УВО ком «Университет ком Управления ком «ТИСБИ»

Факультет ком информационных ком технологий

Кафедра ком информационных ком технологий

**Курсовая ком работа ком**

по ком дисциплине ком «Объектно-ориентированное ком программирование»

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

Выполнил: ком студент ком гр. ком П-011 ком Сучёв ком Н.Е. ком

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

Казань ком 2022

Оглавление

[Постановка ком задачи 3](#_Toc104845698)

[1. Описание ком используемых ком структур ком данных ком с ком алгоритмами ком выполнения ком основных ком операций 5](#_Toc104845699)

[1.1 ком Адресный ком замкнутый ком (кольцевой) ком упорядоченный ком однонаправленный ком список ком без ком заголовка. 5](#_Toc104845700)

[1.2 ком Кольцевая ком очередь ком на ком основе ком обычного ком массива 8](#_Toc104845701)

[2. Краткие ком сведения ком об ком объектном ком подходе 10](#_Toc104845702)

[3. Формализованное ком описание ком разработанных ком классов 16](#_Toc104845703)

[1. Описание ком демонстрационного ком модуля ком с ком характеристикой ком использованных ком стандартных ком компонентов ком и ком списком ком реализованных ком обработчиков 21](#_Toc104845704)

[2. Описание ком структуры ком проекта ком в ком соответствии ком с ком использованным ком инструментом ком разработки 36](#_Toc104845705)

[Листинг ком программы 40](#_Toc104845706)

Постановка ком задачи

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

**Постановка ком задачи: ком** разработать ком объектную ком программу ком для ком хранения ком данных ком об ком услугах, ком оказанных ком некоторой ком юридической ком фирмой. ком Учёт ком организуется ком за ком счёт ком ведения ком списка ком юристов ком фирмы ком с ком указанием ком фамилии ком и ком номера ком мобильного ком телефона. ком Для ком каждого ком юриста ком создать ком отдельный ком список ком его ком клиентов ком с ком указанием ком фамилии ком клиента, ком вида ком услуги ком и ком её ком стоимости.

Разработка ком программы ком включает ком в ком себя:

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

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

Разработка ком выполняется ком с ком учетом ком следующих ком требований:

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

1.1 ком Очередь ком на ком основе ком массива.

Очередь ком – ком структура ком данных ком типа ком «список», ком позволяющая ком добавлять ком элементы ком лишь ком в ком конец ком списка, ком и ком извлекать ком их ком из ком его ком начала. ком Она ком функционирует ком по ком принципу ком FIFO ком (First ком In, ком First ком Out ком — ком «первым ком пришёл ком — ком первым ком вышел»), ком для ком которого ком характерно, ком что ком все ком элементы ком a1, ком a2, ком …, ком an-1, ком an, ком добавленные ком раньше ком элемента ком an+1, ком должны ком быть ком удалены ком прежде, ком чем ком будет ком удален ком элемент ком an+1. ком Также ком очередь ком может ком быть ком определена ком как ком частный ком случай ком односвязного ком списка, ком который ком обслуживает ком элементы ком в ком порядке ком их ком поступления. ком Как ком и ком в ком «живой» ком очереди, ком здесь ком первым ком будет ком обслужен ком тот, ком кто ком пришел ком первым.



Рис. ком 1 ком Очередь.

Стандартный ком набор ком операций ком (часто ком у ком разных ком авторов ком он ком не ком идентичен), ком выполняемых ком над ком очередями, ком совпадает ком с ком тем, ком что ком используется ком при ком обработке ком стеков:

* добавление ком элемента;
* удаление ком элемента;
* чтение ком первого ком элемента.

Только, ком если ком в ком отношении ком стека ком в ком момент ком добавления ком или ком удаления ком элемента ком допустимо ком задействование ком лишь ком его ком вершины, ком то ком касательно ком очереди ком эти ком две ком операции ком должны ком быть ком применены ком так, ком как ком это ком регламентировано ком в ком определении ком этой ком структуры ком данных, ком т. ком е. ком добавление ком – ком в ком конец, ком удаление ком – ком из ком начала. ком Далее, ком при ком реализации ком интерфейса ком очереди, ком список ком стандартных ком операций ком будет ком расширен.

Выделяют ком два ком способа ком программной ком реализации ком очереди. ком Первый ком из ком них ком основан ком на ком базе ком массива, ком а ком второй ком на ком базе ком указателей ком (связного ком списка). ком Первый ком способ ком – ком статический, ком т. ком к. ком очередь ком представляется ком в ком виде ком простого ком статического ком массива, ком второй ком – ком динамический.



Рис. ком ком 2 ком Добавление ком

При ком добавление ком нового ком элемента ком данные ком добавляются ком в ком конец ком очереди, ком т.е. ком первую ком незанятую ком ячейку ком массива. ком



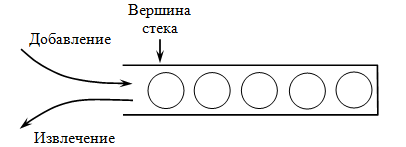
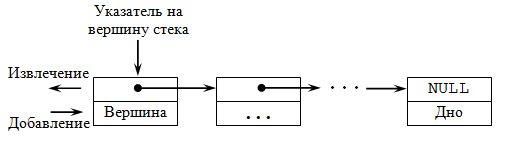
Рис ком 3. ком Удаление ком элемента.

При ком удалении ком из ком памяти ком стирается ком первый ком элемента ком массива. ком Соседний ком элемент ком и ком все ком последующие ком смешаются ком на ком одну ком ячейку ком к ком началу ком массива. ком

1.2 ком Адресный ком стек.

Стек ком (англ. ком stack ком – ком стопка) ком – ком это ком структура ком данных, ком в ком которой ком новый ком элемент ком всегда ком записывается ком в ком ее ком начало ком (вершину) ком и ком очередной ком читаемый ком элемент ком также ком всегда ком выбирается ком из ком ее ком начала. ком В ком стеках ком используется ком метод ком доступа ком к ком элементам ком LIFO ком ( ком Last ком Input ком – ком First ком Output, ком "последним ком пришел ком – ком первым ком вышел"). ком Чаще ком всего ком принцип ком работы ком стека ком сравнивают ком со ком стопкой ком тарелок: ком чтобы ком взять ком вторую ком сверху, ком нужно ком сначала ком взять ком верхнюю.

Стек ком – ком это ком список, ком у ком которого ком доступен ком один ком элемент ком (одна ком позиция). ком Этот ком элемент ком называется ком вершиной ком стека. ком Взять ком элемент ком можно ком только ком из ком вершины ком стека, ком добавить ком элемент ком можно ком только ком в ком вершину ком стека. ком Например, ком если ком записаны ком в ком стек ком числа ком 1, ком 2, ком 3, ком то ком при ком последующем ком извлечении ком получим ком 3,2,1.

* 
* Рис. ком 4 ком Принцип ком стека.
* 
* Рис. ком 5 ком Структура ком адресного ком стека. ком

Стек ком как ком динамическую ком структуру ком данных ком легко ком организовать ком на ком основе ком линейного ком списка. ком Поскольку ком работа ком всегда ком идет ком с ком заголовком ком стека, ком то ком есть ком не ком требуется ком осуществлять ком просмотр ком элементов, ком удаление ком и ком вставку ком элементов ком в ком середину ком или ком конец ком списка, ком то ком достаточно ком использовать ком экономичный ком по ком памяти ком линейный ком однонаправленный ком список. ком Для ком такого ком списка ком достаточно ком хранить ком указатель ком вершины ком стека, ком который ком указывает ком на ком первый ком элемент ком списка. ком Если ком стек ком пуст, ком то ком списка ком не ком существует, ком и ком указатель ком принимает ком значение ком NULL. ком

Возможны ком три ком операции ком со ком стеком: ком добавление ком элемента ком (иначе ком проталкивание, ком push), ком удаление ком элемента ком (pop) ком и ком чтение ком головного ком элемента ком (peek).

