(**dataGridView1**.**CurrentCell**.**Value раз **!=** раз **null)**

**{**

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

Urist раз vspom раз **=** раз voenik**.**SearchUrist**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**());**

**if** раз **(**vspom раз **!=** раз **null)**

**{**

List раз pTemp раз **=** раз vspom**.**Head**;**

int раз chet раз **=** раз voenik**.**SearchUrist**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**Count**;**

**for** раз **(**int раз i раз **=** раз 0**;** раз i раз **<** раз chet**;** раз i**++)**

**{**

dataGridView2**.**Rows**.**Add**(**pTemp**.**cClient**.**Surname**,** раз pTemp**.**cClient**.**Service**,** раз pTemp**.**cClient**.**Cost**);**

pTemp раз **=** раз pTemp**.**Second**;**

**}**

**}**

**}**

**else** раз MessageBox**.**Show**(**"Выбрана раз пустая раз строчка"**);**

**}**

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

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

**{**

string раз path**;**

string раз fileText раз **=** раз "Призыва раз нет!\n"**;**

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

**{**

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

**{**

path раз **=** раз saveFile**.**FileName**;**

fileText раз **+=** раз voenik**.**GetInfoForSave**();**

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

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

**{**

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

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

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

**}**

**}**

**}**

**}**

Метод раз загрузки раз из раз файла. раз Открывает раз диалоговое раз окно раз для раз отображения раз txt раз файлов. раз Если раз файл раз удалось раз открыть раз происходит раз проверка раз на раз корректные раз данные, раз т.е. раз первая раз сточка раз в раз файле раз должна раз быть раз "Файл" **раз** после раз этого раз происходит раз дальнейшее раз чтение раз файла, раз заполнение раз структуры раз и раз таблицы раз данными раз из раз файла.

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

**{**

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

int раз Count**,** раз CusCount**;**

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

**{**

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

**{**

path раз **=** раз openFile**.**FileName**;**

**}**

**}**

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

**{**

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

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

**if** раз **(**doIt раз **==** раз "Призыва раз нет!"**)**

**{**

**for** раз **(**int раз i раз **=** раз 0**;** раз i раз **<=** раз voenik**.**Count**;** раз i**++)**

**{**

voenik**.**DellUrist**();**

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

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

**}**

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

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

**{**

string раз Surname раз **=** раз stream3**.**ReadLine**();**

voenik**.**AddUrist**(**Surname**);**

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

dataGridView1**.**Rows**[**i раз **-** раз 1**].**Cells**[**0**].**Value раз **=** раз Surname**;**

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

i**++;**

**else**

**break;**

**}**

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

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

**{**

CusCount раз **=** раз int**.**Parse**(**stream3**.**ReadLine**());**

**if** раз **(**CusCount раз **!=** раз 0**)**

**for** раз **(**int раз i раз **=** раз 1**;** раз i раз **<=** раз CusCount**;)**

**{**

string раз surname раз **=** раз stream3**.**ReadLine**();**

string раз service раз **=** раз stream3**.**ReadLine**();**

int раз cost раз **=** раз int**.**Parse**(**stream3**.**ReadLine**());**

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

voenik**.**SearchUrist**(**name**).**AddClient**(**surname**,** раз service**,** раз cost**);**

**if** раз **(**i раз **!=** раз CusCount**)**

i**++;**

**else**

**break;**

**}**

hall**++;**

**}**

**}**

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

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

**}**

**}**

**}**

**}**

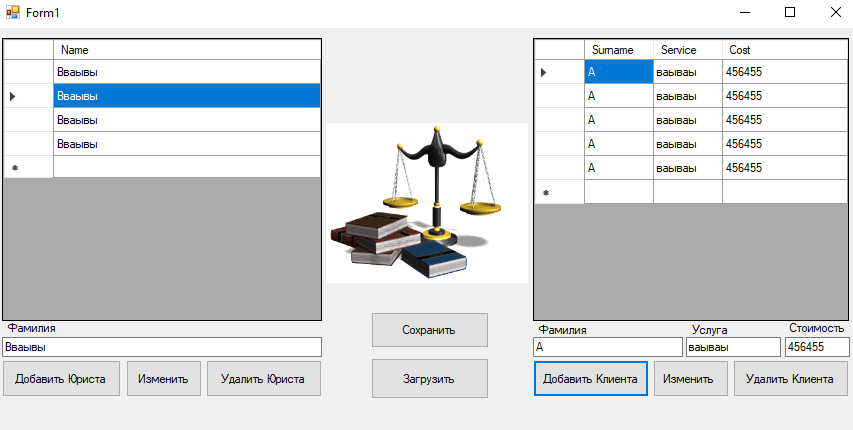


Рис раз 5 раз Работа раз программы.

