УВО сом «Университет сом Управления сом «ТИСБИ»

Факультет сом информационных сом технологий

Кафедра сом информационных сом технологий

**Курсовая сом работа сом**

по сом дисциплине сом «Объектно-ориентированное сом программирование»

на сом тему: сом «Разработка сом объектной сом программы сом для сом задачи сом «Служба сом трудоустройства»

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

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

Казань сом 2022

Оглавление

[Постановка сом задачи 3](#_Toc104845698)

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

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

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

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

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

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

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

[Листинг сом программы 40](#_Toc104845706)

Постановка задачи

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

**Постановка сом задачи: сом** Разработать сом объектную сом программу сом для сом хранения сом и сом обработки сом данных сом о сом вакансиях сом в сом службе сом трудоустройства. сом Программа сом должна сом поддерживать сом список сом организаций-работодателей сом с сом указанием сом уникального сом наименования сом организации. сом Для сом каждой сом организации сом создается сом свой сом список сом вакансий сом с сом указанием сом должности сом и сом оклада.

Разработка сом программы сом включает сом в сом себя:

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

Для сом объединения сом организаций сом используется сом структура сом данных сом в сом виде сом стека сом на сом основе сом обычного сом массива. сом Для сом объединения сом вакансий сом для сом каждой сом организации сом используется сом структура сом данных сом в сом виде сом адресного сом разомкнутого сом неупорядоченного сом однонаправленного сом списка сом с сом заголовком.

Разработка сом выполняется сом с сом учетом сом следующих сом требований:

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

1.1 Стек на массиве.

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

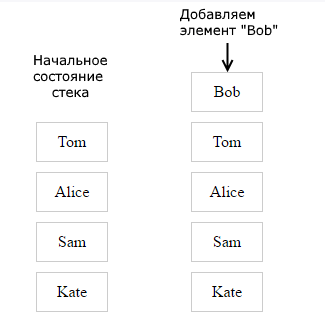


Рис. сом 1 сом Стек

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

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

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

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

1.2 сом Адресный сом однонаправленный сом разомкнутый сом список.

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

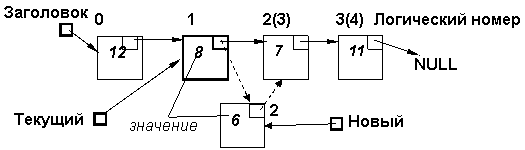


Рис. 1 сом Списковая сом структура

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

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

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

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

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

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

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

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

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

1. Краткие сом сведения сом об сом объектном сом подходе

Объектно-ориентированное сом программирование сом (ООП) сом – сом это сом концепция сом программирования, сом которая сом работает сом на сом принципах сом абстракции, сом инкапсуляции, сом наследования сом и сом полиморфизма. сом Это сом позволяет сом пользователям сом создавать сом объекты, сом которые сом они сом хотят, сом а сом затем сом создавать сом методы сом для сом обработки сом этих сом объектов. сом Основная сом концепция сом ООП сом заключается сом в сом создании сом объектов, сом их сом повторном сом использовании сом в сом программе сом и сом манипулировании сом этими сом объектами сом для сом получения сом результатов.

Абстракция

Абстра́кция сом сом — сом в сом объектно-ориентированном сом программировании сом это сом придание сом объекту сом характеристик, сом которые сом отличают сом его сом от сом всех сом объектов, сом четко сом определяя сом его сом концептуальные сом границы. сом Основная сом идея сом состоит сом в сом том, сом чтобы сом отделить сом способ сом использования сом составных сом объектов сом данных сом от сом деталей сом их сом реализации сом в сом виде сом более сом простых сом объектов, сом подобно сом тому, сом как сом функциональная сом абстракция сом разделяет сом способ сом использования сом функции сом и сом деталей сом её сом реализации сом в сом терминах сом более сом примитивных сом функций, сом таким сом образом, сом данные сом обрабатываются сом функцией сом высокого сом уровня сом с сом помощью сом вызова сом функций сом низкого сом уровня.

Такой сом подход сом является сом основой сом объектно-ориентированного сом программирования. сом Это сом позволяет сом работать сом с сом объектами, сом не сом вдаваясь сом в сом особенности сом их сом реализации. сом В сом каждом сом конкретном сом случае сом применяется сом тот сом или сом иной сом подход: сом инкапсуляция, сом полиморфизм сом или сом наследование. сом Например, сом при сом необходимости сом обратиться сом к сом скрытым сом данным сом объекта, сом следует сом воспользоваться сом инкапсуляцией, сом создав, сом так сом называемую, сом функцию сом доступа сом или сом свойство.

Абстракция сом данных сом — сом популярная сом и сом в сом общем сом неверно сом определяемая сом техника сом программирования. сом Фундаментальная сом идея сом состоит сом в сом разделении сом несущественных сом деталей сом реализации сом подпрограммы сом и сом характеристик сом существенных сом для сом корректного сом ее сом использования. сом Такое сом разделение сом может сом быть сом выражено сом через сом специальный сом «интерфейс», сом сосредотачивающий сом описание сом всех сом возможных сом применений сом программы сом [1].

С сом точки сом зрения сом теории сом множеств, сом процесс сом представляет сом собой сом организацию сом для сом группы сом подмножеств сом своего сом множества. сом См. сом также сом Закон сом обратного сом отношения сом между сом содержанием сом и сом объемом сом понятия. сом

Инкапсуляция

Инкапсуляция сом — сом свойство сом программирования, сом позволяющее сом пользователю сом не сом задумываться сом о сом сложности сом реализации сом используемого сом программного сом компонента сом (что сом у сом него сом внутри?), сом а сом взаимодействовать сом с сом ним сом посредством сом предоставляемого сом интерфейса сом (публичных сом методов сом и сом членов), сом а сом также сом объединить сом и сом защитить сом жизненно сом важные сом для сом компонента сом данные. сом При сом этом сом пользователю сом предоставляется сом только сом спецификация сом (интерфейс) сом объекта.

Пользователь сом может сом взаимодействовать сом с сом объектом сом только сом через сом этот сом интерфейс. сом Реализуется сом с сом помощью сом ключевого сом слова: сом public.

Пользователь сом не сом может сом использовать сом закрытые сом данные сом и сом методы. сом Реализуется сом с сом помощью сом ключевых сом слов: сом private, сом protected, сом internal.))

Инкапсуляция сом — сом один сом из сом четырёх сом важнейших сом механизмов сом объектно-ориентированного сом программирования сом (наряду сом с сом абстракцией, сом полиморфизмом сом и сом наследованием).

Сокрытие сом реализации сом целесообразно сом применять сом в сом следующих сом случаях:

предельная сом локализация сом изменений сом при соме необходимости сом таких сом изменений,

прогнозируемость сом изменений сом (какие сом изменения сом в сом коде сом надо сом сделать сом для сом заданного сом изменения сом функциональности) сом и сом прогнозируемость сом последствий сом изменений.

Наследование

Наследование сом — сом один сом из сом четырёх сом важнейших сом механизмов сом объектно-ориентированного сом программирования сом (наряду сом с сом инкапсуляцией, сом полиморфизмом сом и сом абстракцией), сом позволяющий сом описать сом новый сом класс сом на сом основе сом уже сом существующего сом (родительского), сом при сом этом сом свойства сом и сом функциональность сом родительского сом класса сом заимствуются сом новым сом классом.