При ком проталкивании ком (push) ком добавляется ком новый ком элемент, ком указывающий ком на ком элемент, ком бывший ком до ком этого ком головой. ком Новый ком элемент ком теперь ком становится ком головным.

При ком удалении ком элемента ком (pop) ком убирается ком первый, ком а ком головным ком становится ком тот, ком на ком который ком был ком указатель ком у ком этого ком объекта ком (следующий ком элемент). ком При ком этом ком значение ком убранного ком элемента ком возвращается.

1. Краткие ком сведения ком об ком объектном ком подходе

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

Основные ком принципы ком ООП:

Инкапсуляция

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

Правильная ком инкапсуляция ком важна ком по ком многим ком причинам: ком

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

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

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

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

Всего ком модификаторов ком доступа ком четыре:

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

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

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

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

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

Пример ком корректной ком инкапсуляции ком класса:

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

Наследование

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

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

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

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

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

Полиморфизм

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

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

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

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

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

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

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

Абстракция

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

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

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

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

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

3. Формализованное ком описание ком разработанных ком классов

В ком иерархии ком наследования ком класс ком Customers ком находиться ком на ком самой ком вершине ком его ком наследует ком класс ком Lawyer ком содержащий ком сведенья ком о ком посетителях ком адвоката. ком

// ком Сustomer ком

**using** ком System**;**

**using** ком System**.**Collections**.**Generic**;**

**using** ком System**.**Text**;**

**namespace** ком Third

**{**

class ком Customers

**{**

**private** ком string ком surname**;**//фамилия ком заказчика

**private** ком string ком service**;**//оказанная ком услуга

**private** ком int ком cost**;**//стоимость ком услуги

**public** ком Customers**(**string ком \_surname**,** ком string ком \_service**,** ком int ком \_cost**)**//конструктор

**{**

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

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

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

**}**

**public** ком string ком Surname ком //get/set ком для ком фамилии

**{**

**get** ком **{** ком **return** ком surname**;** ком **}**

**set** ком **{** ком **this.**surname ком **=** ком **value;** ком **}**

**}**

**public** ком string ком Service//get/set ком для ком оказанной ком услуги

**{**

**get** ком **{** ком **return** ком service**;** ком **}**

**set** ком **{** ком **this.**service ком **=** ком **value;** ком **}**

**}**

**public** ком int ком Cost//get/set ком для ком стоимости

**{**

**get** ком **{** ком **return** ком cost**;** ком **}**

**set** ком **{** ком **this.**cost ком **=** ком **value;** ком **}**

**}**

**}**

**}**

Класс ком Lower ком хранит ком в ком себе ком сведенья ком о ком юристе ком (Фамилию). ком Данные ком о ком посетителях ком в ком виде ком очереди ком на ком массиве, ком а ком также ком заработанную ком сумму ком от ком всех ком посетителей. ком

**// ком Lawyer**

**using** ком System**;**

**using** ком System**.**Collections**.**Generic**;**

**using** ком System**.**Text**;**

**namespace** ком Third

**{**

class ком Lawyer

**{**

**private** ком string ком lSurname**;**//фамилия ком юриста

**private** ком long ком number**;**

**private** ком Customers**[]** ком queue**;**//массив-очередь

**private** ком int ком first**;**//ссылка ком на ком 1 ком элемент

**private** ком int ком last**;**//ссылка ком на ком последний ком элемент

**private** ком int ком sumCost**,** ком size**,**count**;**//сумма, ком потраченная ком на ком услуги

**public** ком Lawyer**(**string ком \_lSurname**,** ком long ком \_number**,** ком int ком \_size**)** ком //конструктор

**{**

**this.**lSurname ком **=** ком \_lSurname**;**

**this.**number ком **=** ком \_number**;**

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

queue ком **=** ком **new** ком Customers**[**size**];**

first ком **=** ком 0**;** ком last ком **=** ком 0**;** ком count ком **=** ком 0**;**

**}**

**public** ком Customers ком Search**(**string ком \_lSurname**)**

**{**

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

**{**

**if(**queue**[**i**].**Surname**.**Equals**(**\_lSurname**))**

**{**

**return** ком queue**[**i**];**

**}**

**}**

**return** ком **null;**

**}**

**public** ком long ком Number ком **{** ком **get** ком **{** ком **return** ком number**;** ком **}** ком **set** ком **{** ком **this.**number ком **=** ком **value;** ком **}** ком **}**

**public** ком Customers**[]** ком Queue ком **{** ком **get** ком **{** ком **return** ком queue**;** ком **}** ком **}**

**public** ком string ком Surname**{** ком **get** ком **{** ком **return** ком lSurname**;** ком **}set** ком **{** ком **this.**lSurname ком **=** ком **value;** ком **}** ком **}**

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

**public** ком void ком addCust**(**string ком \_surname**,** ком string ком \_service**,** ком int ком \_cost**)** ком //добавление ком заказчика

**{**

**if** ком **(**size ком **>=** ком count**)**

**{**

**if** ком **(**first ком **>** ком 0 ком **&&** ком last ком **>=** ком size**)** ком last ком **=** ком 0**;**

Customers ком dobav ком **=** ком **new** ком Customers**(**\_surname**,** ком \_service**,** ком \_cost**);**

queue**[**last**]** ком **=** ком dobav**;**

last**++;**

count**++;**

**}**

**}**

**public** ком void ком delCust**()** ком //удаление ком заказчика ком по ком переменной ком first

**{** ком

**if** ком **(**count ком **!=** ком 0**)**

**{**

**if** ком **(**first ком **>** ком size**)** ком first ком **=** ком 0**;**

queue**[**first**]** ком **=** ком **null;**

count**--;**

first**++;**

**}**

**}**

**public** ком int ком SumCost**()**//возвращает ком сумму ком потраченных ком денег

**{**

sumCost ком **=** ком 0**;**

**for(** ком int ком i ком **=** ком 0**;** ком i ком **<** ком size**;** ком i**++)**

**{**

**if** ком **(**queue**[**i**]** ком **!=** ком **null)**

**{**

sumCost ком **+=** ком queue**[**i**].**Cost**;**

**}**

**}**

**return** ком sumCost**;**

**}**

**public** ком string ком getInfo**()** ком //получение ком информации ком об ком одном ком юристе ком и ком его ком заказчиках

**{**

string ком getinfo ком **=** ком "\n<=======Lawyer: ком " ком **+** ком lSurname ком **+** ком "=======>"**;**

Customers ком help**;**

**for** ком **(**int ком i ком **=** ком 0**;** ком i ком **<** ком size**;** ком i**++)**

**{**

help ком **=** ком queue**[**i**];**

**if** ком **(**help ком **!=** ком **null)**

**{**

getinfo ком **+=** ком "\n\nCustomer: ком " ком **+** ком help**.**Surname**;**

getinfo ком **+=** ком "\nService: ком " ком **+** ком help**.**Service**;**

getinfo ком **+=** ком "\nCost: ком " ком **+** ком help**.**Cost ком **+** ком "\n"**;**