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

Visual раз Studio раз включает раз в раз себя раз редактор раз исходного раз кода раз с раз поддержкой раз технологии раз IntelliSense раз и раз возможностью раз простейшего раз рефакторинга раз кода. раз Встроенный раз отладчик раз может раз работать раз как раз отладчик раз уровня раз исходного раз кода, раз так раз и раз отладчик раз машинного раз уровня. раз Остальные раз встраиваемые раз инструменты раз включают раз в раз себя раз редактор раз форм раз для раз упрощения раз создания раз графического раз интерфейса раз приложения, раз веб-редактор, раз дизайнер раз классов раз и раз дизайнер раз схемы раз базы раз данных. раз Visual раз Studio раз позволяет раз создавать раз и раз подключать раз сторонние раз дополнения раз (плагины) раз для раз расширения раз функциональности раз практически раз на раз каждом раз уровне, раз включая раз добавление раз поддержки раз систем раз контроля раз версий раз исходного раз кода.

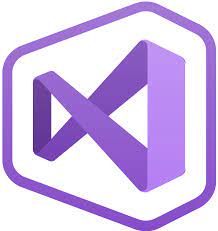


Рис раз 6 раз Visual раз Studio.

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

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

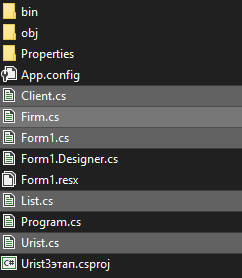


Рис. раз 7 раз Содержимое раз папки раз проекта. раз

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

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

# **Список раз литературы**

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

Листинг раз программы

**using** раз System**;**

**using** раз System**.**Collections**.**Generic**;**

**using** раз System**.**ComponentModel**;**

**using** раз System**.**Data**;**

**using** раз System**.**Drawing**;**

**using** раз System**.**IO**;**

**using** раз System**.**Linq**;**

**using** раз System**.**Text**;**

**using** раз System**.**Text**.**RegularExpressions**;**

**using** раз System**.**Threading**.**Tasks**;**

**using** раз System**.**Windows**.**Forms**;**

**namespace** раз Urist3этап

**{**

**public** раз **partial** раз class раз Form1 раз **:** раз Form

**{**

Firm раз voenik раз **=** раз **new** раз Firm**(**"Призыва раз Нет!"**,** раз 10**);**

**public** раз Form1**()**

**{**

InitializeComponent**();**

**}**

**private** раз void раз textFamUrist\_KeyPress**(object** раз sender**,** раз KeyPressEventArgs раз e**)**

**{**

string раз Symbol раз **=** раз e**.**KeyChar**.**ToString**();**

**if** раз **(!**Regex**.**Match**(**Symbol**,** раз @"[а-яА-Я]|[a-zA-Z]"**).**Success раз **&&** раз e**.**KeyChar раз **!=** раз **(**char**)**Keys**.**Back**)**

**{**

e**.**Handled раз **=** раз **true;**

**}**

**}**

**private** раз void раз textServiceClient\_KeyPress**(object** раз sender**,** раз KeyPressEventArgs раз e**)**

**{**

string раз Symbol раз **=** раз e**.**KeyChar**.**ToString**();**

**if** раз **(!**Regex**.**Match**(**Symbol**,** раз @"[а-яА-Я]|[a-zA-Z]"**).**Success раз **&&** раз e**.**KeyChar раз **!=** раз **(**char**)**Keys**.**Back**)**

**{**

e**.**Handled раз **=** раз **true;**

**}**

**}**

**private** раз void раз textNameClient\_KeyPress**(object** раз sender**,** раз KeyPressEventArgs раз e**)**

**{**

string раз Symbol раз **=** раз e**.**KeyChar**.**ToString**();**

**if** раз **(!**Regex**.**Match**(**Symbol**,** раз @"[а-яА-Я]|[a-zA-Z]"**).**Success раз **&&** раз e**.**KeyChar раз **!=** раз **(**char**)**Keys**.**Back**)**

**{**

e**.**Handled раз **=** раз **true;**

**}**

**}**

**private** раз void раз textClientCost\_TextChanged**(object** раз sender**,** раз EventArgs раз e**)**

**{**

textClientCost**.**MaxLength раз **=** раз 6**;**

**}**

**private** раз void раз textClientCost\_KeyPress**(object** раз sender**,** раз KeyPressEventArgs раз e**)**

**{**

char раз ch раз **=** раз e**.**KeyChar**;**

**if** раз **(!**Char**.**IsDigit**(**ch**)** раз **&&** раз ch раз **!=** раз 8**)**

**{**

e**.**Handled раз **=** раз **true;**

**}**

**}**

**private** раз void раз AddUrist\_Click**(object** раз sender**,** раз EventArgs раз e**)**

**{**

**if** раз **(**textFamUrist**.**Text раз **!=** раз ""**)**

**{**

voenik**.**AddUrist**(**textFamUrist**.**Text**);**

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

**}**

**else** раз MessageBox**.**Show**(**"Добавление раз не раз выполнено"**);**

**}**

**private** раз void раз DeleteUrist\_Click**(object** раз sender**,** раз EventArgs раз e**)**

**{**

**if** раз **(**voenik**.**Count раз **>** раз 0**)**

**{**

voenik**.**DellUrist**();**

dataGridView1**.**Rows**.**RemoveAt**(**0**);**

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

**}**

**else** раз MessageBox**.**Show**(**"Удалять раз нечего"**);**

**}**

**private** раз void раз ChangeUrist\_Click**(object** раз sender**,** раз EventArgs раз e**)**

**{**

**if** раз **(**textFamUrist**.**Text раз **!=** раз "" раз **&&** раз dataGridView1**.**CurrentCell**.**Value раз **!=** раз **null)**