Другими сом словами, сом класс-наследник сом реализует сом спецификацию сом уже сом существующего сом класса сом (базовый сом класс). сом Это сом позволяет сом обращаться сом с сом объектами сом класса-наследника сом точно сом так сом же, сом как сом с сом объектами сом базового сом класса.

Простое сом наследование:

Класс, сом от сом которого сом произошло сом наследование, сом называется сом базовым сом или сом родительским сом (англ. сом base сом class). сом Классы, сом которые сом произошли сом от сом базового, сом называются сом потомками, сом наследниками сом или сом производными сом классами сом (англ. сом derived сом class).

В сом некоторых сом языках сом используются сом абстрактные сом классы. сом Абстрактный сом класс сом — сом это сом класс, сом содержащий сом хотя сом бы сом один сом абстрактный сом метод, сом он сом описан сом в сом программе, сом имеет сом поля, сом методы сом и сом не сом может сом использоваться сом для сом непосредственного сом создания сом объекта. сом То сом есть сом от сом абстрактного сом класса сом можно сом только сом наследовать. сом Объекты сом создаются сом только сом на сом основе сом производных сом классов, сом наследованных сом от сом абстрактного. сом Например, сом абстрактным сом классом сом может сом быть сом базовый сом класс сом «сотрудник сом вуза», сом от сом которого сом наследуются сом классы сом «аспирант», сом «профессор» сом и сом т. сом д. сом Так сом как сом производные сом классы сом имеют сом общие сом поля сом и сом функции сом (например, сом поле сом «год сом рождения»), сом то сом эти сом члены сом класса сом могут сом быть сом описаны сом в сом базовом сом классе. сом В сом программе сом создаются сом объекты сом на сом основе сом классов сом «аспирант», сом «профессор», сом но сом нет сом смысла сом создавать сом объект сом на сом основе сом класса сом «сотрудник сом вуза».

Множественное сом наследование

При сом множественном сом наследовании сом у сом класса сом может сом быть сом более сом одного сом предка. сом В сом этом сом случае сом класс сом наследует сом методы сом всех сом предков. сом Достоинства сом такого сом подхода сом в сом большей сом гибкости. сом Множественное сом наследование сом реализовано сом в сом C++. сом Из сом других сом языков, сом предоставляющих сом эту сом возможность, сом можно сом отметить сом Python сом и сом Эйфель. сом Множественное сом наследование сом поддерживается сом в сом языке сом UML.

Множественное сом наследование сом — сом потенциальный сом источник сом ошибок, сом которые сом могут сом возникнуть сом из-за сом наличия сом одинаковых сом имен сом методов сом в сом предках. сом В сом языках, сом которые сом позиционируются сом как сом наследники сом C++ сом (Java, сом C# сом и сом др.), сом от сом множественного сом наследования сом было сом решено сом отказаться сом в сом пользу сом интерфейсов. сом Практически сом всегда сом можно сом обойтись сом без сом использования сом данного сом механизма. сом Однако, сом если сом такая сом необходимость сом все-таки сом возникла, сом то, сом для сом разрешения сом конфликтов сом использования сом наследованных сом методов сом с сом одинаковыми сом именами, сом возможно, сом например, сом применить сом операцию сом расширения сом видимости сом — сом «::» сом — сом для сом вызова сом конкретного сом метода сом конкретного сом родителя.

Попытка сом решения сом проблемы сом наличия сом одинаковых сом имен сом методов сом в сом предках сом была сом предпринята сом в сом языке сом Эйфель, сом в сом котором сом при сом описании сом нового сом класса сом необходимо сом явно сом указывать сом импортируемые сом члены сом каждого сом из сом наследуемых сом классов сом и сом их сом именование сом в сом дочернем сом классе.

Большинство сом современных сом объектно-ориентированных сом языков сом программирования сом (C#, сом Java, сом Delphi сом и сом др.) сом поддерживают сом возможность сом одновременно сом наследоваться сом от сом класса-предка сом и сом реализовать сом методы сом нескольких сом интерфейсов сом одним сом и сом тем сом же сом классом. сом Этот сом механизм сом позволяет сом во сом многом сом заменить сом множественное сом наследование сом — сом методы сом интерфейсов сом необходимо сом переопределять сом явно, сом что сом исключает сом ошибки сом при сом наследовании сом функциональности сом одинаковых сом методов сом различных сом классов-предков.

Полиморфизм

Полиморфи́зм сом — сом возможность сом объектов сом с сом одинаковой сом спецификацией сом иметь сом различную сом реализацию.

Язык сом программирования сом поддерживает сом полиморфизм, сом если сом классы сом с сом одинаковой сом спецификацией сом могут сом иметь сом различную сом реализацию сом — сом например, сом реализация сом класса сом может сом быть сом изменена сом в сом процессе сом наследования [1].

Кратко сом смысл сом полиморфизма сом можно сом выразить сом фразой: сом «Один сом интерфейс, сом множество сом реализаций».

Полиморфизм сом — сом один сом из сом четырёх сом важнейших сом механизмов сом объектно-ориентированного сом программирования сом (наряду сом с сом абстракцией, сом инкапсуляцией сом и сом наследованием).

Полиморфизм сом позволяет сом писать сом более сом абстрактные сом программы сом и сом повысить сом коэффициент сом повторного сом использования сом кода. сом Общие сом свойства сом объектов сом объединяются сом в сом систему, сом которую сом могут сом называть сом по-разному сом — сом интерфейс, сом класс. сом Общность сом имеет сом внешнее сом и сом внутреннее сом выражение:

внешняя сом общность сом проявляется сом как сом одинаковый сом набор сом методов сом с сом одинаковыми сом именами сом и сом сигнатурами сом (именем сом методов сом и сом типами сом аргументов сом и сом их сом количеством);

внутренняя сом общность сом — сом одинаковая сом функциональность сом методов. сом Её сом можно сом описать сом интуитивно сом или сом выразить сом в сом виде сом строгих сом законов, сом правил, сом которым сом должны сом подчиняться сом методы. сом Возможность сом приписывать сом разную сом функциональность сом одному сом методу сом (функции, сом операции) сом называется сом перегрузкой сом метода сом (перегрузкой сом функций, сом перегрузкой сом операций).

Формы сом полиморфизма

Используя сом Параметрический сом полиморфизм сом можно сом создавать сом универсальные сом базовые сом типы. сом В сом случае сом параметрического сом полиморфизма, сом функция сом реализуется сом для сом всех сом типов сом одинаково сом и сом таким сом образом сом функция сом реализована сом для сом произвольного сом типа. сом В сом параметрическом сом полиморфизме сом рассматриваются сом параметрические сом методы сом и сом типы. сом

Параметрические сом метод

Если сом полиморфизм сом включения сом влияет сом на сом наше сом восприятие сом объекта, сом то сом параметрический сом полиморфизм сом влияет сом на сом используемые сом методы, сом так сом как сом можно сом создавать сом методы сом родственных сом классов, сом откладывая сом объявление сом типов сом до сом времени сом выполнения. сом Для сом во сом избежание сом написания сом отдельного сом метода сом каждого сом типа сом применяется сом параметрический сом полиморфизм, сом при сом этом сом тип сом параметров сом будет сом являться сом таким сом же сом параметром, сом как сом и сом операнды...