**}**

**}**

getinfo ком **+=** ком "\nsum ком of ком cost: ком " ком **+** ком SumCost**();** ком

**return** ком getinfo**;**

**}**

**public** ком string ком getInfoFor**()**

**{**

string ком getinfo ком **=** ком Convert**.**ToString**(**Count**);**

Customers ком help**;**

**for** ком **(**int ком i ком **=** ком 0**;** ком i ком **<** ком size**;** ком i**++)**

**{**

help ком **=** ком queue**[**i**];**

**if** ком **(**help ком **!=** ком **null)**

**{**

getinfo ком **+=** ком "\n" ком **+** ком help**.**Surname**;**

getinfo ком **+=** ком "\n" ком **+** ком help**.**Service**;**

getinfo ком **+=** ком "\n" ком **+** ком help**.**Cost**;**

**}**

**}**

**return** ком getinfo**;**

**}**

**}**

**}**

Класс ком Company ком содержит ком в ком себе ком экземпляры ком с ком юристами, ком хранящимися ком в ком контейнере ком – ком стек. ком Company ком является ком самым ком низшим ком классом ком в ком наследовании ком и ком позволяет ком обращаться ком к ком методам ком класса ком Lower ком и ком Customers. ком

// ком Company

**using** ком System**;**

**using** ком System**.**Collections**.**Generic**;**

**using** ком System**.**Text**;**

**namespace** ком Third

**{**

class ком Company

**{**

**private** ком string ком nameComp**;**//название ком юр.фирмы

**private** ком Stack ком firstElenem**,**SP**;**//первый ком элемент ком в ком стеке

**private** ком int ком countElem**;**//количество ком элементов

**public** ком Company**(**string ком \_nameComp**)** ком //конструктор

**{**

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

SP ком **=** ком **null;**

firstElenem ком **=** ком **null;**

countElem ком **=** ком 0**;**

**}**

**public** ком string ком Info**()**

**{**

string ком info ком **=** ком Convert**.**ToString**(**Count**);**

Stack ком vspom ком **=** ком SP**;**

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

**{**

info ком **+=** ком "\n" ком **+** ком vspom**.**date**.**Surname**;**

info ком **+=** ком "\n" ком **+** ком vspom**.**date**.**Number**;**

vspom ком **=** ком vspom**.**getLaw**();**

**}**

**return** ком info**;**

**}**

**public** ком int ком Count ком **{** ком **get** ком **{** ком **return** ком countElem**;** ком **}** ком **}**

**public** ком void ком addLaw**(**string ком \_lSurname**,** ком long ком \_number**,** ком int ком \_size**)** ком //добавление ком юриста

**{**

Lawyer ком dobLaw ком **=** ком **new** ком Lawyer**(**\_lSurname**,** ком \_number**,** ком \_size**);**

Stack ком help ком **=** ком **new** ком Stack**(**dobLaw**,** ком SP**);**

**if** ком **(**SP ком **==** ком **null)** ком firstElenem ком **=** ком help**;**

SP ком **=** ком help**;**

countElem**++;**

**}**

**public** ком void ком delLaw**()** ком //удаление ком юриста

**{**

**if(**SP ком **!=** ком **null)**

**{**

SP ком **=** ком SP**.**getLaw**();**

countElem**--;**

**}**

**}**

**public** ком Lawyer ком searchLaw**(**string ком \_lSurname**)** ком //поиск ком юриста ком по ком фамилии

**{**

Stack ком vspom ком **=** ком SP**;**

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

**{**

**if** ком **(**vspom**.**date**.**Surname**.**Equals**(**\_lSurname**))**

**{**

**return** ком vspom**.**date**;**

**}**

**else** ком vspom ком **=** ком vspom**.**getLaw**();**

**}**

**return** ком **null;**

**}**

**public** ком int ком SumCostLaw**()**//заработок ком всей ком компании

**{**

int ком summ ком **=** ком 0**;**

Stack ком vspom ком **=** ком SP**;**

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

**{**

summ ком **+=** ком vspom**.**date**.**SumCost**();**

vspom ком **=** ком vspom**.**getLaw**();**

**}**

**return** ком summ**;**

**}**

**public** ком string ком getInfoComp**()** ком //получение ком всей ком информации ком о ком компании

**{**

string ком getInfoAll ком **=** ком "\n\nName ком Company: ком " ком **+** ком nameComp**;**

Stack ком vspom ком **=** ком SP**;**

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

**{**

getInfoAll ком **+=** ком "\n" ком **+** ком vspom**.**date**.**getInfo**()** ком **+** ком "\n"**;**

vspom ком **=** ком vspom**.**getLaw**();**

**}**

getInfoAll ком **+=** ком "\n[summ ком Cost ком all ком company: ком " ком **+** ком SumCostLaw**()** ком **+** ком "]"**;**

**return** ком getInfoAll**;**

**}**

**public** ком string ком getInfoFor**()** ком //получение ком всей ком информации ком о ком компании

**{**

string ком getInfoAll ком **=** ком ""**;**

Stack ком vspom ком **=** ком SP**;**

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

**{**

getInfoAll ком **+=** ком "\n" ком **+** ком vspom**.**date**.**getInfoFor**();**

vspom ком **=** ком vspom**.**getLaw**();**

**}**

**return** ком getInfoAll**;**

**}**

**}**

**}**

В ком классе ком Company ком данные ком о ком юристах ком храниться ком в ком стеке. ком В ком программе ком реализована ком структура ком стек ком на ком основе ком адресных ком переменных. ком

// ком Stack

**using** ком System**;**

**using** ком System**.**Collections**.**Generic**;**

**using** ком System**.**Text**;**

**namespace** ком Third

**{**

class ком Stack

**{**

**public** ком Lawyer ком date**;**//Данные

**private** ком Stack ком pred**;**//ссылка ком на ком предка

**public** ком Stack**(**Lawyer ком \_date**,** ком Stack ком \_SP**)**//конструктор

**{**

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

pred ком **=** ком \_SP**;**

**}**

**public** ком Stack ком getLaw**()**//получение ком ссылки ком на ком предка

**{**

**return** ком pred**;**

**}**

**}**

**}**

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

При ком запуске ком программы ком создаётся ком экземпляр ком класса ком Company ком с ком названием ком компании ком необходимое ком для ком создания ком файлов ком сохранения ком данных ком в ком файл. ком

// ком Form1 ком

**using** ком System**;**

**using** ком System**.**Collections**.**Generic**;**

**using** ком System**.**ComponentModel**;**

**using** ком System**.**Data**;**

**using** ком System**.**Drawing**;**

**using** ком System**.**IO**;**

**using** ком System**.**Linq**;**

**using** ком System**.**Text**;**

**using** ком System**.**Threading**.**Tasks**;**

**using** ком System**.**Windows**.**Forms**;**

**namespace** ком Third

**{**

**public** ком **partial** ком class ком Form1 ком **:** ком Form

**{**

Company ком Akyi ком **=** ком **new** ком Company**(**"Company ком Akyi"**);**

**public** ком Form1**()**

**{**

InitializeComponent**();**

**}**

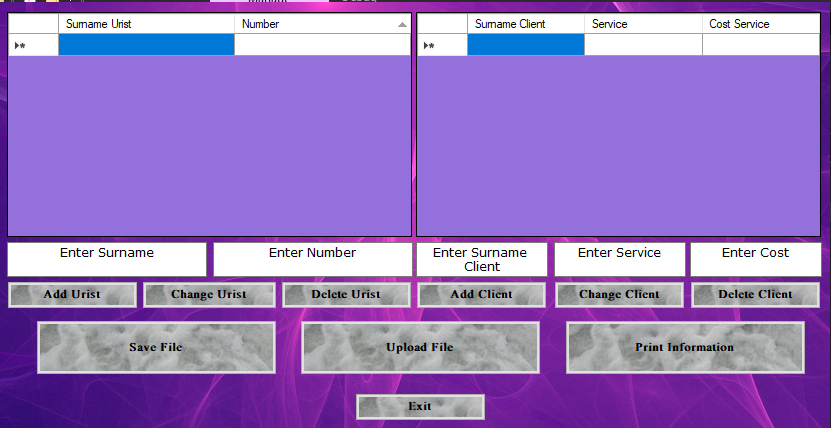


Рис. ком 3 ком ком Запуск ком программы.

