Учреждение род высшего род образования

род «Университет род управления род «ТИСБИ»

Факультет род информационных род технологий

Кафедра род информационных род технологий род

КУРСОВАЯ род РАБОТА

по род дисциплине род «Программная род инженерия»

на род тему: род «Разработка род объектной род программы род для род учета род налогов род физических род лиц»

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

Фамилия род И.О.

Проверил: род ст. род преподаватель

Якунина род Е.А.

Казань род 2022

Оглавление

[Постановка род задачи 3](#_Toc105329067)

[1. Описание род используемых род структур род данных род с род алгоритмами род выполнения род основных род операций 5](#_Toc105329068)

[1.1 род Динамическая род очередь. 5](#_Toc105329069)

[1.2 род Адресный род однонаправленный род список. 6](#_Toc105329070)

[2. Краткие род сведения род об род объектном род подходе 8](#_Toc105329071)

[3. Формализованное род описание род разработанных род классов 14](#_Toc105329072)

[4. род Описание род демонстрационного род модуля род с род характеристикой род использованных род стандартных род компонентов род и род списком род реализованных род обработчиков 21](#_Toc105329073)

[5. Описание род структуры род проекта род в род соответствии род с род использованным род инструментом род разработки 28](#_Toc105329074)

[Список род литературы 30](#_Toc105329075)

[Листинг род программы 32](#_Toc105329076)

Постановка род задачи

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

**Постановка род задачи: род** разработать род объектную род программу род для род хранения род и род обработки род данных род об род уплате род налогов род физическими род лицами. род Программа род должна род поддерживать род список род налогоплательщиков род с род указанием род уникального род ИНН род и род фамилии. род Для род каждого род налогоплательщика род создается род свой род список род уплаченных род налогов род с род указанием род вида род налога род (квартира, род дом, род земля, род дача, род автомобиль) род и род его род сумм.

Разработка род программы род включает род в род себя:

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

Для род объединения род налогоплательщиков род используется род структура род данных род в род виде род адресного род замкнутого род упорядоченного род однонаправленного род списка род с род заголовком. род Для род объединения род налогов род у род каждого род налогоплательщика род используется род стек род на род основе род динамического род массива.

Разработка род выполняется род с род учетом род следующих род требований:

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

1.1 род Динамическая род очередь.

Очередь род – род это род динамическая род структура род данных, род которая род состоит род из род набора род элементов род которые род размещены род последовательно род друг род за род другом. род При род этом род добавление род элементов род осуществляется род с род одной род стороны, род а род удаление род (вытягивание) род – род с род другой род стороны.

Очередь род работает род по род принципу род FIFO род (First род In род — род First род Out), род то род есть род «первым род пришел род – род первым род вышел». род На род рисунке род 1 род отображен род принцип род работы род очереди.

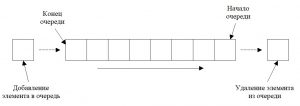


Рисунок род 1. род Динамическая род структура род данных род «очередь»

Одной род из род особенностей род очереди род есть род то, род что род в род очереди род часть род последовательных род элементов род может род быть род занята род на род данный род момент род времени род (рисунок род 2).

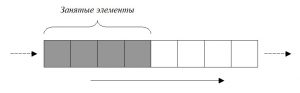


Рисунок род 2. род Очередь, род в род которой род часть род элементов род занята род на род данный род момент род времени

В род повседневной род жизни род использование род очереди род встречается род довольно род часто, род например:

очередь род в род магазине;

очередь род документов род на род печать род на род принтере;

последовательность род операторов род в род алгоритме род которые род выполняются род друг род за род другом;

другие род случаи.

Виды род очередей

Различают род следующие род виды род очередей:

простая род очередь;

кольцевая род очередь. род В род такой род очереди род элемент, род который род выходит род с род начала род очереди, род будет род помещен род в род ее род конец;

очередь род с род приоритетами. род В род такой род очереди род элементы род размещаются род по род их род приоритетам род (весовым род коэффициентам). род Первым род из род очереди род выходит род элемент род с род наивысшим род приоритетом.