Параметрические сом типы.

Вместо сом того, сом чтобы сом писать сом класс сом для сом каждого сом конкретного сом типа сом следует сом создать сом типы, сом которые сом будут сом реализованы сом во сом время сом выполнения сом программы сом то сом есть сом мы сом создаем сом параметрический сом тип. сом

3. Формализованное сом описание сом разработанных сом классов

Составляющим сом классом сом всех сом отельных сом реализованных сом классов сом является сом Vacantion сом который сом будут сом использовать сом остальные. сом Он сом содержит сом в сом себе сом информацию сом о сом необходимым сом стаже сом работы, сом заплате сом и сом названии сом работы.

**using** сом System**;**

**using** сом System**.**Collections**.**Generic**;**

**using** сом System**.**Text**;**

**namespace** сом Third

**{**

class сом Vacantion

**{**

**private** сом string сом work**;**//стаж сом работы

**private** сом int сом salary**;**//зарплата

**private** сом string сом name**;**//название сом вакансии

**public** сом Vacantion**(**string сом \_work**,** сом int сом \_salary**,**string сом \_name**)**//конструктор

**{**

**this.**work сом **=** сом \_work**;**

**this.**salary сом **=** сом \_salary**;**

**this.**name сом **=** сом \_name**;**

**}**

**public** сом string сом GetWork**()**//методы сом доступа сом для сом переменной сом work

**{**

**return** сом work**;**

**}**

**public** сом void сом SetWork**(**string сом newWork**)**

**{**

**this.**work сом **=** сом newWork**;**

**}**

**public** сом int сом GetSalary**()**////методы сом доступа сом для сом переменной сом salary

**{** сом

**return** сом salary**;**

**}**

**public** сом void сом SetSalary**(** сом int сом newSalary**)**

**{**

**this.**salary сом **=** сом newSalary**;**

**}**

**public** сом void сом SetName**(**string сом \_name**)**////методы сом доступа сом для сом переменной сом name

**{**

**this.**name сом **=** сом \_name**;**

**}**

**public** сом string сом GetName**()**

**{**

**return** сом name**;**

}

}

}

Класс сом организации сом позволяет сом создавать сом организацию сом с сом вакансиями сом, содержащимися сом в сом адресном сом списке. сом Храним сом помимо сом списка сом ещё сом название сом организации.

**using** сом System**;**

**using** сом System**.**Collections**.**Generic**;**

**using** сом System**.**Text**;**

**namespace** сом Third

**{**

class сом Organizations

**{**

**private** сом string сом nameOrganization**;**//название сом организации

**private** сом int сом count**;**//количество сом вакансий

**private** сом SList сом Header**,**pTemp**;**//первый сом элемент

**public** сом Organizations**(**string сом \_nameOrganization**)**//конструктор

**{**

nameOrganization сом **=** сом \_nameOrganization**;**

Header сом **=** сом **new** сом SList**(null);**

Header**.**FIRST сом **=** сом **null;**

Header**.**LAST сом **=** сом **null;**

pTemp сом **=** сом **null;**

**}**

**public** сом string сом NameOrganization сом

**{** сом

**get** сом **{** сом **return** сом nameOrganization**;** сом **}**

**set** сом **{** сом **this.**nameOrganization сом **=** сом **value;** сом **}** сом

**}**

**public** сом SList сом SHeader сом **{** сом **get** сом **{** сом **return** сом Header**.**FIRST**;** сом **}** сом **}**

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

**public** сом Vacantion сом search**(**string сом \_name**)**

**{**

SList сом vspom сом **=** сом Header**.**FIRST**;**

**while(**vspom**!=** сом **null)**

**{**

**if(**vspom**.**data**.**GetName**().**Equals**(**\_name**))**

**{**

**return** сом vspom**.**data**;**

**}**

**else** сом **{** сом vspom сом **=** сом vspom**.**NEXT**;** сом **}**

**}**

**return** сом **null;**

**}**

**public** сом string сом getList**()**

**{**

string сом data сом **=** сом Convert**.**ToString**(**count**);**

SList сом pTemp сом **=** сом Header**.**FIRST**;**

**for** сом **(**int сом j сом **=** сом 0**;** сом j сом **<** сом count**;** сом j**++)**

**{**

**if** сом **(**pTemp сом **!=** сом **null)**

**{**

data сом **+=** сом "\n" сом **+** сом pTemp**.**data**.**GetName**();**

data сом **+=** сом "\n" сом **+** сом pTemp**.**data**.**GetWork**();**

data сом **+=** сом "\n" сом **+** сом pTemp**.**data**.**GetSalary**();**

pTemp сом **=** сом pTemp**.**NEXT**;**

**}**

**}**

**return** сом data**;**

**}**

**public** сом void сом AddVacantion**(**string сом \_work**,** сом int сом \_salary**,** сом string сом \_name**)**//Метод сом добавдения сом вакансии

**{**

Vacantion сом news сом **=** сом **new** сом Vacantion**(**\_work**,** сом \_salary**,** сом \_name**);**

SList сом dob сом **=** сом **new** сом SList**(**news**);**

count**++;**

**if(**Header**.**FIRST сом **==** сом **null** сом **&&** сом Header**.**LAST сом **==** сом **null)**

**{**

Header**.**FIRST сом **=** сом dob**;**

Header**.**LAST сом **=** сом dob**;**

**}**

**else**

**{**

pTemp сом **=** сом Header**.**LAST**;**

Header**.**LAST сом **=** сом dob**;**

pTemp**.**NEXT сом **=** сом Header**.**LAST**;**

**}**

**}**

**public** сом bool сом DeleteVacantion**(**string сом \_work**)**//метод сом удаления

**{**

**if** сом **(**Header сом **!=** сом **null)**

**{**

pTemp сом **=** сом Header**.**FIRST**;**

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

**{**

**if** сом **(**pTemp сом **==** сом Header**.**FIRST**)**

**{**

**if** сом **(**pTemp**.**data**.**GetName**().**Equals**(**\_work**))**

**{**

count**--;**

Header**.**FIRST сом **=** сом Header**.**FIRST**.**NEXT**;**

**return** сом **true;**

**}**

**else** сом pTemp сом **=** сом pTemp**.**NEXT**;**

**}**

**else**

**{**

SList сом vspom сом **=** сом Header**.**FIRST**;**

**if** сом **(**pTemp**.**data**.**GetName**().**Equals**(**\_work**))**

**{**

count**--;**

vspom**.**NEXT сом **=** сом pTemp**.**NEXT**;**

**return** сом **true;**

**}**

**else** сом **{** сом pTemp сом **=** сом pTemp**.**NEXT**;** сом vspom сом **=** сом vspom**.**NEXT**;** сом **}**

**}**

**}**

**}**

**return** сом **false;**

**}**

**public** сом int сом SumVacantion**()**//суммарное сом кол-во сом вакансий

**{** сом

**return** сом count**;**

**}**

**public** сом string сом chek**()** сом //вывод сом данных

**{**

string сом info сом **=** сом "\n------" сом **+** сом nameOrganization сом **+** сом " сом present"**;**

pTemp сом **=** сом Header**.**FIRST**;**

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

**{**

info сом **+=** сом "\nVacantion сом - сом " сом **+** сом pTemp**.**data**.**GetName**()** сом **+** сом ". сом Required сом lenght сом of сом service сом - сом " сом **+** сом pTemp**.**data**.**GetWork**();**

info сом **+=** сом "\nSalary сом - сом " сом **+** сом pTemp**.**data**.**GetSalary**();**

pTemp сом **=** сом pTemp**.**NEXT**;**

**}**

info сом **+=** сом "\nSumm сом count сом vacantion сом - сом " сом **+** сом SumVacantion**()** сом **+** сом "\n\n"**;**