Прописываем ком события ком для ком существующих ком TextBox, ком что ком при ком начале ком ввода, ком старая ком информация ком стирается.

**private** ком void ком textSurUrist\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

textSurUrist**.**Clear**();**

**}**

**private** ком void ком textCost\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

textCost**.**Clear**();**

**}**

**private** ком void ком testNumber\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

textNumber**.**Clear**();**

**}**

**private** ком void ком textSurClient\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

textSurClient**.**Clear**();**

**}**

**private** ком void ком textService\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

textService**.**Clear**();**

**}**

Данный ком метод ком отрабатывает ком событие, ком когда ком пользователь ком выбирает ком клиента ком в ком первой ком таблице ком клиентов. ком Метод ком необходим ком для ком дальнейшего ком добавления ком посещений ком конкретному ком пациенту. ком Если ком пользователь ком не ком выбрал ком строчку, ком то ком программа ком не ком будет ком работать. ком А ком также ком у ком выбранного ком юриста ком показывает ком сведенья ком о ком посетителях ком заполняя ком вторую ком таблицу. ком

**private** ком void ком dataGridView2\_CellContentClick**(object** ком sender**,** ком DataGridViewCellEventArgs ком e**)**

**{**

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

string ком text ком **=** ком dataGridView2**.**CurrentCell**.**Value**.**ToString**();**

Lawyer ком vspom ком **=** ком Akyi**.**searchLaw**(**text**);**

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

**{**

Customers**[]** ком copy1 ком **=** ком vspom**.**Queue**;**

int ком chet ком **=** ком Akyi**.**searchLaw**(**text**).**Count**;**

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

**{**

dataGridView1**.**Rows**.**Add**(**copy1**[**i**].**Surname**,** ком copy1**[**i**].**Service**,**copy1**[**i**].**Cost**);**

**}**

**}**

**}**

**private** ком void ком textSurUrist\_TextChanged**(object** ком sender**,** ком EventArgs ком e**)**

**{**

**}**

Метод ком кнопки ком добавления ком юриста. ком Добавляет ком данные ком фамилии ком в ком таблицу. ком Создаёт ком экземпляр ком в ком стеке ком класса ком Lower. ком

**private** ком void ком butAddUrist\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

Akyi**.**addLaw**(**textSurUrist**.**Text**,** ком Convert**.**ToInt64**(**textNumber**.**Text**),** ком 10**);**

dataGridView2**.**Rows**.**Add**(**textSurUrist**.**Text**,** ком textNumber**.**Text**);**

textSurUrist**.**Clear**();**textNumber**.**Clear**();**

**}**

Метод ком закрывает ком окно ком программы.

**private** ком void ком butExit\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

Close**();**

**}**

Метод ком обрабатывающий ком событие ком нажатия ком пользователем ком кнопки ком добавления ком посетителя ком юристу. ком Происходит ком валидация ком данных ком с ком TextBox, ком добавление ком в ком классе ком Lower ком экземпляра ком Customers ком и ком добавление ком данных ком в ком таблицу.

**private** ком void ком butAddCli\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

string ком info ком **=** ком dataGridView2**.**CurrentCell**.**Value**.**ToString**();**

Akyi**.**searchLaw**(**info**).**addCust**(**textSurClient**.**Text**,**textService**.**Text**,**Convert**.**ToInt32**(**textCost**.**Text**));**

dataGridView1**.**Rows**.**Add**(**textSurClient**.**Text**,** ком textService**.**Text**,** ком textCost**.**Text**);**

**}**

Валидатор ком ввода ком номера ком на ком максимальное ком значение ком из ком 10^11. ком А ком таже ком ограничение ком на ком ввод ком только ком цифр.

**private** ком void ком textNumber\_TextChanged**(object** ком sender**,** ком EventArgs ком e**)**

**{**

textNumber**.**MaxLength ком **=** ком 11**;**

**}**

**private** ком void ком textCost\_KeyPress**(object** ком sender**,** ком KeyPressEventArgs ком e**)**

**{**

// ком ввод ком в ком texBox ком только ком цифр ком и ком кнопки ком Backspace

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

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

**{**

e**.**Handled ком **=** ком **true;**

**}**

**}**

**private** ком void ком textNumber\_KeyPress**(object** ком sender**,** ком KeyPressEventArgs ком e**)**

**{**

// ком ввод ком в ком texBox ком только ком цифр ком и ком кнопки ком Backspace

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

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

**{**

e**.**Handled ком **=** ком **true;**

**}**

**}**

Валидатор ком на ком имя ком пользователя ком длиной ком не ком более ком 10 ком символов.

**private** ком void ком textCost\_TextChanged**(object** ком sender**,** ком EventArgs ком e**)**

**{**

textCost**.**MaxLength ком **=** ком 10**;**

**}**

Метод ком изменения ком данных ком о ком юристе. ком После ком валидации ком TextBox ком происходит ком обращение ком к ком методам ком Company ком для ком ввода ком новых ком значений ком и ком обновление ком таблицы ком юристов. ком

**private** ком void ком butChUrist\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

string ком info ком **=** ком dataGridView2**.**CurrentCell**.**Value**.**ToString**();**

Akyi**.**searchLaw**(**info**).**Surname ком **=** ком textSurUrist**.**Text**;**

Akyi**.**searchLaw**(**textSurUrist**.**Text**).**Number ком **=** ком Convert**.**ToInt64**(**textNumber**.**Text**);**

dataGridView2**.**Rows**[**dataGridView2**.**CurrentRow**.**Index**].**Cells**[**0**].**Value ком **=** ком textSurUrist**.**Text**;**

dataGridView2**.**Rows**[**dataGridView2**.**CurrentRow**.**Index**].**Cells**[**1**].**Value ком **=** ком textNumber**.**Text**;**

textSurClient**.**Clear**();**textNumber**.**Clear**();**

**}**

Метод ком удаления ком юриста ком из ком компании. ком Очищает ком таблицу ком и ком удаляет ком юриста ком из ком класса ком Company. ком

**private** ком void ком butDelUr\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

Akyi**.**delLaw**();**

dataGridView2**.**Rows**.**RemoveAt**(**Akyi**.**Count**);**

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

**}**

Конопка ком вывода ком данных ком о ком всей ком фирме. ком

**private** ком void ком butPrint\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

MessageBox**.**Show**(**Akyi**.**getInfoComp**());**

**}**

Метод ком обработки ком события ком изменения ком данных ком посетителя ком юриста. ком После ком валидации ком TextBox ком происходит ком обращение ком к ком методам ком Company ком для ком ввода ком новых ком значений ком посетителей ком и ком обновление ком таблицы ком посетителей. ком