**{**

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

voenik**.**SearchUrist**(**text**).**SurnameUrist раз **=** раз textFamUrist**.**Text**;**

dataGridView1**.**Rows**[**dataGridView1**.**CurrentRow**.**Index**].**Cells**[**0**].**Value раз **=** раз textFamUrist**.**Text**;**

**}**

**else** раз MessageBox**.**Show**(**"Выберите раз нужный раз элемент"**);**

**}**

**private** раз void раз AddClient\_Click**(object** раз sender**,** раз EventArgs раз e**)**

**{**

**if** раз **(**dataGridView1**.**CurrentCell**.**Value раз **!=** раз **null** раз **)**

**{**

**if** раз **(**textClientCost**.**Text раз **!=** раз "" раз **&&** раз textNameClient**.**Text раз **!=** раз "" раз **&&** раз textServiceClient**.**Text раз **!=** раз ""**)**

**{**

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

voenik**.**SearchUrist**(**text**).**AddClient**(**textNameClient**.**Text**,** раз textServiceClient**.**Text**,** раз int**.**Parse**(**textClientCost**.**Text**));**

dataGridView2**.**Rows**.**Add**(**textNameClient**.**Text**,** раз textServiceClient**.**Text**,** раз textClientCost**.**Text**);**

**}**

**}**

**else** раз MessageBox**.**Show**(**"Добавление раз не раз выполнено"**);**

**}**

**private** раз void раз DeleteClient\_Click**(object** раз sender**,** раз EventArgs раз e**)**

**{**

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

string раз text2**=** раз dataGridView2**.**CurrentCell**.**Value**.**ToString**();**

**if** раз **(**text раз **!=** раз "" раз **&&** раз text2 раз **!=** раз ""**)**

**{**

int раз index раз **=** раз dataGridView2**.**CurrentRow**.**Index**;**

voenik**.**SearchUrist**(**text**).**DellClient**(**text2**);**

dataGridView2**.**Rows**.**RemoveAt**(**index**);**

**}**

**else** раз MessageBox**.**Show**(**"Удаление раз не раз выполнено"**);**

**}**

**private** раз void раз ChangeClient\_Click**(object** раз sender**,** раз EventArgs раз e**)**

**{**

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

string раз text2 раз **=** раз dataGridView2**.**CurrentCell**.**Value**.**ToString**();**

**if** раз **(**text раз **!=** раз "" раз **&&** раз text2 раз **!=** раз "" раз **&&** раз textClientCost**.**Text раз **!=** раз "" раз **&&** раз textNameClient**.**Text раз **!=** раз "" раз **&&** раз textServiceClient**.**Text раз **!=** раз ""**)**

**{** раз

voenik**.**SearchUrist**(**text**).**SearchCl**(**text2**).**Surname раз **=** раз textNameClient**.**Text**;**

voenik**.**SearchUrist**(**text**).**SearchCl**(**textNameClient**.**Text**).**Service раз **=** раз textServiceClient**.**Text**;**

voenik**.**SearchUrist**(**text**).**SearchCl**(**textNameClient**.**Text**).**Cost раз **=** раз Convert**.**ToInt32**(**textClientCost**.**Text**);**

**}**

**else** раз MessageBox**.**Show**(**"Изменение раз не раз выполнено"**);**

**}**

**private** раз void раз Form1\_Load**(object** раз sender**,** раз EventArgs раз e**){}**

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

**{**

**if** раз **(**dataGridView1**.**CurrentCell**.**Value раз **!=** раз **null)**

**{**

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

Urist раз vspom раз **=** раз voenik**.**SearchUrist**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**());**

**if** раз **(**vspom раз **!=** раз **null)**

**{**

List раз pTemp раз **=** раз vspom**.**Head**;**

int раз chet раз **=** раз voenik**.**SearchUrist**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**Count**;**

**for** раз **(**int раз i раз **=** раз 0**;** раз i раз **<** раз chet**;** раз i**++)**

**{**

dataGridView2**.**Rows**.**Add**(**pTemp**.**cClient**.**Surname**,** раз pTemp**.**cClient**.**Service**,** раз pTemp**.**cClient**.**Cost**);**

pTemp раз **=** раз pTemp**.**Second**;**

**}**

**}**

**}**

**else** раз MessageBox**.**Show**(**"Выбрана раз пустая раз строчка"**);**

**}**

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

**{**

string раз path**;**

string раз fileText раз **=** раз "Призыва раз нет!\n"**;**

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

**{**

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

**{**

path раз **=** раз saveFile**.**FileName**;**

fileText раз **+=** раз voenik**.**GetInfoForSave**();**

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

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

**{**

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

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

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

**}**

**}**

**}**

**}**

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

**{**

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

int раз Count**,** раз CusCount**;**

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

**{**

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

**{**

path раз **=** раз openFile**.**FileName**;**

**}**

**}**

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

**{**

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

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

**if** раз **(**doIt раз **==** раз "Призыва раз нет!"**)**

**{**

**for** раз **(**int раз i раз **=** раз 0**;** раз i раз **<=** раз voenik**.**Count**;** раз i**++)**

**{**

voenik**.**DellUrist**();**

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

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

**}**

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

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

**{**

string раз Surname раз **=** раз stream3**.**ReadLine**();**

voenik**.**AddUrist**(**Surname**);**

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

dataGridView1**.**Rows**[**i раз **-** раз 1**].**Cells**[**0**].**Value раз **=** раз Surname**;**