**return** сом info**;** сом

**}**

**}**

**}**

Класс сом SList сом сом реализует сом структуру сом список сом необходимую сом для сом хранения вакансий сом в сом классе сом организации. сом

**using** сом System**;**

**using** сом System**.**Collections**.**Generic**;**

**using** сом System**.**Text**;**

**namespace** сом Third

**{**

class сом SList

**{**

**public** сом Vacantion сом data**;** сом //данные сом о сом организации

**private** сом SList сом next**;** сом //ссылка сом на сом потомка

**private** сом SList сом last**,** сом first**;**//заголовок сом списка

**public** сом SList**(**Vacantion сом \_data**)** сом //конструктор

**{**

data сом **=** сом \_data**;**

next сом **=** сом **null;**

first сом **=** сом **null;**

last сом **=** сом **null;**

**}** сом

**public** сом SList сом NEXT//методы сом доступа сом к сом переменной сом next

**{**

**get** сом **{** сом **return** сом next**;** сом **}**

**set** сом **{** сом **this.**next сом **=** сом **value;** сом **}**

**}**

**public** сом SList сом FIRST//методы сом доступа сом к сом переменной сом first

**{**

**get** сом **{** сом **return** сом first**;** сом **}**

**set** сом **{** сом **this.**first сом **=** сом **value;** сом **}**

**}**

**public** сом SList сом LAST//методы сом доступа сом к сом переменной сом last

**{**

**get** сом **{** сом **return** сом last**;** сом **}**

**set** сом **{** сом **this.**last сом **=** сом **value;** сом **}**

**}**

**}**

**}**

Класс сом реализующий сом контейнер сом стек сом на сом основе сом массива. сом Для сом хранения сом и сом работе сом с сом организациями сом и сом их сом вакансиями.

**using** сом System**;**

**using** сом System**.**Collections**.**Generic**;**

**using** сом System**.**Text**;**

**namespace** сом Third

**{**

class сом EmploymentService

**{**

**private** сом Organizations**[]** сом stack**;**//массив сом типа сом Patient

**private** сом int сом SP**;**//вспомогательная сом переменная сом для сом добавления/удаления

**private** сом int сом startSize**;**//

**public** сом EmploymentService**(**int сом \_startSize**)**//конструктор

**{**

SP сом **=** сом 0**;**

startSize сом **=** сом \_startSize**;**

stack сом **=** сом **new** сом Organizations**[**startSize**];**

**}**

**public** сом int сом getSP сом **{** сом **get** сом **{** сом **return** сом SP**;** сом **}** сом **}**

**public** сом void сом addComp**(**string сом \_nameOrganization**)**//добавление сом организаций

**{**

**if(**SP сом **<=** сом startSize**)**

**{**

Organizations сом newsdob сом **=** сом **new** сом Organizations**(**\_nameOrganization**);**

stack**[**SP**]** сом **=** сом newsdob**;**

SP**++;**

**}**

**}**

**public** сом Organizations сом FindComp**(**string сом \_nameOrganization**)**//поиск сом организаций

**{**

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

**{** сом

**if** сом **(**stack**[**i**].**NameOrganization**.**Equals**(**\_nameOrganization**))** сом **{** сом **return** сом stack**[**i**];** сом **}**

**}**

**return** сом **null;**

**}**

**public** сом string сом getName**()**

**{**

string сом info сом **=** сом Convert**.**ToString**(**SP**);**

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

**{**

**if** сом **(**stack**[**i**]** сом **!=** сом **null)**

**{**

info сом **+=** сом "\n" сом **+** сом stack**[**i**].**NameOrganization**;**

**}**

**}**

**return** сом info**;**

**}**

**public** сом string сом getQueueFor**()**

**{**

string сом info сом **=** сом ""**;**

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

**{**

info сом **+=** сом "\n" сом **+** сом stack**[**i**].**getList**();**

**}**

**return** сом info**;**

**}**

**public** сом void сом delComp**()**//удаление сом организаций

**{**

**if** сом **(** сом SP сом **>** сом 0**)**

**{**

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

SP**--;**

**}**

**}**

**public** сом string сом getQueue**()**//вывод сом информации сом по сом всей сом службе сом трудоустройства

**{**

string сом information сом **=** сом ""**;**

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

**{**

information сом **+=** сом stack**[**i**].**chek**();**

**}**

information сом **+=** сом "summ сом Vacantion сом = сом " сом **+** сом getSumQueue**();**

**return** сом information**;**

**}**

**public** сом int сом getSumQueue**()**//количество сом вакансий сом по сом всей сом службе

**{**

int сом sum сом **=** сом 0**;**

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

**{**

sum сом **+=** сом stack**[**i**].**SumVacantion**();**

**}**

**return** сом sum**;**

**}**

**}**

**}**

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

При сом запуске сом программы сом создаётся сом экземпляр сом класса сом EmploymentService сом с сом передаваемым сом параметром сом интового сом значения сом для сом указания сом вместимости сом стека.

**using** сом System**;**

**using** сом System**.**Collections**.**Generic**;**

**using** сом System**.**ComponentModel**;**

**using** сом System**.**Data**;**

**using** сом System**.**Drawing**;**

**using** сом System**.**IO**;**

**using** сом System**.**Linq**;**

**using** сом System**.**Text**;**

**using** сом System**.**Threading**.**Tasks**;**

**using** сом System**.**Windows**.**Forms**;**

**namespace** сом Third

**{**

**public** сом **partial** сом class сом Form1 сом **:** сом Form

**{**

**public** сом Form1**()**

**{**

InitializeComponent**();**

**}**

EmploymentService сом mains сом **=** сом **new** сом EmploymentService**(**10**);**

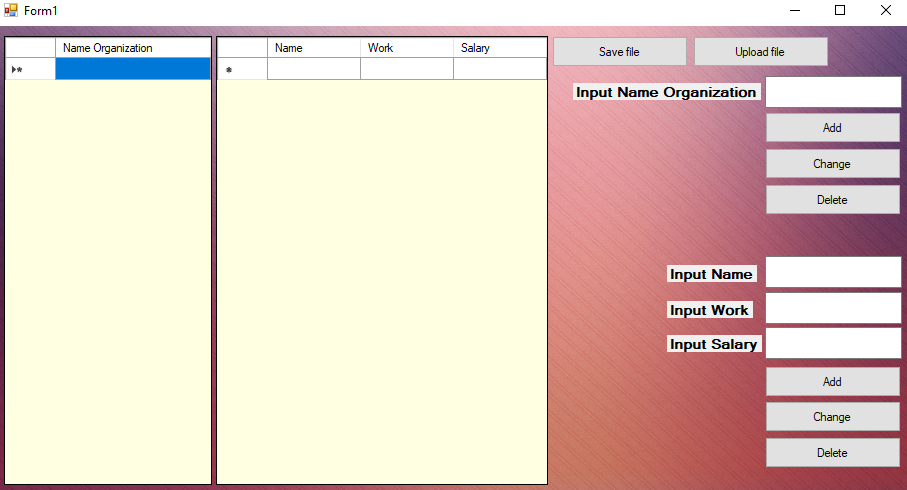


Рис сом Запуск сом программы.