**private** ком void ком butChCli\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

string ком infoUrist ком **=** ком dataGridView2**.**CurrentCell**.**Value**.**ToString**();**

string ком infoClient ком **=** ком dataGridView1**.**CurrentCell**.**Value**.**ToString**();**

Akyi**.**searchLaw**(**infoUrist**).**Search**(**infoClient**).**Surname ком **=** ком textSurClient**.**Text**;**

Akyi**.**searchLaw**(**infoUrist**).**Search**(**textSurClient**.**Text**).**Cost ком **=** ком Convert**.**ToInt32**(**textCost**.**Text**);**

Akyi**.**searchLaw**(**infoUrist**).**Search**(**textSurClient**.**Text**).**Service ком **=** ком textService**.**Text**;**

dataGridView1**.**Rows**[**dataGridView2**.**CurrentRow**.**Index**].**Cells**[**0**].**Value ком **=** ком textSurClient**.**Text**;**

dataGridView1**.**Rows**[**dataGridView2**.**CurrentRow**.**Index**].**Cells**[**1**].**Value ком **=** ком textService**.**Text**;**

dataGridView1**.**Rows**[**dataGridView2**.**CurrentRow**.**Index**].**Cells**[**2**].**Value ком **=** ком textCost**.**Text**;**

**}**

Метод ком удаления ком посетителя ком из ком класса ком и ком из ком таблицы ком удаляется ком экземпляр ком Customers. ком

**private** ком void ком butDelCli\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

string ком infoUr ком **=** ком dataGridView2**.**CurrentCell**.**Value**.**ToString**();**

Akyi**.**searchLaw**(**infoUr**).**delCust**();**

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

**}**

Метод ком выгрузки ком данных ком в ком файл ком формата ком txt. ком Для ком дальнейшего ком чтения ком в ком начале ком файла ком пишем ком "Akyi ком Company\n", ком чтобы ком можно ком было ком идентифицировать ком файл ком программы ком от ком других ком txt.

**private** ком void ком butSave\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

string ком path**;**

string ком fileText ком **=** ком "Akyi ком Company\n"**;**

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

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

**{**

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

**{**

path ком **=** ком saveFile**.**FileName**;**

fileText ком **+=** ком Akyi**.**Info**()** ком **+** ком "\n"**;**

fileText ком **+=** ком Akyi**.**getInfoFor**();**

//Та ком же ком работа ком с ком экземплярами ком файла ком и ком записи ком файла

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

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

**{**

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

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

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

**}**

**}**

**}**

**}**

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

**private** ком void ком butUpload\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

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

int ком Count**,** ком CusCount**,** ком gName**;**

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

**{**

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

**{**

path ком **=** ком openFile**.**FileName**;**

**}**

**}**

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

**{**

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

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

**if** ком **(**doIt ком **==** ком "Akyi ком Company"**)**

**{**

**for** ком **(**int ком i ком **=** ком 0**;** ком i ком **<=** ком Akyi**.**Count**;** ком i**++)**

**{**

Akyi**.**delLaw**();**

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

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

**}**

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

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

**{**

string ком SurnameUrist ком **=** ком stream3**.**ReadLine**();**

long ком Number ком **=** ком long**.**Parse**(**stream3**.**ReadLine**());**

Akyi**.**addLaw**(**SurnameUrist**,** ком Number**,** ком 10**);**

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

dataGridView2**.**Rows**[**i ком **-** ком 1**].**Cells**[**0**].**Value ком **=** ком SurnameUrist**;**

dataGridView2**.**Rows**[**i ком **-** ком 1**].**Cells**[**1**].**Value ком **=** ком Number**;**

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

i**++;**

**else**

**break;**

**}**

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

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

**{**

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

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

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

**{**

string ком SurnCli ком **=** ком stream3**.**ReadLine**();**

string ком Service ком **=** ком stream3**.**ReadLine**();**

gName ком **=** ком int**.**Parse**(**stream3**.**ReadLine**());**

string ком name ком **=** ком Convert**.**ToString**(**dataGridView2**.**Rows**[**hall**].**Cells**[**0**].**Value**);**

Akyi**.**searchLaw**(**name**).**addCust**(**SurnCli**,** ком Service**,** ком gName**);**