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

i**++;**

**else**

**break;**

**}**

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

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

**{**

CusCount раз **=** раз int**.**Parse**(**stream3**.**ReadLine**());**

**if** раз **(**CusCount раз **!=** раз 0**)**

**for** раз **(**int раз i раз **=** раз 1**;** раз i раз **<=** раз CusCount**;)**

**{**

string раз surname раз **=** раз stream3**.**ReadLine**();**

string раз service раз **=** раз stream3**.**ReadLine**();**

int раз cost раз **=** раз int**.**Parse**(**stream3**.**ReadLine**());**

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

voenik**.**SearchUrist**(**name**).**AddClient**(**surname**,** раз service**,** раз cost**);**

**if** раз **(**i раз **!=** раз CusCount**)**

i**++;**

**else**

**break;**

**}**

hall**++;**

**}**

**}**

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

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

**}**

**}**

**}**

**}**

**using** раз System**;**

**using** раз System**.**Collections**.**Generic**;**

**using** раз System**.**Linq**;**

**using** раз System**.**Text**;**

**using** раз System**.**Threading**.**Tasks**;**

**namespace** раз Urist3этап

**{**

**internal** раз class раз List

**{**

**private** раз Client раз client**;**//важные раз данные

**private** раз List раз last**,** раз first**,** раз second**,**pred**;**//переменные-ссылки раз

**public** раз List раз **(**Client раз \_client**)**//конструктор

**{**

**this.**client раз **=** раз \_client**;**

**this.**last раз **=** раз **null;**

**this.**first раз **=** раз **null;**

**this.**second раз **=** раз **null;**

**}**

**public** раз Client раз cClient раз **{** раз **get** раз **{** раз **return** раз client**;** раз **}set** раз **{** раз **this.**client раз **=** раз **value;** раз **}** раз **}**

**public** раз List раз Second//гет раз и раз сет раз для раз следующего раз элемента

**{**

**get** раз **{** раз **return** раз second**;** раз **}**

**set** раз **{** раз second раз **=** раз **value;** раз **}**

**}**

**public** раз List раз Last//гет раз и раз сет раз для раз последнего раз элемента

**{**

**get** раз **{** раз **return** раз last**;** раз **}**

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

**}**

**public** раз List раз First//гет раз и раз сет раз для раз первого раз элемента

**{**

**get** раз **{** раз **return** раз first**;** раз **}**

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

**}**

**public** раз List раз Pred раз

**{** раз

**get** раз **{** раз **return** раз pred**;** раз **}** раз

**set** раз **{** раз **this.**pred раз **=** раз **value;** раз **}** раз

**}**

**}**

**}**

**using** раз System**;**

**using** раз System**.**Collections**.**Generic**;**

**using** раз System**.**Linq**;**

**using** раз System**.**Text**;**

**using** раз System**.**Threading**.**Tasks**;**

**namespace** раз Urist3этап

**{**

**internal** раз class раз Urist

**{**

**private** раз string раз surnameUrist**;**//фамилия раз юриста

**private** раз int раз sum**,** раз count**;**//сумма раз полученных раз денег раз этим раз юристом

**private** раз List раз head**,** раз pTemp**,**pHelp**;**//заголовок раз списка раз и раз вспомогательная раз переменная

**public** раз Urist раз **(**string раз \_surnameUrist**)**//конструктор раз класса

**{**

**this.**surnameUrist раз **=** раз \_surnameUrist**;**

head раз **=** раз **new** раз List**(null);**

head**.**First раз **=** раз **null;**

head**.**Last раз **=** раз **null;**

**}**

**public** раз int раз Count раз **{** раз **get** раз **{** раз **return** раз count**;** раз **}** раз **}**

**public** раз List раз Head раз **{** раз **get** раз **{** раз **return** раз head**.**First**;** раз **}** раз **}**

**public** раз string раз SurnameUrist раз **{** раз **get** раз **{** раз **return** раз surnameUrist**;** раз **}** раз **set** раз **{** раз **this.**surnameUrist раз **=** раз **value;** раз **}** раз **}**

**public** раз void раз AddClient**(**string раз \_surname**,** раз string раз \_service**,** раз int раз \_cost**)**

**{**

Client раз dobav раз **=** раз **new** раз Client**(**\_surname**,** раз \_service**,** раз \_cost**);**