Любой род вид род очереди род может род иметь род ограничение род по род размеру род (количество род элементов род в род очереди).

Способы род реализации род очереди

В род программе род очередь род можно род реализовывать род в род виде:

статического род массива род с род ограничением род на род размер род в род очереди;

динамического род массива;

односвязного род списка;

двусвязного род списка.

1.2 род Адресный род однонаправленный род список.

Однонаправленный род (односвязный) род список род – род это род структура род данных, род представляющая род собой род последовательность род элементов, род в род каждом род из род которых род хранится род значение род и род указатель род на род следующий род элемент род списка род ( род рис. род 3). род В род последнем род элементе род указатель род на род следующий род элемент род равен род NULL.

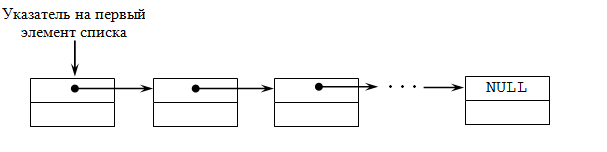


Рис. род 3. род Линейный род однонаправленный род список

Основными род операциями, род осуществляемыми род с род однонаправленными род списками, род являются:

* создание род списка;
* печать род (просмотр) род списка;
* вставка род элемента род в род список;
* удаление род элемента род из род списка;
* поиск род элемента род в род списке
* проверка род пустоты род списка;
* удаление род списка.

Особое род внимание род следует род обратить род на род то, род что род при род выполнении род любых род операций род с род линейным род однонаправленным род списком род необходимо род обеспечивать род позиционирование род какого-либо род указателя род на род первый род элемент. род В род противном род случае род часть род или род весь род список род будет род недоступен.

1. Краткие род сведения род об род объектном род подходе

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

Абстракция

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

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

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

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

Инкапсуляция

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

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

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

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

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

предельная род локализация род изменений род при род необходимости род таких род изменений,

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

Наследование

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

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

Простое род наследование:

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

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

Множественное род наследование

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

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

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

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

Полиморфизм

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

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

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

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

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

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

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

Формы род полиморфизма

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

Параметрические род метод

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

Параметрические род типы.

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

3. Формализованное род описание род разработанных род классов

Самый род главный род класс род для род налоговой род – род это род сами род налоги. род В род классе род tax род хранятся род данные род о род размере род налога, род вид род налога.

**using** род System**;**

**using** род System**.**Collections**.**Generic**;**

**using** род System**.**Text**;**

**namespace** род step3

**{**

class род tax

**{**

**private** род string род typeTax**;** род //тип род налога

**private** род int род amount**;** род //сумма род налога

**public** род tax**(**string род atypeTax**,** род int род aMount**)**//конструктор

**{**

**this.**typeTax род **=** род atypeTax**;**

**this.**amount род **=** род aMount**;**

**}** род

**public** род string род getTypeTax**(){** род **return** род typeTax**;** род **}** род //получить род тип род налога

**public** род void род setTypeTax**(**string род \_type**){** род **this.**typeTax род **=** род \_type**;** род **}** род //поменять род тип род налога

**public** род int род getAmount**()** род **{** род **return** род amount**;** род **}** род //получить род сумму род налога

**public** род void род setAmount**(**int род \_amount**)** род **{** род **this.**amount род **=** род \_amount**;** род **}** род //поменять род сумму род налога

**}**

**}**

В род файле род taxpayer род написано род 2 род класса. род Так род как род за род год род у род налогоплательщика род накапливается род несколько род видов род налога, род то род для род хранения род данных род используется род адресный род список род реализованный род в род классе род dinList. род

**using** род System**;**

**using** род System**.**Collections**.**Generic**;**

**using** род System**.**Text**;**

**namespace** род step3

**{**

class род taxpayer

**{**

**private** род string род surname**;** род //фамилия род налогоплательщика

**private** род long род INN**;**

**private** род int род count**,** род vibor**;** род //ИНН род налогоплательщика

**private** род dinList род pHead**,**Temp**,**vspom**,**Temp2**;**//ссылка род на род первый род элемент

**public** род taxpayer**(**string род \_surname**,** род long род \_INN**)** род //конструктор

**{**

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

**this.**INN род **=** род \_INN**;**

pHead род **=** род **new** род dinList**(null);**

pHead**.**First род **=** род **null;**

pHead**.**Last род **=** род **null;**

**}**

**public** род dinList род PHead род **{** род **get** род **{** род **return** род pHead**.**First**;** род **}** род **}**