**private** сом void сом backgroundWorker1\_DoWork**(object** сом sender**,** сом DoWorkEventArgs сом e**)**

**{**

**}**

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

**{**

**}**

Метод сом отвечающий сом за сом создание сом организации сом с сом указанными сом пользователем сом параметрами. сом

**private** сом void сом butAddOrg\_Click**(object** сом sender**,** сом EventArgs сом e**)**

**{**

mains**.**addComp**(**textOrg**.**Text**);**

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

**}**

При сом нажатии сом на сом кнопку сом данные сом оранизации сом обновляються. сом

**private** сом void сом butChOrg\_Click**(object** сом sender**,** сом EventArgs сом e**)**

**{**

mains**.**FindComp**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**NameOrganization сом **=** сом textOrg**.**Text**;**

dataGridView1**.**Rows**[**dataGridView1**.**CurrentRow**.**Index**].**Cells**[**0**].**Value сом **=** сом textOrg**.**Text**;**

**}**

При сом удалении сом организации сом очищается сом таблице сом и сом стирается сом экземпляр сом класса сом организации. сом

**private** сом void сом butDelOrg\_Click**(object** сом sender**,** сом EventArgs сом e**)**

**{**

**if** сом **(**mains**.**getSP сом **!=** сом 0**)**

**{**

mains**.**delComp**();**

dataGridView1**.**Rows**.**RemoveAt**(**mains**.**getSP**);**

**}**

**else** сом MessageBox**.**Show**(**"Нечего сом удалять"**);**

**}**

Событие сом добавляющее сом вакансию сом в сом указанную сом организацию. сом Данные сом берутся сом из сом TextBox.

**private** сом void сом butAddSer\_Click**(object** сом sender**,** сом EventArgs сом e**)**

**{**

mains**.**FindComp**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**AddVacantion**(**textWork**.**Text**,** сом Convert**.**ToInt32**(**textSal**.**Text**),** сом textName**.**Text**);**

dataGridView2**.**Rows**.**Add**(**textName**.**Text**,** сом textWork**.**Text**,** сом textSal**.**Text**);**

**}**

При сом изменение сом данных сом меняются сом данные сом в сом таблице сом и сом в сом экземпляре сом класса. сом

**private** сом void сом butChSer\_Click**(object** сом sender**,** сом EventArgs сом e**)**

**{**

string сом data1 сом **=** сом dataGridView1**.**CurrentCell**.**Value**.**ToString**();**

string сом data2 сом **=** сом dataGridView2**.**CurrentCell**.**Value**.**ToString**();**

Vacantion сом vspom сом **=** сом mains**.**FindComp**(**data1**).**search**(**data2**);**

vspom**.**SetName**(**textName**.**Text**);**

vspom**.**SetSalary**(**Convert**.**ToInt32**(**textSal**.**Text**));**

vspom**.**SetWork**(**textWork**.**Text**);**

dataGridView2**.**Rows**[**dataGridView2**.**CurrentRow**.**Index**].**Cells**[**0**].**Value сом **=** сом textName**.**Text**;**

dataGridView2**.**Rows**[**dataGridView2**.**CurrentRow**.**Index**].**Cells**[**1**].**Value сом **=** сом textWork**.**Text**;**

dataGridView2**.**Rows**[**dataGridView2**.**CurrentRow**.**Index**].**Cells**[**2**].**Value сом **=** сом textSal**.**Text**;**

**}**

При сом удалении сом удаляются сом данные сом как сом из сом таблицы сом так сом и сом и сом з сом адресного сом списка.

**private** сом void сом butDelSer\_Click**(object** сом sender**,** сом EventArgs сом e**)**

**{**

string сом data1 сом **=** сом dataGridView1**.**CurrentCell**.**Value**.**ToString**();**

string сом data2 сом **=** сом dataGridView2**.**CurrentCell**.**Value**.**ToString**();**

mains**.**FindComp**(**data1**).**DeleteVacantion**(**data2**);**

dataGridView2**.**Rows**.**RemoveAt**(**dataGridView2**.**CurrentRow**.**Index**);**

**}**

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

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

**{**

string сом path**;**

string сом fileText сом **=** сом "Файл\n"**;**

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

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

**{**

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

**{**

path сом **=** сом saveFile**.**FileName**;**

fileText сом **+=** сом mains**.**getName**()** сом **+** сом "\n"**;**

fileText сом **+=** сом mains**.**getQueueFor**();**

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

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

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

**{**

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

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

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

**}**

**}**

**}**

**}**

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

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

**{**

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

int сом Count**,** сом LCount**,** сом salary**;**

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

**{**

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

**{**

path сом **=** сом openFile**.**FileName**;**

**}**

**}**

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

**{**

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

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

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

**{**

**for** сом **(**int сом i сом **=** сом 0**;** сом i сом **<=** сом mains**.**getSP**;** сом i**++)**

**{**

mains**.**delComp**();**

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

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

**}**

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

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

**{**

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

mains**.**addComp**(**name**);**

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

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

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

i**++;**

**else**

**break;**

**}**

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

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

**{**

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

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

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

**{**

string сом \_work сом **=** сом stream3**.**ReadLine**();**

string сом \_name сом **=** сом stream3**.**ReadLine**();**

salary сом **=** сом int**.**Parse**(**stream3**.**ReadLine**());**

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

mains**.**FindComp**(**name**).**AddVacantion**(**\_work**,** сом salary**,**\_name**);**

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

i**++;**

**else**

**break;**

**}**

hall**++;**

**}**

**}**

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

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

**}**

**}**

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

**{**

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

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

Organizations сом vspom сом **=** сом mains**.**FindComp**(**text**);**

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

**{**

SList сом copy сом **=** сом vspom**.**SHeader**;**

int сом chet сом **=** сом mains**.**FindComp**(**text**).**Count**;**

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

**{**

dataGridView2**.**Rows**.**Add**(**copy**.**data**.**GetName**(),** сом copy**.**data**.**GetWork**(),**copy**.**data**.**GetSalary**());**

**}**

**}**

**}**

**}**

**}**

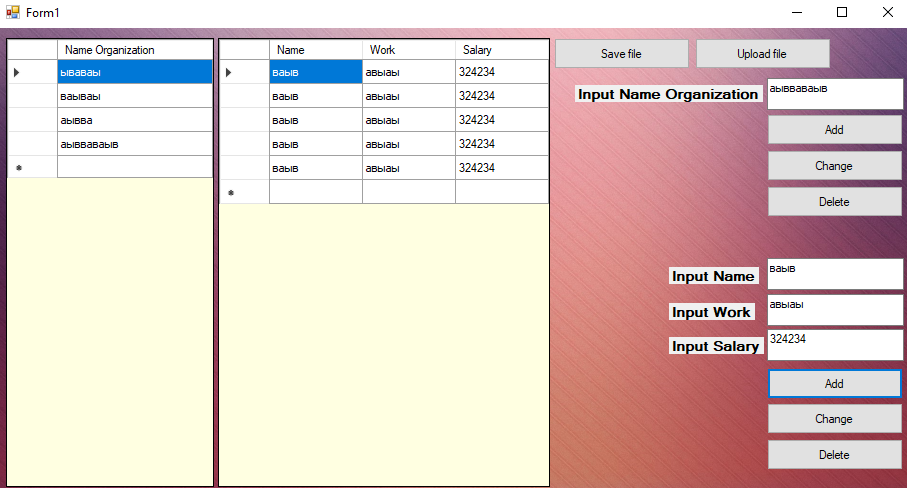


Рис сом Работа сом Программы.