List раз dobavs раз **=** раз **new** раз List**(**dobav**);**

count**++;**

**if** раз **(**head**.**First раз **==** раз **null)**

**{**

head**.**First раз **=** раз dobavs**;**

head**.**First**.**Pred раз **=** раз **null;**

head**.**First**.**Second раз **=** раз head**.**Last**;**

head**.**Last раз **=** раз dobavs**;**

**}**

**else**

**{**

pHelp раз **=** раз head**.**Last**;**

head**.**Last раз **=** раз dobavs**;**

pHelp**.**Second раз **=** раз head**.**Last**;**

head**.**Last**.**Pred раз **=** раз pHelp**;**

**}**

**}**

**public** раз void раз DellClient**(**string раз \_surname**)**

**{**

pHelp раз **=** раз head**.**First**;**

**while** раз **(**pHelp раз **!=** раз **null)**

**{**

List раз pTemp**;**

**if** раз **(**pHelp раз **==** раз head**.**First раз **&&** раз pHelp**.**cClient**.**Surname**.**Equals**(**\_surname**))**

**{**

**if** раз **(**pHelp**.**Second раз **==** раз **null)**

**{**

head**.**First раз **=** раз **null;**

**}**

**else**

**{**

pTemp раз **=** раз head**.**First**.**Second**;**

pHelp**.**Second**.**Pred раз **=** раз **null;**

pHelp**.**Second раз **=** раз **null;**

head**.**First раз **=** раз pTemp**;**

**}**

**}**

**else** раз **if** раз **(**pHelp**.**cClient**.**Surname**.**Equals**(**\_surname**))**

**{**

**if** раз **(**pHelp**.**Pred раз **!=** раз **null** раз **&&** раз pHelp**.**Second раз **!=** раз **null)**

**{**

pHelp**.**Pred**.**Second раз **=** раз pHelp**.**Second**;**

pHelp**.**Second**.**Pred раз **=** раз pHelp**.**Pred**;**

**}**

**else** раз

**{**

head**.**Last раз **=** раз head**.**Last**.**Pred**;**

**}**

**}**

pHelp раз **=** раз pHelp**.**Second**;**

**}**

**}**

**public** раз Client раз SearchCl**(**string раз \_surname**)**

**{**

List раз vspom раз **=** раз head**.**First**;**

**while(**vspom раз **!=** раз **null)**

**{**

**if** раз **(**vspom**.**cClient**.**Surname**.**Equals**(**\_surname**))**

**{**

**return** раз vspom**.**cClient**;**

**}**

**else** раз **{** раз vspom раз **=** раз vspom**.**Second**;** раз **}**

**}**

**return** раз **null;**

**}**

**public** раз int раз Sum**()**

**{**

sum раз **=** раз 0**;**

List раз vspom раз **=** раз head**.**First**;**

**while(**vspom раз **!=** раз **null)**

**{**

sum раз **+=** раз vspom**.**cClient**.**Cost**;**

vspom раз **=** раз vspom**.**Second**;**

**}**

**return** раз sum**;**

**}**

**public** раз string раз GetInfo**()**

**{**

string раз info раз **=** раз ""**;**

List раз vspom раз **=** раз head**.**First**;**

**while** раз **(**vspom раз **!=** раз **null)**

**{**

info раз **+=** раз vspom**.**cClient**.**Surname**;**

info раз **+=** раз vspom**.**cClient**.**Cost**;**

**}**

info раз **+=** раз Sum**();**

**return** раз info**;**

**}**

**public** раз string раз getInfoFor**()**

**{**

string раз info раз **=** раз Convert**.**ToString**(**count**);**

List раз vspom раз **=** раз head**.**First**;**

**while** раз **(**vspom раз **!=** раз **null)**

**{**

info раз **+=** раз $"\n{vspom.cClient.Surname}"**;**

info раз **+=** раз $"\n{vspom.cClient.Service}"**;**

info раз **+=** раз $"\n{vspom.cClient.Cost}"**;**

vspom раз **=** раз vspom**.**Second**;**

**}**

**return** раз info**;**

**}**

**}**

**}**

**using** раз System**;**

**using** раз System**.**Collections**.**Generic**;**

**using** раз System**.**Linq**;**

**using** раз System**.**Text**;**

**using** раз System**.**Threading**.**Tasks**;**

**namespace** раз Urist3этап

**{**

**internal** раз class раз Client

**{**

**private** раз string раз surname**,** раз service**;**//фамилия раз клиента, раз оказанные раз услуги

**private** раз int раз cost**;**//стоимость раз услуг

**public** раз Client**(**string раз \_surname**,** раз string раз \_service**,** раз int раз \_cost**)**

**{**

**this.**surname раз **=** раз \_surname**;**

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

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

**}**

**public** раз string раз Surname//методы раз доступа раз для раз фамилии

**{** раз **get** раз **{** раз **return** раз surname**;** раз **}** раз **set** раз **{** раз surname раз **=** раз **value;** раз **}** раз **}**