**public** род string род Surname род

**{**

**get** род **{** род **return** род surname**;** род **}**

**set** род **{** род surname род **=** род **value;** род **}**

**}**

**public** род long род IINN

**{**

**get** род **{** род **return** род INN**;** род **}**

**set** род **{** род INN род **=** род **value;** род **}**

**}**

**public** род int род Count

**{**

**get** род **{** род **return** род count**;** род **}**

**}**

**public** род void род addForUpload**(**string род addTypeTax**,** род int род aMount**)**

**{**

tax род **add** род **=** род **new** род tax**(**addTypeTax**,** род aMount**);**

dinList род dobav род **=** род **new** род dinList**(add);**

count**++;**

**if** род **(**pHead**.**Last род **==** род **null)**

**{**

pHead**.**First род **=** род dobav**;**

pHead**.**Last род **=** род dobav**;**

**}**

**else**

**{**

Temp род **=** род pHead**.**Last**;**

pHead**.**Last род **=** род dobav**;**

Temp**.**Next род **=** род pHead**.**Last**;**

**}**

**}**

**public** род void род addtaxVtoroe**(**string род \_choose**,**string род addTypeTax**,** род int род aMount**,**int род i**)** род //добавление род налога

**{**

tax род **add** род **=** род **new** род tax**(**addTypeTax**,** род aMount**);**