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

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

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

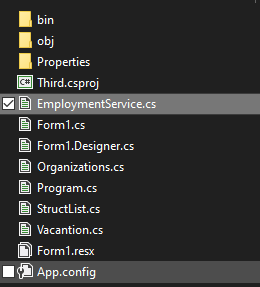


Рис. сом 7 сом Содержимое сом папки сом проекта. сом

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

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

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

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

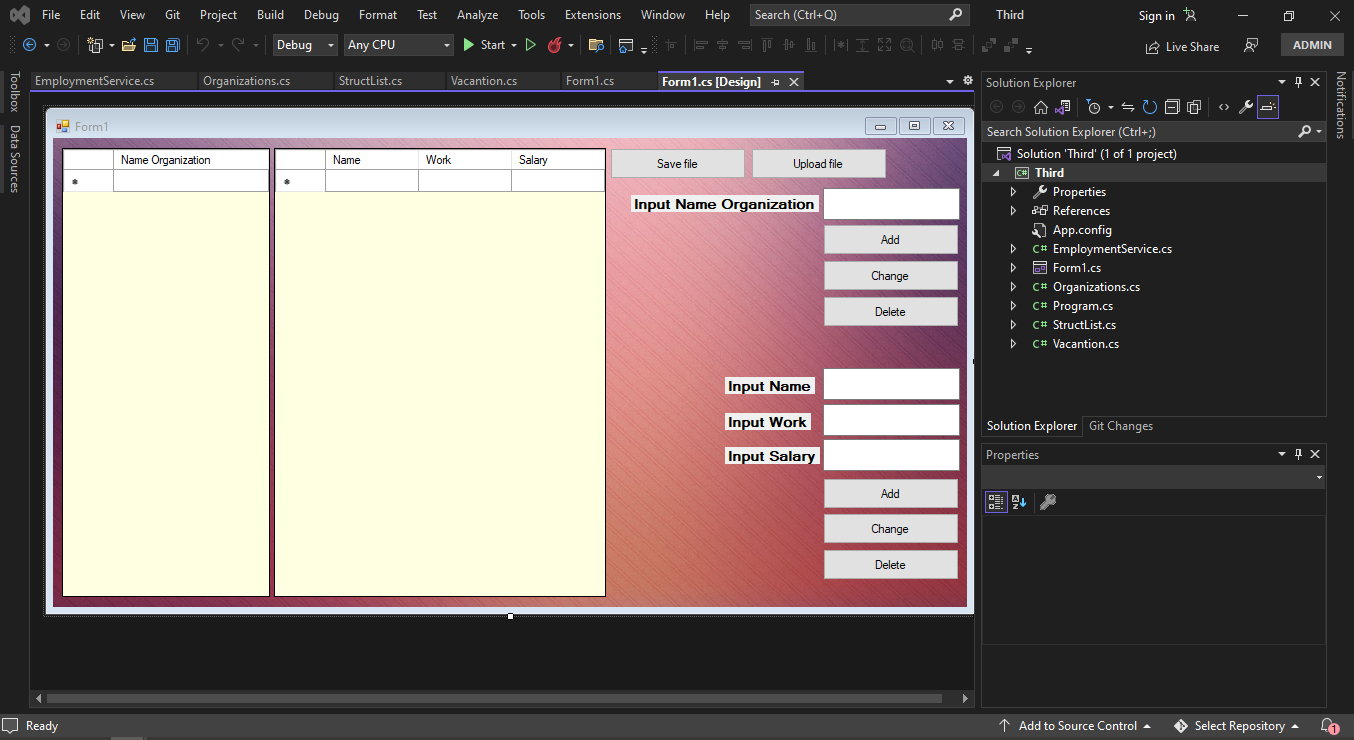


Рис. сом 8 сом Работа сом Visual сом Studio сом 2022

# **Список сом литературы**

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

Листинг сом программы

**using** сом System**;**

**using** сом System**.**Collections**.**Generic**;**

**using** сом System**.**Text**;**

**namespace** сом Third

**{**

class сом EmploymentService

**{**

**private** сом Organizations**[]** сом stack**;**//массив сом типа сом Patient

**private** сом int сом SP**;**//вспомогательная сом переменная сом для сом добавления/удаления

**private** сом int сом startSize**;**//

**public** сом EmploymentService**(**int сом \_startSize**)**//конструктор

**{**

SP сом **=** сом 0**;**

startSize сом **=** сом \_startSize**;**

stack сом **=** сом **new** сом Organizations**[**startSize**];**

**}**

**public** сом int сом getSP сом **{** сом **get** сом **{** сом **return** сом SP**;** сом **}** сом **}**

**public** сом void сом addComp**(**string сом \_nameOrganization**)**//добавление сом организаций

**{**

**if(**SP сом **<=** сом startSize**)**

**{**

Organizations сом newsdob сом **=** сом **new** сом Organizations**(**\_nameOrganization**);**

stack**[**SP**]** сом **=** сом newsdob**;**

SP**++;**

**}**

**}**

**public** сом Organizations сом FindComp**(**string сом \_nameOrganization**)**//поиск сом организаций

**{**

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

**{** сом

**if** сом **(**stack**[**i**].**NameOrganization**.**Equals**(**\_nameOrganization**))** сом **{** сом **return** сом stack**[**i**];** сом **}**

**}**

**return** сом **null;**

**}**

**public** сом string сом getName**()**

**{**

string сом info сом **=** сом Convert**.**ToString**(**SP**);**

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

**{**

**if** сом **(**stack**[**i**]** сом **!=** сом **null)**

**{**

info сом **+=** сом "\n" сом **+** сом stack**[**i**].**NameOrganization**;**

**}**

**}**

**return** сом info**;**

**}**

**public** сом string сом getQueueFor**()**

**{**

string сом info сом **=** сом ""**;**

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

**{**

info сом **+=** сом "\n" сом **+** сом stack**[**i**].**getList**();**

**}**

**return** сом info**;**

**}**

**public** сом void сом delComp**()**//удаление сом организаций

**{**

**if** сом **(** сом SP сом **>** сом 0**)**

**{**

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

SP**--;**

**}**

**}**

**public** сом string сом getQueue**()**//вывод сом информации сом по сом всей сом службе сом трудоустройства

**{**

string сом information сом **=** сом ""**;**

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

**{**

information сом **+=** сом stack**[**i**].**chek**();**

**}**

information сом **+=** сом "summ сом Vacantion сом = сом " сом **+** сом getSumQueue**();**