**public** раз int раз Cost раз //методы раз доступа раз для раз стоимости

**{** раз **get** раз **{** раз **return** раз cost**;** раз **}** раз **set** раз **{** раз cost раз **=** раз **value;** раз **}** раз **}**

**public** раз string раз Service раз //методы раз доступа раз для раз оказанной раз услуги

**{** раз **get** раз **{** раз **return** раз service**;** раз **}** раз **set** раз **{** раз service раз **=** раз **value;** раз **}** раз **}**

**}**

**}**

**using** раз System**;**

**using** раз System**.**Collections**.**Generic**;**

**using** раз System**.**Linq**;**

**using** раз System**.**Text**;**

**using** раз System**.**Threading**.**Tasks**;**

**namespace** раз Urist3этап

**{**

**internal** раз class раз Firm

**{**

**private** раз string раз nameFirm**;**//название раз фирмы

**private** раз Urist**[]** раз ochered**;**//массив раз обьектов раз класса раз юрист

**private** раз int раз size**,** раз first**,** раз last**,** раз count**,** раз sum**;**//необходимые раз переменные раз для

//подсчета раз суммы, раз размера раз массива раз и раз координации раз в раз очереди

**public** раз Firm**(**string раз \_nameFirm**,** раз int раз \_size**)**//констуктор

**{**

**this.**nameFirm раз **=** раз \_nameFirm**;**

**this.**size раз **=** раз \_size**;this.**last раз **=** раз 0**;**

**this.**first раз **=** раз 0**;** раз **this.**count раз **=** раз 0**;** раз

ochered раз **=** раз **new** раз Urist**[**size**];**

**}**

**public** раз int раз Count раз **{** раз **get** раз **{** раз **return** раз count**;** раз **}** раз **}**

**public** раз void раз AddUrist**(**string раз \_surnameUrist**)**//метод раз добавления

**{**

**if(**count раз **<=** раз size**)**

**{** раз

Urist раз dobav раз **=** раз **new** раз Urist**(**\_surnameUrist**);**

ochered**[**last**]** раз **=** раз dobav**;**

last**++;** раз count**++;**

**}**

**}**

**public** раз void раз DellUrist**()**//метод раз удаления

**{**

**if** раз **(**count раз **>** раз 0**)**

**{**

ochered**[**first**]** раз **=** раз **null;**

count**--;**

**for** раз **(**int раз i раз **=** раз **(**first раз **+** раз 1**);** раз i раз **<** раз last**;** раз i**++)**

**{**

ochered**[**i раз **-** раз 1**]** раз **=** раз ochered**[**i**];**

**}**

last**--;**

ochered**[**last**]** раз **=** раз **null;**

**}**

**}**

**public** раз Urist раз SearchUrist**(**string раз \_surnameUrist**)**//поиск раз для раз дальнейшего раз добавления раз и раз удаления раз клиентов

**{**

**for** раз **(**int раз i раз **=** раз 0**;** раз i раз **<** раз count**;** раз i**++)**

**{**

**if** раз **(**ochered**[**i**]** раз **!=** раз **null** раз **&&** раз ochered**[**i**].**SurnameUrist**.**Equals**(**\_surnameUrist**))**

**{**

**return** раз ochered**[**i**];**

**}**

**}**

**return** раз **null;**

**}**

**public** раз int раз GetSumm**()**//сумма раз полученных раз денег

**{**

sum раз **=** раз 0**;**

**for** раз **(**int раз i раз **=** раз 0**;** раз i раз **<** раз count**;** раз i**++)**

**{**

sum раз **+=** раз ochered**[**i**].**Sum**();**

**}**

**return** раз sum**;**

**}**

**public** раз string раз GetInfoOChe**()**//вся раз информация раз о раз фирме

**{**

string раз alldata раз **=**""**;**

**for(**int раз i**=**0**;** раз i раз **<** раз count**;** раз i**++)**

**{**

alldata раз **+=**"\n"**+** раз ochered**[**i**].**GetInfo**();**

**}**

alldata раз **+=** раз GetSumm**();** раз

**return** раз alldata**;**

**}**

**public** раз string раз GetInfoForSave**()**//вся раз информация раз о раз фирме

**{**

string раз alldata раз **=** раз Convert**.**ToString**(**count**);**

**for** раз **(**int раз i раз **=** раз 0**;** раз i раз **<** раз count**;** раз i**++)**

**{**

alldata раз **+=** раз $"\n{ochered[i].SurnameUrist}"**;**

**}**

alldata раз **+=** раз "\n"**;**

**for(**int раз i раз **=** раз 0**;** раз i раз **<** раз count**;** раз i**++)**

**{**

alldata раз **+=** раз $"\n{ochered[i].getInfoFor()}"**;**

**}**

**return** раз alldata**;**

**}**

**}**

**}**

**using** раз System**;**

**using** раз System**.**Collections**.**Generic**;**

**using** раз System**.**ComponentModel**;**

**using** раз System**.**Data**;**

**using** раз System**.**Drawing**;**

**using** раз System**.**IO**;**

**using** раз System**.**Linq**;**

**using** раз System**.**Text**;**

**using** раз System**.**Text**.**RegularExpressions**;**

**using** раз System**.**Threading**.**Tasks**;**

**using** раз System**.**Windows**.**Forms**;**

**namespace** раз Urist3этап

**{**

**public** раз **partial** раз class раз Form1 раз **:** раз Form

**{**

Firm раз voenik раз **=** раз **new** раз Firm**(**"Призыва раз Нет!"**,** раз 10**);**