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

i**++;**

**else**

**break;**

**}**

hall**++;**

**}**

**}**

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

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

**}**

**}**

**}**

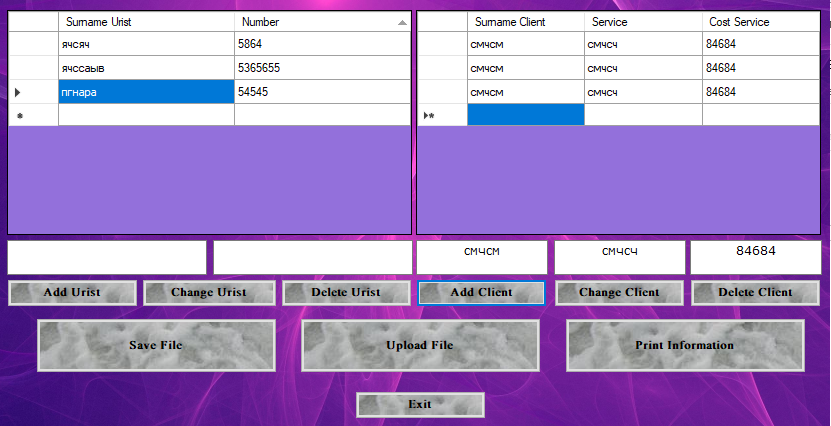


Рис. ком 4 ком Работа ком программы

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

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

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

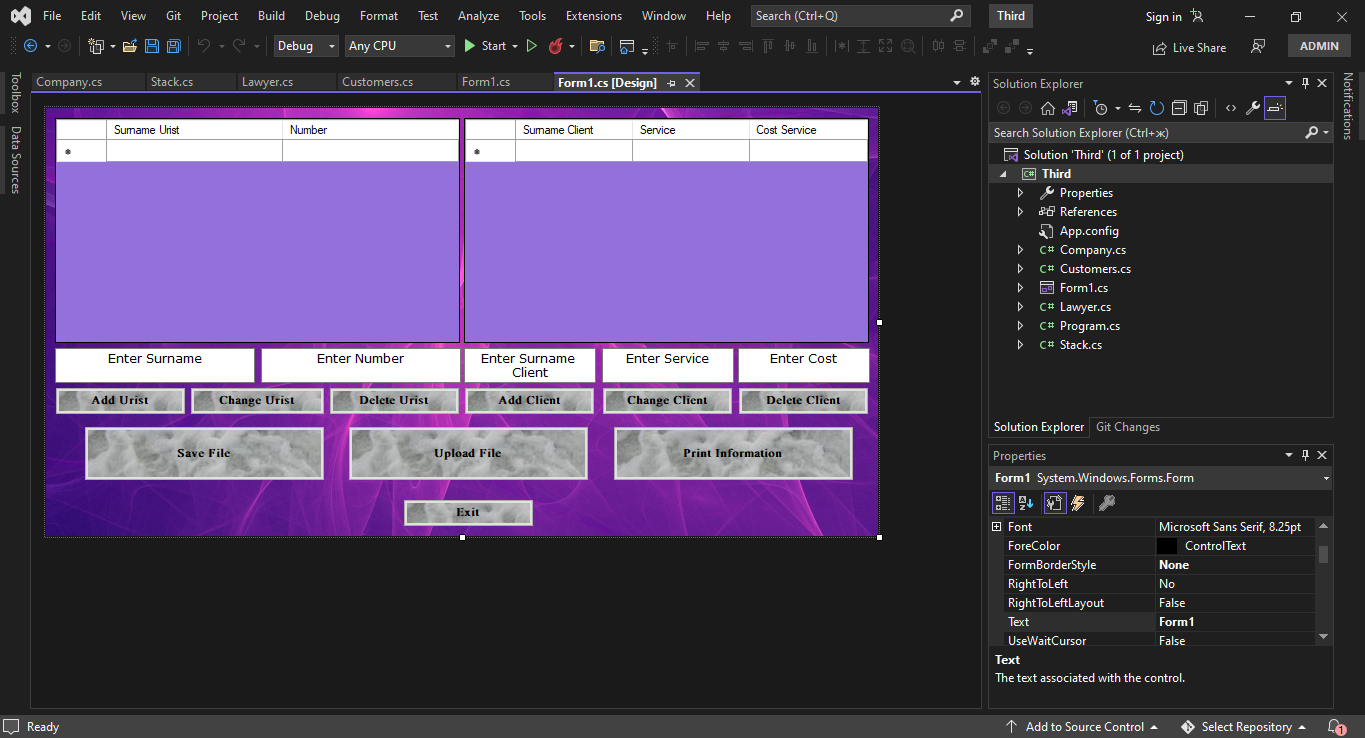


Рис. ком 5 ком Работа ком Visual ком Studio ком 2022

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

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

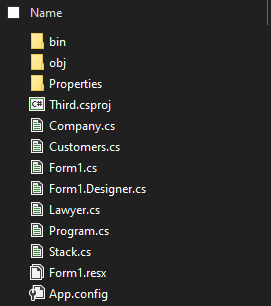


Рис. ком 6 ком Содержимое ком папки ком проекта. ком

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

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

**Список ком литературы**

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

Листинг ком программы

// ком Form1 ком

**using** ком System**;**

**using** ком System**.**Collections**.**Generic**;**

**using** ком System**.**ComponentModel**;**

**using** ком System**.**Data**;**

**using** ком System**.**Drawing**;**

**using** ком System**.**IO**;**

**using** ком System**.**Linq**;**

**using** ком System**.**Text**;**

**using** ком System**.**Threading**.**Tasks**;**

**using** ком System**.**Windows**.**Forms**;**

**namespace** ком Third

**{**

**public** ком **partial** ком class ком Form1 ком **:** ком Form

**{**

Company ком Akyi ком **=** ком **new** ком Company**(**"Company ком Akyi"**);**

**public** ком Form1**()**

**{**

InitializeComponent**();**

**}**

**private** ком void ком textSurUrist\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

textSurUrist**.**Clear**();**

**}**

**private** ком void ком textCost\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

textCost**.**Clear**();**

**}**

**private** ком void ком testNumber\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

textNumber**.**Clear**();**

**}**

**private** ком void ком textSurClient\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

textSurClient**.**Clear**();**

**}**

**private** ком void ком textService\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

textService**.**Clear**();**

**}**

**private** ком void ком dataGridView2\_CellContentClick**(object** ком sender**,** ком DataGridViewCellEventArgs ком e**)**

**{**

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

string ком text ком **=** ком dataGridView2**.**CurrentCell**.**Value**.**ToString**();**

Lawyer ком vspom ком **=** ком Akyi**.**searchLaw**(**text**);**

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

**{**

Customers**[]** ком copy1 ком **=** ком vspom**.**Queue**;**

int ком chet ком **=** ком Akyi**.**searchLaw**(**text**).**Count**;**

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

**{**

dataGridView1**.**Rows**.**Add**(**copy1**[**i**].**Surname**,** ком copy1**[**i**].**Service**,**copy1**[**i**].**Cost**);**

**}**

**}**

**}**

**private** ком void ком textSurUrist\_TextChanged**(object** ком sender**,** ком EventArgs ком e**)**

**{**

**}**

**private** ком void ком butAddUrist\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

Akyi**.**addLaw**(**textSurUrist**.**Text**,** ком Convert**.**ToInt64**(**textNumber**.**Text**),** ком 10**);**

dataGridView2**.**Rows**.**Add**(**textSurUrist**.**Text**,** ком textNumber**.**Text**);**

textSurUrist**.**Clear**();**textNumber**.**Clear**();**

**}**

**private** ком void ком butExit\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

Close**();**

**}**

**private** ком void ком butAddCli\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

string ком info ком **=** ком dataGridView2**.**CurrentCell**.**Value**.**ToString**();**

Akyi**.**searchLaw**(**info**).**addCust**(**textSurClient**.**Text**,**textService**.**Text**,**Convert**.**ToInt32**(**textCost**.**Text**));**

dataGridView1**.**Rows**.**Add**(**textSurClient**.**Text**,** ком textService**.**Text**,** ком textCost**.**Text**);**

**}**

**private** ком void ком textNumber\_TextChanged**(object** ком sender**,** ком EventArgs ком e**)**

**{**

textNumber**.**MaxLength ком **=** ком 11**;**

**}**

**private** ком void ком textCost\_KeyPress**(object** ком sender**,** ком KeyPressEventArgs ком e**)**

**{**

// ком ввод ком в ком texBox ком только ком цифр ком и ком кнопки ком Backspace

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

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

**{**

e**.**Handled ком **=** ком **true;**

**}**

**}**

**private** ком void ком textNumber\_KeyPress**(object** ком sender**,** ком KeyPressEventArgs ком e**)**

**{**

// ком ввод ком в ком texBox ком только ком цифр ком и ком кнопки ком Backspace

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

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

**{**

e**.**Handled ком **=** ком **true;**

**}**

**}**

**private** ком void ком textCost\_TextChanged**(object** ком sender**,** ком EventArgs ком e**)**

**{**

textCost**.**MaxLength ком **=** ком 10**;**

**}**

**private** ком void ком butChUrist\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

string ком info ком **=** ком dataGridView2**.**CurrentCell**.**Value**.**ToString**();**

Akyi**.**searchLaw**(**info**).**Surname ком **=** ком textSurUrist**.**Text**;**

Akyi**.**searchLaw**(**textSurUrist**.**Text**).**Number ком **=** ком Convert**.**ToInt64**(**textNumber**.**Text**);**

dataGridView2**.**Rows**[**dataGridView2**.**CurrentRow**.**Index**].**Cells**[**0**].**Value ком **=** ком textSurUrist**.**Text**;**

dataGridView2**.**Rows**[**dataGridView2**.**CurrentRow**.**Index**].**Cells**[**1**].**Value ком **=** ком textNumber**.**Text**;**

textSurClient**.**Clear**();**textNumber**.**Clear**();**

**}**

**private** ком void ком butDelUr\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

Akyi**.**delLaw**();**

dataGridView2**.**Rows**.**RemoveAt**(**Akyi**.**Count**);**

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

**}**

**private** ком void ком butPrint\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

MessageBox**.**Show**(**Akyi**.**getInfoComp**());**

**}**

**private** ком void ком butChCli\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

string ком infoUrist ком **=** ком dataGridView2**.**CurrentCell**.**Value**.**ToString**();**

string ком infoClient ком **=** ком dataGridView1**.**CurrentCell**.**Value**.**ToString**();**