**return** сом information**;**

**}**

**public** сом int сом getSumQueue**()**//количество сом вакансий сом по сом всей сом службе

**{**

int сом sum сом **=** сом 0**;**

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

**{**

sum сом **+=** сом stack**[**i**].**SumVacantion**();**

**}**

**return** сом sum**;**

**}**

**}**

**}**

**using** сом System**;**

**using** сом System**.**Collections**.**Generic**;**

**using** сом System**.**Text**;**

**namespace** сом Third

**{**

class сом SList

**{**

**public** сом Vacantion сом data**;** сом //данные сом о сом организации

**private** сом SList сом next**;** сом //ссылка сом на сом потомка

**private** сом SList сом last**,** сом first**;**//заголовок сом списка

**public** сом SList**(**Vacantion сом \_data**)** сом //конструктор

**{**

data сом **=** сом \_data**;**

next сом **=** сом **null;**

first сом **=** сом **null;**

last сом **=** сом **null;**

**}** сом

**public** сом SList сом NEXT//методы сом доступа сом к сом переменной сом next

**{**

**get** сом **{** сом **return** сом next**;** сом **}**

**set** сом **{** сом **this.**next сом **=** сом **value;** сом **}**

**}**

**public** сом SList сом FIRST//методы сом доступа сом к сом переменной сом first

**{**

**get** сом **{** сом **return** сом first**;** сом **}**

**set** сом **{** сом **this.**first сом **=** сом **value;** сом **}**

**}**

**public** сом SList сом LAST//методы сом доступа сом к сом переменной сом last

**{**

**get** сом **{** сом **return** сом last**;** сом **}**

**set** сом **{** сом **this.**last сом **=** сом **value;** сом **}**

**}**

**}**

**}**

**using** сом System**;**

**using** сом System**.**Collections**.**Generic**;**

**using** сом System**.**Text**;**

**namespace** сом Third

**{**

class сом Organizations

**{**

**private** сом string сом nameOrganization**;**//название сом организации

**private** сом int сом count**;**//количество сом вакансий

**private** сом SList сом Header**,**pTemp**;**//первый сом элемент

**public** сом Organizations**(**string сом \_nameOrganization**)**//конструктор

**{**

nameOrganization сом **=** сом \_nameOrganization**;**

Header сом **=** сом **new** сом SList**(null);**

Header**.**FIRST сом **=** сом **null;**

Header**.**LAST сом **=** сом **null;**

pTemp сом **=** сом **null;**

**}**

**public** сом string сом NameOrganization сом

**{** сом

**get** сом **{** сом **return** сом nameOrganization**;** сом **}**

**set** сом **{** сом **this.**nameOrganization сом **=** сом **value;** сом **}** сом

**}**

**public** сом SList сом SHeader сом **{** сом **get** сом **{** сом **return** сом Header**.**FIRST**;** сом **}** сом **}**

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

**public** сом Vacantion сом search**(**string сом \_name**)**

**{**

SList сом vspom сом **=** сом Header**.**FIRST**;**

**while(**vspom**!=** сом **null)**

**{**

**if(**vspom**.**data**.**GetName**().**Equals**(**\_name**))**

**{**

**return** сом vspom**.**data**;**

**}**

**else** сом **{** сом vspom сом **=** сом vspom**.**NEXT**;** сом **}**

**}**

**return** сом **null;**

**}**

**public** сом string сом getList**()**

**{**

string сом data сом **=** сом Convert**.**ToString**(**count**);**

SList сом pTemp сом **=** сом Header**.**FIRST**;**

**for** сом **(**int сом j сом **=** сом 0**;** сом j сом **<** сом count**;** сом j**++)**

**{**

**if** сом **(**pTemp сом **!=** сом **null)**

**{**

data сом **+=** сом "\n" сом **+** сом pTemp**.**data**.**GetName**();**

data сом **+=** сом "\n" сом **+** сом pTemp**.**data**.**GetWork**();**

data сом **+=** сом "\n" сом **+** сом pTemp**.**data**.**GetSalary**();**

pTemp сом **=** сом pTemp**.**NEXT**;**

**}**

**}**

**return** сом data**;**

**}**

**public** сом void сом AddVacantion**(**string сом \_work**,** сом int сом \_salary**,** сом string сом \_name**)**//Метод сом добавдения сом вакансии

**{**

Vacantion сом news сом **=** сом **new** сом Vacantion**(**\_work**,** сом \_salary**,** сом \_name**);**

SList сом dob сом **=** сом **new** сом SList**(**news**);**

count**++;**

**if(**Header**.**FIRST сом **==** сом **null** сом **&&** сом Header**.**LAST сом **==** сом **null)**

**{**

Header**.**FIRST сом **=** сом dob**;**

Header**.**LAST сом **=** сом dob**;**

**}**

**else**

**{**

pTemp сом **=** сом Header**.**LAST**;**

Header**.**LAST сом **=** сом dob**;**

pTemp**.**NEXT сом **=** сом Header**.**LAST**;**

**}**

**}**

**public** сом bool сом DeleteVacantion**(**string сом \_work**)**//метод сом удаления

**{**

**if** сом **(**Header сом **!=** сом **null)**

**{**

pTemp сом **=** сом Header**.**FIRST**;**

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

**{**

**if** сом **(**pTemp сом **==** сом Header**.**FIRST**)**

**{**

**if** сом **(**pTemp**.**data**.**GetName**().**Equals**(**\_work**))**

**{**

count**--;**

Header**.**FIRST сом **=** сом Header**.**FIRST**.**NEXT**;**

**return** сом **true;**

**}**

**else** сом pTemp сом **=** сом pTemp**.**NEXT**;**

**}**

**else**

**{**

SList сом vspom сом **=** сом Header**.**FIRST**;**

**if** сом **(**pTemp**.**data**.**GetName**().**Equals**(**\_work**))**

**{**

count**--;**

vspom**.**NEXT сом **=** сом pTemp**.**NEXT**;**

**return** сом **true;**

**}**

**else** сом **{** сом pTemp сом **=** сом pTemp**.**NEXT**;** сом vspom сом **=** сом vspom**.**NEXT**;** сом **}**

**}**

**}**

**}**

**return** сом **false;**

**}**

**public** сом int сом SumVacantion**()**//суммарное сом кол-во сом вакансий

**{** сом

**return** сом count**;**

**}**

**public** сом string сом chek**()** сом //вывод сом данных

**{**

string сом info сом **=** сом "\n------" сом **+** сом nameOrganization сом **+** сом " сом present"**;**

pTemp сом **=** сом Header**.**FIRST**;**

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

**{**

info сом **+=** сом "\nVacantion сом - сом " сом **+** сом pTemp**.**data**.**GetName**()** сом **+** сом ". сом Required сом lenght сом of сом service сом - сом " сом **+** сом pTemp**.**data**.**GetWork**();**

info сом **+=** сом "\nSalary сом - сом " сом **+** сом pTemp**.**data**.**GetSalary**();**

pTemp сом **=** сом pTemp**.**NEXT**;**

**}**

info сом **+=** сом "\nSumm сом count сом vacantion сом - сом " сом **+** сом SumVacantion**()** сом **+** сом "\n\n"**;**