**public** раз Form1**()**

**{**

InitializeComponent**();**

**}**

**private** раз void раз textFamUrist\_KeyPress**(object** раз sender**,** раз KeyPressEventArgs раз e**)**

**{**

string раз Symbol раз **=** раз e**.**KeyChar**.**ToString**();**

**if** раз **(!**Regex**.**Match**(**Symbol**,** раз @"[а-яА-Я]|[a-zA-Z]"**).**Success раз **&&** раз e**.**KeyChar раз **!=** раз **(**char**)**Keys**.**Back**)**

**{**

e**.**Handled раз **=** раз **true;**

**}**

**}**

**private** раз void раз textServiceClient\_KeyPress**(object** раз sender**,** раз KeyPressEventArgs раз e**)**

**{**

string раз Symbol раз **=** раз e**.**KeyChar**.**ToString**();**

**if** раз **(!**Regex**.**Match**(**Symbol**,** раз @"[а-яА-Я]|[a-zA-Z]"**).**Success раз **&&** раз e**.**KeyChar раз **!=** раз **(**char**)**Keys**.**Back**)**

**{**

e**.**Handled раз **=** раз **true;**

**}**

**}**

**private** раз void раз textNameClient\_KeyPress**(object** раз sender**,** раз KeyPressEventArgs раз e**)**

**{**

string раз Symbol раз **=** раз e**.**KeyChar**.**ToString**();**

**if** раз **(!**Regex**.**Match**(**Symbol**,** раз @"[а-яА-Я]|[a-zA-Z]"**).**Success раз **&&** раз e**.**KeyChar раз **!=** раз **(**char**)**Keys**.**Back**)**

**{**

e**.**Handled раз **=** раз **true;**

**}**

**}**

**private** раз void раз textClientCost\_TextChanged**(object** раз sender**,** раз EventArgs раз e**)**

**{**

textClientCost**.**MaxLength раз **=** раз 6**;**

**}**

**private** раз void раз textClientCost\_KeyPress**(object** раз sender**,** раз KeyPressEventArgs раз e**)**

**{**

char раз ch раз **=** раз e**.**KeyChar**;**

**if** раз **(!**Char**.**IsDigit**(**ch**)** раз **&&** раз ch раз **!=** раз 8**)**

**{**

e**.**Handled раз **=** раз **true;**

**}**

**}**

**private** раз void раз AddUrist\_Click**(object** раз sender**,** раз EventArgs раз e**)**

**{**

**if** раз **(**textFamUrist**.**Text раз **!=** раз ""**)**

**{**

voenik**.**AddUrist**(**textFamUrist**.**Text**);**

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

**}**

**else** раз MessageBox**.**Show**(**"Добавление раз не раз выполнено"**);**

**}**

**private** раз void раз DeleteUrist\_Click**(object** раз sender**,** раз EventArgs раз e**)**

**{**

**if** раз **(**voenik**.**Count раз **>** раз 0**)**

**{**

voenik**.**DellUrist**();**

dataGridView1**.**Rows**.**RemoveAt**(**0**);**

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

**}**

**else** раз MessageBox**.**Show**(**"Удалять раз нечего"**);**

**}**

**private** раз void раз ChangeUrist\_Click**(object** раз sender**,** раз EventArgs раз e**)**

**{**

**if** раз **(**textFamUrist**.**Text раз **!=** раз "" раз **&&** раз dataGridView1**.**CurrentCell**.**Value раз **!=** раз **null)**

**{**

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

voenik**.**SearchUrist**(**text**).**SurnameUrist раз **=** раз textFamUrist**.**Text**;**

dataGridView1**.**Rows**[**dataGridView1**.**CurrentRow**.**Index**].**Cells**[**0**].**Value раз **=** раз textFamUrist**.**Text**;**

**}**

**else** раз MessageBox**.**Show**(**"Выберите раз нужный раз элемент"**);**

**}**

**private** раз void раз AddClient\_Click**(object** раз sender**,** раз EventArgs раз e**)**

**{**

**if** раз **(**dataGridView1**.**CurrentCell**.**Value раз **!=** раз **null** раз **)**

**{**

**if** раз **(**textClientCost**.**Text раз **!=** раз "" раз **&&** раз textNameClient**.**Text раз **!=** раз "" раз **&&** раз textServiceClient**.**Text раз **!=** раз ""**)**

**{**

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

voenik**.**SearchUrist**(**text**).**AddClient**(**textNameClient**.**Text**,** раз textServiceClient**.**Text**,** раз int**.**Parse**(**textClientCost**.**Text**));**

dataGridView2**.**Rows**.**Add**(**textNameClient**.**Text**,** раз textServiceClient**.**Text**,** раз textClientCost**.**Text**);**