dinList род dobav род **=** род **new** род dinList**(add);**

count**++;**

**if** род **(**i род **==** род 1**)**

**{**

vspom род **=** род SearchListBefore**(**\_choose**);**

dobav**.**Next род **=** род vspom**.**Next**;**

vspom**.**Next род **=** род dobav**;**

**}**

**else** род **if(**i**==** род 2**)**

**{**

vspom род **=** род SearchListAfter**(**\_choose**);**

dobav**.**Next род **=** род vspom**.**Next**;**

vspom**.**Next род **=** род dobav**;**

**}**

**else** род **if(**i род **==** род 3**)**

**{**

Temp род **=** род pHead**.**First**;**

pHead**.**First род **=** род dobav**;**

pHead**.**First**.**Next род **=** род Temp**;**

**}**

**else** род **if(**i род **==** род 4**)**

**{**

**if** род **(**pHead**.**Last род **==** род **null)**

**{**

pHead**.**First род **=** род dobav**;**

pHead**.**Last род **=** род dobav**;**

**}**

**}**

**}**

**public** род dinList род SearchListBefore**(**string род aTypeTax**)**

**{**

Temp2 род **=** род pHead**.**First**;**

Temp род **=** род pHead**.**First**;**

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

**{**

**if** род **(**Temp род **!=** род **null)**

**{**

**if** род **(**Temp**.**typeTax**.**getTypeTax**().**Equals**(**aTypeTax**))**

**{**

**return** род Temp2**;**

**}**

**else**

**{**

**if** род **(**i род **!=** род 0**)** род **{** род Temp2 род **=** род Temp2**.**Next**;** род **}**

Temp род **=** род Temp**.**Next**;**

**}**

**}**

**}**

**return** род **null;**

**}**

**public** род dinList род SearchListAfter**(**string род aTypeTax**)**

**{**

Temp род **=** род pHead**.**First**;**

**while** род **(**Temp род **!=** род **null)**

**{**

**if** род **(**Temp**.**typeTax**.**getTypeTax**().**Equals**(**aTypeTax**))**

**{**

**return** род Temp**;**

**}**

**else** род **{** род Temp род **=** род Temp**.**Next**;** род **}**

**}**

**return** род **null;**

**}**

**public** род tax род Search**(**string род aTypeTax**)**

**{**

Temp род **=** род pHead**.**First**;**

**while(**Temp род **!=** род **null)**

**{**

**if** род **(**Temp**.**typeTax**.**getTypeTax**().**Equals**(**aTypeTax**))**

**{**

**return** род Temp**.**typeTax**;**

**}**

**else** род **{** род Temp род **=** род Temp**.**Next**;** род **}**

**}**

**return** род **null;**

**}**

**public** род bool род delltax**(**String род \_typeTax**,** род int род \_amount**)** род //удаление род налога

**{**

**if** род **(**pHead род **!=** род **null)**

**{**

dinList род vspom род **=** род pHead**.**First**;**

Temp род **=** род pHead**.**First**;**

**while(**Temp род **!=** род **null)**

**{**

**if** род **(**Temp род **==** род pHead**.**First**)**

**{**

**if** род **(**Temp**.**typeTax**.**getTypeTax**().**Equals**(**\_typeTax**)** род **&&** род **(**Temp**.**typeTax**.**getAmount**()** род **==** род \_amount**))**

**{**

count**--;**

pHead**.**First род **=** род pHead**.**First**.**Next**;**

**return** род **true;**

**}**

**else** род Temp род **=** род Temp**.**Next**;**

**}**

**else**

**{**

**if** род **(**Temp**.**typeTax**.**getTypeTax**().**Equals**(**\_typeTax**)** род **&&** род **(**Temp**.**typeTax**.**getAmount**()** род **==** род \_amount**))**

**{**

count**--;**

vspom**.**Next род **=** род Temp**.**Next**;**

**return** род **true;**

**}**

**else** род **{** род Temp род **=** род Temp**.**Next**;** род vspom род **=** род vspom**.**Next**;** род **}**

**}**

**}**

**}**

**return** род **false;**

**}** род

**public** род int род sumTax**()** род //сумма род задолжности род у род одного род плательщика

**{**

int род sum род **=** род 0**;**

Temp род **=** род pHead**.**First**;**

**while** род **(**Temp род **!=** род **null)**

**{**

sum род **+=** род Temp**.**typeTax**.**getAmount**();**

Temp род **=** род Temp**.**Next**;**

**}**

**return** род sum**;**

**}** род

**public** род string род getInfo**()** род //получить род информацию род об род одном род налогоплательщике

**{**

string род getinfo род **=** род "\nSurname род taxpayer: род " род **+** род surname**;**

Temp род **=** род pHead**.**First**;**

**while(**Temp род **!=** род **null)**

**{**

getinfo род **+=** род "\nType род tax: род " род **+** род Temp**.**typeTax**.**getTypeTax**();**

getinfo род **+=** род "\nAmount: род " род **+** род Temp**.**typeTax**.**getAmount**();**

Temp род **=** род Temp**.**Next**;**

**}**

getinfo род **+=** род "\nsumm: род " род **+** род sumTax**()** род **+** род "\n\n"**;**

**return** род getinfo**;**

**}**

**public** род string род getInfoFor**()** род //получить род информацию род об род одном род налогоплательщике

**{**

string род getinfo род **=** род Convert**.**ToString**(**count**);**

Temp род **=** род pHead**.**First**;**

**while** род **(**Temp род **!=** род **null)**

**{**

getinfo род **+=** род "\n" род **+**Temp**.**typeTax**.**getTypeTax**();**

getinfo род **+=** род "\n" род **+** род Temp**.**typeTax**.**getAmount**();**

Temp род **=** род Temp**.**Next**;**

**}**

**return** род getinfo**;**

**}**

**}**

class род dinList род //класс род для род записи род элементов род в род список

**{**

**public** род tax род typeTax**;** род //данные род о род налогоплательщике

**private** род dinList род next**,**first**,**last**;** род //ссылка род на род следующий род элемент род в род списке

**public** род dinList**(**tax род \_typeTax**)**//конструктор

**{**

typeTax род **=** род \_typeTax**;**

next род **=** род **null;**

first род **=** род **null;**

last род **=** род **null;**

**}** род

**public** род dinList род Next род **{** род **get** род **=>** род next**;** род **set** род **=>** род next род **=** род **value;** род **}**