Akyi**.**searchLaw**(**infoUrist**).**Search**(**infoClient**).**Surname ком **=** ком textSurClient**.**Text**;**

Akyi**.**searchLaw**(**infoUrist**).**Search**(**textSurClient**.**Text**).**Cost ком **=** ком Convert**.**ToInt32**(**textCost**.**Text**);**

Akyi**.**searchLaw**(**infoUrist**).**Search**(**textSurClient**.**Text**).**Service ком **=** ком textService**.**Text**;**

dataGridView1**.**Rows**[**dataGridView2**.**CurrentRow**.**Index**].**Cells**[**0**].**Value ком **=** ком textSurClient**.**Text**;**

dataGridView1**.**Rows**[**dataGridView2**.**CurrentRow**.**Index**].**Cells**[**1**].**Value ком **=** ком textService**.**Text**;**

dataGridView1**.**Rows**[**dataGridView2**.**CurrentRow**.**Index**].**Cells**[**2**].**Value ком **=** ком textCost**.**Text**;**

**}**

**private** ком void ком butDelCli\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

string ком infoUr ком **=** ком dataGridView2**.**CurrentCell**.**Value**.**ToString**();**

Akyi**.**searchLaw**(**infoUr**).**delCust**();**

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

**}**

**private** ком void ком butSave\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

string ком path**;**

string ком fileText ком **=** ком "Akyi ком Company\n"**;**

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

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

**{**

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

**{**

path ком **=** ком saveFile**.**FileName**;**

fileText ком **+=** ком Akyi**.**Info**()** ком **+** ком "\n"**;**

fileText ком **+=** ком Akyi**.**getInfoFor**();**

//Та ком же ком работа ком с ком экземплярами ком файла ком и ком записи ком файла

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

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

**{**

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

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

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

**}**

**}**

**}**

**}**

**private** ком void ком butUpload\_Click**(object** ком sender**,** ком EventArgs ком e**)**

**{**

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

int ком Count**,** ком CusCount**,** ком gName**;**

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

**{**

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

**{**

path ком **=** ком openFile**.**FileName**;**

**}**

**}**

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

**{**

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

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

**if** ком **(**doIt ком **==** ком "Akyi ком Company"**)**

**{**

**for** ком **(**int ком i ком **=** ком 0**;** ком i ком **<=** ком Akyi**.**Count**;** ком i**++)**

**{**

Akyi**.**delLaw**();**

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

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

**}**

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

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

**{**

string ком SurnameUrist ком **=** ком stream3**.**ReadLine**();**

long ком Number ком **=** ком long**.**Parse**(**stream3**.**ReadLine**());**

Akyi**.**addLaw**(**SurnameUrist**,** ком Number**,** ком 10**);**

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

dataGridView2**.**Rows**[**i ком **-** ком 1**].**Cells**[**0**].**Value ком **=** ком SurnameUrist**;**

dataGridView2**.**Rows**[**i ком **-** ком 1**].**Cells**[**1**].**Value ком **=** ком Number**;**

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

i**++;**

**else**

**break;**

**}**

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

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

**{**

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

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

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

**{**

string ком SurnCli ком **=** ком stream3**.**ReadLine**();**

string ком Service ком **=** ком stream3**.**ReadLine**();**

gName ком **=** ком int**.**Parse**(**stream3**.**ReadLine**());**

string ком name ком **=** ком Convert**.**ToString**(**dataGridView2**.**Rows**[**hall**].**Cells**[**0**].**Value**);**

Akyi**.**searchLaw**(**name**).**addCust**(**SurnCli**,** ком Service**,** ком gName**);**

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

i**++;**

**else**

**break;**

**}**

hall**++;**

**}**

**}**

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

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

**}**

**}**

**}**

**}**

// ком Сustomer ком

**using** ком System**;**

**using** ком System**.**Collections**.**Generic**;**

**using** ком System**.**Text**;**

**namespace** ком Third

**{**

class ком Customers

**{**

**private** ком string ком surname**;**//фамилия ком заказчика

**private** ком string ком service**;**//оказанная ком услуга

**private** ком int ком cost**;**//стоимость ком услуги

**public** ком Customers**(**string ком \_surname**,** ком string ком \_service**,** ком int ком \_cost**)**//конструктор

**{**

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

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

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

**}**

**public** ком string ком Surname ком //get/set ком для ком фамилии

**{**

**get** ком **{** ком **return** ком surname**;** ком **}**

**set** ком **{** ком **this.**surname ком **=** ком **value;** ком **}**

**}**

**public** ком string ком Service//get/set ком для ком оказанной ком услуги

**{**

**get** ком **{** ком **return** ком service**;** ком **}**

**set** ком **{** ком **this.**service ком **=** ком **value;** ком **}**

**}**

**public** ком int ком Cost//get/set ком для ком стоимости

**{**

**get** ком **{** ком **return** ком cost**;** ком **}**

**set** ком **{** ком **this.**cost ком **=** ком **value;** ком **}**

**}**

**}**

**}**

// ком Company

**using** ком System**;**

**using** ком System**.**Collections**.**Generic**;**

**using** ком System**.**Text**;**

**namespace** ком Third

**{**

class ком Company

**{**

**private** ком string ком nameComp**;**//название ком юр.фирмы

**private** ком Stack ком firstElenem**,**SP**;**//первый ком элемент ком в ком стеке

**private** ком int ком countElem**;**//количество ком элементов

**public** ком Company**(**string ком \_nameComp**)** ком //конструктор

**{**

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

SP ком **=** ком **null;**

firstElenem ком **=** ком **null;**

countElem ком **=** ком 0**;**

**}**

**public** ком string ком Info**()**

**{**

string ком info ком **=** ком Convert**.**ToString**(**Count**);**

Stack ком vspom ком **=** ком SP**;**

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

**{**

info ком **+=** ком "\n" ком **+** ком vspom**.**date**.**Surname**;**

info ком **+=** ком "\n" ком **+** ком vspom**.**date**.**Number**;**

vspom ком **=** ком vspom**.**getLaw**();**

**}**

**return** ком info**;**

**}**

**public** ком int ком Count ком **{** ком **get** ком **{** ком **return** ком countElem**;** ком **}** ком **}**

**public** ком void ком addLaw**(**string ком \_lSurname**,** ком long ком \_number**,** ком int ком \_size**)** ком //добавление ком юриста

**{**

Lawyer ком dobLaw ком **=** ком **new** ком Lawyer**(**\_lSurname**,** ком \_number**,** ком \_size**);**

Stack ком help ком **=** ком **new** ком Stack**(**dobLaw**,** ком SP**);**

**if** ком **(**SP ком **==** ком **null)** ком firstElenem ком **=** ком help**;**

SP ком **=** ком help**;**

countElem**++;**

**}**

**public** ком void ком delLaw**()** ком //удаление ком юриста

**{**

**if(**SP ком **!=** ком **null)**

**{**

SP ком **=** ком SP**.**getLaw**();**

countElem**--;**

**}**

**}**

**public** ком Lawyer ком searchLaw**(**string ком \_lSurname**)** ком //поиск ком юриста ком по ком фамилии

**{**

Stack ком vspom ком **=** ком SP**;**

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

**{**

**if** ком **(**vspom**.**date**.**Surname**.**Equals**(**\_lSurname**))**

**{**

**return** ком vspom**.**date**;**

**}**

**else** ком vspom ком **=** ком vspom**.**getLaw**();**

**}**

**return** ком **null;**

**}**

**public** ком int ком SumCostLaw**()**//заработок ком всей ком компании

**{**

int ком summ ком **=** ком 0**;**

Stack ком vspom ком **=** ком SP**;**

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

**{**

summ ком **+=** ком vspom**.**date**.**SumCost**();**

vspom ком **=** ком vspom**.**getLaw**();**

**}**

**return** ком summ**;**

**}**

**public** ком string ком getInfoComp**()** ком //получение ком всей ком информации ком о ком компании

**{**

string ком getInfoAll ком **=** ком "\n\nName ком Company: ком " ком **+** ком nameComp**;**