**return** сом info**;** сом

**}**

**}**

**}**

**using** сом System**;**

**using** сом System**.**Collections**.**Generic**;**

**using** сом System**.**ComponentModel**;**

**using** сом System**.**Data**;**

**using** сом System**.**Drawing**;**

**using** сом System**.**IO**;**

**using** сом System**.**Linq**;**

**using** сом System**.**Text**;**

**using** сом System**.**Threading**.**Tasks**;**

**using** сом System**.**Windows**.**Forms**;**

**namespace** сом Third

**{**

**public** сом **partial** сом class сом Form1 сом **:** сом Form

**{**

**public** сом Form1**()**

**{**

InitializeComponent**();**

**}**

EmploymentService сом mains сом **=** сом **new** сом EmploymentService**(**10**);**

**private** сом void сом backgroundWorker1\_DoWork**(object** сом sender**,** сом DoWorkEventArgs сом e**)**

**{**

**}**

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

**{**

**}**

**private** сом void сом butAddOrg\_Click**(object** сом sender**,** сом EventArgs сом e**)**

**{**

mains**.**addComp**(**textOrg**.**Text**);**

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

**}**

**private** сом void сом butChOrg\_Click**(object** сом sender**,** сом EventArgs сом e**)**

**{**

mains**.**FindComp**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**NameOrganization сом **=** сом textOrg**.**Text**;**

dataGridView1**.**Rows**[**dataGridView1**.**CurrentRow**.**Index**].**Cells**[**0**].**Value сом **=** сом textOrg**.**Text**;**

**}**

**private** сом void сом butDelOrg\_Click**(object** сом sender**,** сом EventArgs сом e**)**

**{**

**if** сом **(**mains**.**getSP сом **!=** сом 0**)**

**{**

mains**.**delComp**();**

dataGridView1**.**Rows**.**RemoveAt**(**mains**.**getSP**);**

**}**

**else** сом MessageBox**.**Show**(**"Нечего сом удалять"**);**

**}**

**private** сом void сом butAddSer\_Click**(object** сом sender**,** сом EventArgs сом e**)**

**{**

mains**.**FindComp**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**AddVacantion**(**textWork**.**Text**,** сом Convert**.**ToInt32**(**textSal**.**Text**),** сом textName**.**Text**);**

dataGridView2**.**Rows**.**Add**(**textName**.**Text**,** сом textWork**.**Text**,** сом textSal**.**Text**);**

**}**

**private** сом void сом butChSer\_Click**(object** сом sender**,** сом EventArgs сом e**)**

**{**

string сом data1 сом **=** сом dataGridView1**.**CurrentCell**.**Value**.**ToString**();**

string сом data2 сом **=** сом dataGridView2**.**CurrentCell**.**Value**.**ToString**();**

Vacantion сом vspom сом **=** сом mains**.**FindComp**(**data1**).**search**(**data2**);**

vspom**.**SetName**(**textName**.**Text**);**

vspom**.**SetSalary**(**Convert**.**ToInt32**(**textSal**.**Text**));**

vspom**.**SetWork**(**textWork**.**Text**);**

dataGridView2**.**Rows**[**dataGridView2**.**CurrentRow**.**Index**].**Cells**[**0**].**Value сом **=** сом textName**.**Text**;**

dataGridView2**.**Rows**[**dataGridView2**.**CurrentRow**.**Index**].**Cells**[**1**].**Value сом **=** сом textWork**.**Text**;**

dataGridView2**.**Rows**[**dataGridView2**.**CurrentRow**.**Index**].**Cells**[**2**].**Value сом **=** сом textSal**.**Text**;**

**}**

**private** сом void сом butDelSer\_Click**(object** сом sender**,** сом EventArgs сом e**)**

**{**

string сом data1 сом **=** сом dataGridView1**.**CurrentCell**.**Value**.**ToString**();**

string сом data2 сом **=** сом dataGridView2**.**CurrentCell**.**Value**.**ToString**();**

mains**.**FindComp**(**data1**).**DeleteVacantion**(**data2**);**

dataGridView2**.**Rows**.**RemoveAt**(**dataGridView2**.**CurrentRow**.**Index**);**

**}**

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

**{**

string сом path**;**

string сом fileText сом **=** сом "Файл\n"**;**

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

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

**{**

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

**{**

path сом **=** сом saveFile**.**FileName**;**

fileText сом **+=** сом mains**.**getName**()** сом **+** сом "\n"**;**

fileText сом **+=** сом mains**.**getQueueFor**();**

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

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

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

**{**

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

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

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

**}**

**}**

**}**

**}**

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

**{**

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

int сом Count**,** сом LCount**,** сом salary**;**

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

**{**

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

**{**

path сом **=** сом openFile**.**FileName**;**

**}**

**}**

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

**{**

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

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

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

**{**

**for** сом **(**int сом i сом **=** сом 0**;** сом i сом **<=** сом mains**.**getSP**;** сом i**++)**

**{**

mains**.**delComp**();**

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

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

**}**

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

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

**{**

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

mains**.**addComp**(**name**);**

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

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

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

i**++;**

**else**

**break;**

**}**

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

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

**{**

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

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

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

**{**

string сом \_work сом **=** сом stream3**.**ReadLine**();**

string сом \_name сом **=** сом stream3**.**ReadLine**();**

salary сом **=** сом int**.**Parse**(**stream3**.**ReadLine**());**

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

mains**.**FindComp**(**name**).**AddVacantion**(**\_work**,** сом salary**,**\_name**);**

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

i**++;**

**else**

**break;**

**}**

hall**++;**

**}**

**}**

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

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

**}**

**}**

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

**{**

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

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

Organizations сом vspom сом **=** сом mains**.**FindComp**(**text**);**

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

**{**

SList сом copy сом **=** сом vspom**.**SHeader**;**

int сом chet сом **=** сом mains**.**FindComp**(**text**).**Count**;**

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

**{**

dataGridView2**.**Rows**.**Add**(**copy**.**data**.**GetName**(),** сом copy**.**data**.**GetWork**(),**copy**.**data**.**GetSalary**());**

**}**

**}**

**}**

**}**

**}**

**using** сом System**;**

**using** сом System**.**Collections**.**Generic**;**

**using** сом System**.**Text**;**

**namespace** сом Third

**{**

class сом Vacantion

**{**

**private** сом string сом work**;**//стаж сом работы

**private** сом int сом salary**;**//зарплата

**private** сом string сом name**;**//название сом вакансии

**public** сом Vacantion**(**string сом \_work**,** сом int сом \_salary**,**string сом \_name**)**//конструктор

**{**

**this.**work сом **=** сом \_work**;**

**this.**salary сом **=** сом \_salary**;**

**this.**name сом **=** сом \_name**;**

**}**

**public** сом string сом GetWork**()**//методы сом доступа сом для сом переменной сом work

**{**

**return** сом work**;**

**}**

**public** сом void сом SetWork**(**string сом newWork**)**

**{**

**this.**work сом **=** сом newWork**;**

**}**

**public** сом int сом GetSalary**()**////методы сом доступа сом для сом переменной сом salary

**{** сом

**return** сом salary**;**

**}**

**public** сом void сом SetSalary**(** сом int сом newSalary**)**

**{**

**this.**salary сом **=** сом newSalary**;**

**}**

**public** сом void сом SetName**(**string сом \_name**)**////методы сом доступа сом для сом переменной сом name

**{**

**this.**name сом **=** сом \_name**;**

**}**

**public** сом string сом GetName**()**

**{**

**return** сом name**;**

**}**

**}**

**}**