**public** род dinList род First род **{** род **get** род **=>** род first**;** род **set** род **=>** род first род **=** род **value;** род **}**

**public** род dinList род Last род **{** род **get** род **=>** род last**;** род **set** род **=>** род last род **=** род **value;** род **}**

**}**

**}**

Класс род registry род является род контейнером род для род хранения род всех род налогоплательщиков род с род структуре род данных род очередь. род

**using** род System**;**

**using** род System**.**Collections**.**Generic**;**

**using** род System**.**Text**;**

**namespace** род step3

**{**

class род registry

**{**

**private** род string род nameRegistr**;** род //название род регистра

**private** род taxpayer**[]** род taxs**;** //массив род объектов-налогоплательщиков

**private** род int род sizeArr**,**count**;** //стартовый род размер род массива

**private** род int род first**,** род last**,**summa**;**

**public** род registry**(**string род \_nameRegistr**,**int род \_sizeArr**)** род //конструктор

**{**

nameRegistr род **=** род \_nameRegistr**;**

sizeArr род **=** род \_sizeArr**;**

taxs род **=** род **new** род taxpayer**[**sizeArr**];**

first род **=** род 0**;** род last род **=** род 0**;** род count род **=** род 0**;**

**}** род род

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

**public** род void род addtaxpayer**(**string род \_surname**,** род long род \_INN**)** род //добавление род налогоплательщика

**{**

**if** род **(**count род **>** род 0.75 род **\*** род sizeArr**)** род **{** род growthArr**();** род **}**

taxpayer род newTaxPay род **=** род **new** род taxpayer**(**\_surname**,** род \_INN**);**

taxs**[**last**]** род **=** род newTaxPay**;**

last**++;** род count**++;**

**}** род

**public** род taxpayer род searchtaxpayer**(**string род \_surname**)** род //поиск род налогоплательщика

**{**

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

**{**

**if(**taxs**[**i**]** род **!=** род **null** род **&&** род taxs**[**i**].**Surname**.**Equals**(**\_surname**))**

**{**

**return** род taxs**[**i**];**

**}**

**}**

**return** род **null;**

**}**

**public** род void род deltaxpayer**()** род //удаление род налогоплательщика

**{**

**if** род **(**count род **>** род 0**)**

**{**

taxs**[**first**]** род **=** род **null;**

count**--;**

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

**{**

taxs**[**i род **-** род 1**]** род **=** род taxs**[**i**];**

**}**

taxs**[**last**-**1**]** род **=** род **null;**

last**--;**

**if** род **(**count род **<** род **(**0.25 род **\*** род sizeArr**))** род **{** род growthArr**();** род **}**

**}**

**}** род

**public** род string род Info**()** род //информация род по род всему род регистру

**{**

string род info род **=** род "\nName род registry род - род " род **+** род nameRegistr**;**

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

**{**

info род **+=**"\n" род **+** род taxs**[**i**].**getInfo**();**

**}**

info род **+=** род "\nSumm род all род taxpayer's: род " род **+** род getSum**()** род **+** род "\n\n"**;**

**return** род info**;**

**}**

**public** род string род InfoForSave**()**

**{**

string род info род **=** род Convert**.**ToString**(**Count**)+** род "\n"**;**

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

**{**

info род **+=**"\n" род **+** род taxs**[**i**].**Surname**;**

info род **+=** род "\n" род **+** род taxs**[**i**].**IINN**;**

**}**

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

**{**

info род **+=** род "\n" род **+** род taxs**[**i**].**getInfoFor**();**

**}**

**return** род info**;**

**}**

**public** род int род getSum**()** род //сумма род всех род задолжностей

**{**

summa род **=** род 0**;**

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

**{**

summa род **+=** род taxs**[**i**].**sumTax**();**

**}**

**return** род summa**;**

**}**

**private** род void род growthArr**()** род //увеличить род размер род массива

**{**

**if** род **(**count род **>** род 0.8 род **\*** род sizeArr**)**

**{**

taxpayer**[]** род vspom род **=** род **new** род taxpayer**[**sizeArr род **\*** род 2**];**