Stack ком vspom ком **=** ком SP**;**

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

**{**

getInfoAll ком **+=** ком "\n" ком **+** ком vspom**.**date**.**getInfo**()** ком **+** ком "\n"**;**

vspom ком **=** ком vspom**.**getLaw**();**

**}**

getInfoAll ком **+=** ком "\n[summ ком Cost ком all ком company: ком " ком **+** ком SumCostLaw**()** ком **+** ком "]"**;**

**return** ком getInfoAll**;**

**}**

**public** ком string ком getInfoFor**()** ком //получение ком всей ком информации ком о ком компании

**{**

string ком getInfoAll ком **=** ком ""**;**

Stack ком vspom ком **=** ком SP**;**

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

**{**

getInfoAll ком **+=** ком "\n" ком **+** ком vspom**.**date**.**getInfoFor**();**

vspom ком **=** ком vspom**.**getLaw**();**

**}**

**return** ком getInfoAll**;**

**}**

**}**

**}**

// ком Stack

**using** ком System**;**

**using** ком System**.**Collections**.**Generic**;**

**using** ком System**.**Text**;**

**namespace** ком Third

**{**

class ком Stack

**{**

**public** ком Lawyer ком date**;**//Данные

**private** ком Stack ком pred**;**//ссылка ком на ком предка

**public** ком Stack**(**Lawyer ком \_date**,** ком Stack ком \_SP**)**//конструктор

**{**

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

pred ком **=** ком \_SP**;**

**}**

**public** ком Stack ком getLaw**()**//получение ком ссылки ком на ком предка

**{**

**return** ком pred**;**

**}**

**}**

**}**

// ком Lawyer ком

**using** ком System**;**

**using** ком System**.**Collections**.**Generic**;**

**using** ком System**.**Text**;**

**namespace** ком Third

**{**

class ком Lawyer

**{**

**private** ком string ком lSurname**;**//фамилия ком юриста

**private** ком long ком number**;**

**private** ком Customers**[]** ком queue**;**//массив-очередь

**private** ком int ком first**;**//ссылка ком на ком 1 ком элемент

**private** ком int ком last**;**//ссылка ком на ком последний ком элемент

**private** ком int ком sumCost**,** ком size**,**count**;**//сумма, ком потраченная ком на ком услуги

**public** ком Lawyer**(**string ком \_lSurname**,** ком long ком \_number**,** ком int ком \_size**)** ком //конструктор

**{**

**this.**lSurname ком **=** ком \_lSurname**;**

**this.**number ком **=** ком \_number**;**

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

queue ком **=** ком **new** ком Customers**[**size**];**

first ком **=** ком 0**;** ком last ком **=** ком 0**;** ком count ком **=** ком 0**;**

**}**

**public** ком Customers ком Search**(**string ком \_lSurname**)**

**{**

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

**{**

**if(**queue**[**i**].**Surname**.**Equals**(**\_lSurname**))**

**{**

**return** ком queue**[**i**];**

**}**

**}**

**return** ком **null;**

**}**

**public** ком long ком Number ком **{** ком **get** ком **{** ком **return** ком number**;** ком **}** ком **set** ком **{** ком **this.**number ком **=** ком **value;** ком **}** ком **}**

**public** ком Customers**[]** ком Queue ком **{** ком **get** ком **{** ком **return** ком queue**;** ком **}** ком **}**

**public** ком string ком Surname**{** ком **get** ком **{** ком **return** ком lSurname**;** ком **}set** ком **{** ком **this.**lSurname ком **=** ком **value;** ком **}** ком **}**

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

**public** ком void ком addCust**(**string ком \_surname**,** ком string ком \_service**,** ком int ком \_cost**)** ком //добавление ком заказчика

**{**

**if** ком **(**size ком **>=** ком count**)**

**{**

**if** ком **(**first ком **>** ком 0 ком **&&** ком last ком **>=** ком size**)** ком last ком **=** ком 0**;**

Customers ком dobav ком **=** ком **new** ком Customers**(**\_surname**,** ком \_service**,** ком \_cost**);**

queue**[**last**]** ком **=** ком dobav**;**

last**++;**

count**++;**

**}**

**}**

**public** ком void ком delCust**()** ком //удаление ком заказчика ком по ком переменной ком first

**{** ком

**if** ком **(**count ком **!=** ком 0**)**

**{**

**if** ком **(**first ком **>** ком size**)** ком first ком **=** ком 0**;**

queue**[**first**]** ком **=** ком **null;**

count**--;**

first**++;**

**}**

**}**

**public** ком int ком SumCost**()**//возвращает ком сумму ком потраченных ком денег

**{**

sumCost ком **=** ком 0**;**

**for(** ком int ком i ком **=** ком 0**;** ком i ком **<** ком size**;** ком i**++)**

**{**

**if** ком **(**queue**[**i**]** ком **!=** ком **null)**

**{**

sumCost ком **+=** ком queue**[**i**].**Cost**;**

**}**

**}**

**return** ком sumCost**;**

**}**

**public** ком string ком getInfo**()** ком //получение ком информации ком об ком одном ком юристе ком и ком его ком заказчиках

**{**

string ком getinfo ком **=** ком "\n<=======Lawyer: ком " ком **+** ком lSurname ком **+** ком "=======>"**;**

Customers ком help**;**

**for** ком **(**int ком i ком **=** ком 0**;** ком i ком **<** ком size**;** ком i**++)**

**{**

help ком **=** ком queue**[**i**];**

**if** ком **(**help ком **!=** ком **null)**

**{**

getinfo ком **+=** ком "\n\nCustomer: ком " ком **+** ком help**.**Surname**;**

getinfo ком **+=** ком "\nService: ком " ком **+** ком help**.**Service**;**

getinfo ком **+=** ком "\nCost: ком " ком **+** ком help**.**Cost ком **+** ком "\n"**;**

**}**

**}**

getinfo ком **+=** ком "\nsum ком of ком cost: ком " ком **+** ком SumCost**();** ком

**return** ком getinfo**;**

**}**

**public** ком string ком getInfoFor**()**

**{**

string ком getinfo ком **=** ком Convert**.**ToString**(**Count**);**

Customers ком help**;**

**for** ком **(**int ком i ком **=** ком 0**;** ком i ком **<** ком size**;** ком i**++)**

**{**

help ком **=** ком queue**[**i**];**

**if** ком **(**help ком **!=** ком **null)**

**{**

getinfo ком **+=** ком "\n" ком **+** ком help**.**Surname**;**

getinfo ком **+=** ком "\n" ком **+** ком help**.**Service**;**

getinfo ком **+=** ком "\n" ком **+** ком help**.**Cost**;**

**}**

**}**

**return** ком getinfo**;**

**}**

**}**

**}**