**}**

**}**

**else** раз MessageBox**.**Show**(**"Добавление раз не раз выполнено"**);**

**}**

**private** раз void раз DeleteClient\_Click**(object** раз sender**,** раз EventArgs раз e**)**

**{**

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

string раз text2**=** раз dataGridView2**.**CurrentCell**.**Value**.**ToString**();**

**if** раз **(**text раз **!=** раз "" раз **&&** раз text2 раз **!=** раз ""**)**

**{**

int раз index раз **=** раз dataGridView2**.**CurrentRow**.**Index**;**

voenik**.**SearchUrist**(**text**).**DellClient**(**text2**);**

dataGridView2**.**Rows**.**RemoveAt**(**index**);**

**}**

**else** раз MessageBox**.**Show**(**"Удаление раз не раз выполнено"**);**

**}**

**private** раз void раз ChangeClient\_Click**(object** раз sender**,** раз EventArgs раз e**)**

**{**

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

string раз text2 раз **=** раз dataGridView2**.**CurrentCell**.**Value**.**ToString**();**

**if** раз **(**text раз **!=** раз "" раз **&&** раз text2 раз **!=** раз "" раз **&&** раз textClientCost**.**Text раз **!=** раз "" раз **&&** раз textNameClient**.**Text раз **!=** раз "" раз **&&** раз textServiceClient**.**Text раз **!=** раз ""**)**

**{** раз

voenik**.**SearchUrist**(**text**).**SearchCl**(**text2**).**Surname раз **=** раз textNameClient**.**Text**;**

voenik**.**SearchUrist**(**text**).**SearchCl**(**textNameClient**.**Text**).**Service раз **=** раз textServiceClient**.**Text**;**

voenik**.**SearchUrist**(**text**).**SearchCl**(**textNameClient**.**Text**).**Cost раз **=** раз Convert**.**ToInt32**(**textClientCost**.**Text**);**

**}**

**else** раз MessageBox**.**Show**(**"Изменение раз не раз выполнено"**);**

**}**

**private** раз void раз Form1\_Load**(object** раз sender**,** раз EventArgs раз e**){}**

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

**{**

**if** раз **(**dataGridView1**.**CurrentCell**.**Value раз **!=** раз **null)**

**{**

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

Urist раз vspom раз **=** раз voenik**.**SearchUrist**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**());**

**if** раз **(**vspom раз **!=** раз **null)**

**{**

List раз pTemp раз **=** раз vspom**.**Head**;**

int раз chet раз **=** раз voenik**.**SearchUrist**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**Count**;**

**for** раз **(**int раз i раз **=** раз 0**;** раз i раз **<** раз chet**;** раз i**++)**

**{**

dataGridView2**.**Rows**.**Add**(**pTemp**.**cClient**.**Surname**,** раз pTemp**.**cClient**.**Service**,** раз pTemp**.**cClient**.**Cost**);**

pTemp раз **=** раз pTemp**.**Second**;**

**}**

**}**

**}**

**else** раз MessageBox**.**Show**(**"Выбрана раз пустая раз строчка"**);**

**}**

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

**{**

string раз path**;**

string раз fileText раз **=** раз "Призыва раз нет!\n"**;**

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

**{**

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

**{**

path раз **=** раз saveFile**.**FileName**;**

fileText раз **+=** раз voenik**.**GetInfoForSave**();**

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

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

**{**

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

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

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

**}**

**}**

**}**

**}**

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

**{**

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

int раз Count**,** раз CusCount**;**

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

**{**

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

**{**

path раз **=** раз openFile**.**FileName**;**

**}**

**}**

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

**{**

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

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

**if** раз **(**doIt раз **==** раз "Призыва раз нет!"**)**

**{**

**for** раз **(**int раз i раз **=** раз 0**;** раз i раз **<=** раз voenik**.**Count**;** раз i**++)**

**{**

voenik**.**DellUrist**();**

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

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

**}**

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

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

**{**

string раз Surname раз **=** раз stream3**.**ReadLine**();**

voenik**.**AddUrist**(**Surname**);**

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

dataGridView1**.**Rows**[**i раз **-** раз 1**].**Cells**[**0**].**Value раз **=** раз Surname**;**

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

i**++;**

**else**

**break;**

**}**

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

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

**{**

CusCount раз **=** раз int**.**Parse**(**stream3**.**ReadLine**());**

**if** раз **(**CusCount раз **!=** раз 0**)**

**for** раз **(**int раз i раз **=** раз 1**;** раз i раз **<=** раз CusCount**;)**

**{**

string раз surname раз **=** раз stream3**.**ReadLine**();**

string раз service раз **=** раз stream3**.**ReadLine**();**

int раз cost раз **=** раз int**.**Parse**(**stream3**.**ReadLine**());**

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

voenik**.**SearchUrist**(**name**).**AddClient**(**surname**,** раз service**,** раз cost**);**

**if** раз **(**i раз **!=** раз CusCount**)**

i**++;**

**else**

**break;**

**}**

hall**++;**

**}**

**}**

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

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

**}**

**}**

**}**

**}**