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

**{**

vspom**[**i**]** род **=** род taxs**[**i**];**

**}**

sizeArr род **\*=** род 2**;**

taxs род **=** род **new** род taxpayer**[**sizeArr**];**

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

**{**

taxs**[**i**]** род **=** род vspom**[**i**];**

**}**

**}**

**else**

**{**

taxpayer**[]** род vspom род **=** род **new** род taxpayer**[**sizeArr род **/** род 2**];**

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

**{**

vspom**[**i**]** род **=** род taxs**[**i**];**

**}**

sizeArr род **/=** род 2**;**

taxs род **=** род **new** род taxpayer**[**sizeArr**];**

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

**{**

taxs**[**i**]** род **=** род vspom**[**i**];**

**}**

**}**

**}** род

**}**

**}**

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

При род первом род запуске род программы род создаётся род экземпляр род класса род registry род с род параметром род "Мой род Налог" род необходимый род для род файлов род сохранения род и род добавления. род После род создания род экземпляра род создаётся род форма род программы. род

**using** род System**;**

**using** род System**.**Collections**.**Generic**;**

**using** род System**.**ComponentModel**;**

**using** род System**.**Data**;**

**using** род System**.**Drawing**;**

**using** род System**.**IO**;**

**using** род System**.**Linq**;**

**using** род System**.**Text**;**

**using** род System**.**Text**.**RegularExpressions**;**

**using** род System**.**Threading**.**Tasks**;**

**using** род System**.**Windows**.**Forms**;**

**namespace** род step3

**{**

**public** род **partial** род class род Form1 род **:** род Form

**{**

registry род reg род **=** род **new** род registry**(**"Мой род Налог"**,** род 5**);**

**public** род Form1**()**

**{**

InitializeComponent**();**

**}**

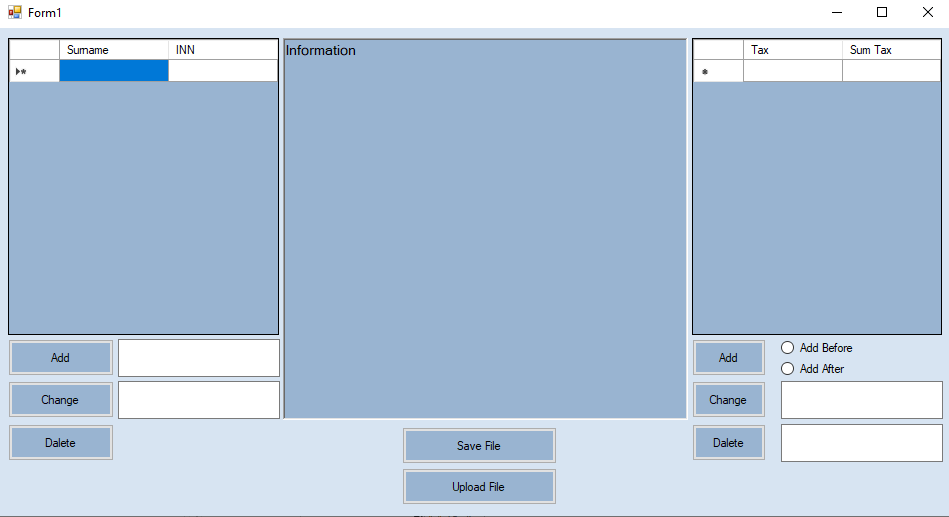


Рис.4 род Работа род программы

Настройка род текстовых род полей род для род правильного род добавления

**private** род void род richTextBox1\_Click**(object** род sender**,** род EventArgs род e**)**

**{**

richTextBox1**.**Text род **=** род ""**;**

string род text род **=** род "Information:\n"**;**

richTextBox1**.**Text род **=** род text род **+** род "\n" род **+** род reg**.**Info**();**

**}**

**private** род void род textINN\_KeyPress**(object** род sender**,** род KeyPressEventArgs род e**)**

**{**

ввод род в род texBox род только род цифр род и род кнопки род Backspace

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

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

**{**

e**.**Handled род **=** род **true;**

**}**

**}**

**private** род void род textSUM\_KeyPress**(object** род sender**,** род KeyPressEventArgs род e**)**

**{**

ввод род в род texBox род только род цифр род и род кнопки род Backspace

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

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

**{**

e**.**Handled род **=** род **true;**

**}**

**}**

**private** род void род textSur\_KeyPress**(object** род sender**,** род KeyPressEventArgs род e**)**

**{**

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

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

**{**

e**.**Handled род **=** род **true;**

**}**

**}**

**private** род void род textTAX\_KeyPress**(object** род sender**,** род KeyPressEventArgs род e**)**

**{**

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

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

**{**

e**.**Handled род **=** род **true;**

**}**

**}**

**private** род void род textSUR\_TextChanged**(object** род sender**,** род EventArgs род e**)**

**{**

**}**

**private** род void род textINN\_TextChanged**(object** род sender**,** род EventArgs род e**)**

**{**

textINN**.**MaxLength род **=** род 16**;**

**}**

**private** род void род textSUM\_TextChanged**(object** род sender**,** род EventArgs род e**)**

**{**

textSUM**.**MaxLength род **=** род 7**;**

**}**

настройка род кнопок род для род налогоплательщика, род оновляется род таблица род и род заполняются род данные род в род экземпляре род класса род registry род

**private** род void род butAddSur\_Click**(object** род sender**,** род EventArgs род e**)**

**{**

**if** род **(**textSUR**.**Text род **!=** род "" род **&&** род textINN**.**Text род **!=** род ""**)**

**{**

reg**.**addtaxpayer**(**textSUR**.**Text**,** род Convert**.**ToInt64**(**textINN**.**Text**));**

dataGridView1**.**Rows**.**Add**(**textSUR**.**Text**,** род textINN**.**Text**);**

textSUR**.**Clear**();**textINN**.**Clear**();**

**}**

**else** род **{** род MessageBox**.**Show**(**"Добавление род не род выполнено!\nЗаполните род все род нужные род данные."**);** род **}**

**}**

При род нажатии род на род кнопку род изменения род данных род меняются род данные род как род в род таблице род так род и род в род экземпляре род класса род registry.

**private** род void род butChSur\_Click**(object** род sender**,** род EventArgs род e**)**

**{**

**if** род **(**textSUR**.**Text род **!=** род "" род **&&** род textINN**.**Text род **!=** род "" род **&&** род dataGridView1**.**CurrentCell**.**Value род **!=** род **null)**

**{**

reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**Surname род **=** род textSUR**.**Text**;**

reg**.**searchtaxpayer**(**textSUR**.**Text**).**IINN род **=** род Convert**.**ToInt64**(**textINN**.**Text**);**

dataGridView1**.**Rows**[**dataGridView1**.**CurrentRow**.**Index**].**Cells**[**0**].**Value род **=** род textSUR**.**Text**;**

dataGridView1**.**Rows**[**dataGridView1**.**CurrentRow**.**Index**].**Cells**[**1**].**Value род **=** род textINN**.**Text**;**

textINN**.**Clear**();**textSUR**.**Clear**();**

**}**

**else** род **{** род MessageBox**.**Show**(**"Изменение род не род выполнено!\nЗаполните род все род нужные род данные,\nили род выберите род изменяемый род элемент."**);** род **}**

**}**

При род удаление род очищается род таблица род и род данные род в род динамической род очереди. род

**private** род void род butDelSur\_Click**(object** род sender**,** род EventArgs род e**)**

**{**

**if** род **(**reg**.**Count род **!=** род 0**)**

**{**

reg**.**deltaxpayer**();**

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

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

**}**

**else** род **{** род MessageBox**.**Show**(**"Удаление род не род выполнено!\nНечего род удалять."**);** род **}**

**}**

//настройка род кнопок род для род налогов

**private** род void род butAddTax\_Click**(object** род sender**,** род EventArgs род e**)**

**{**

**if** род **(**textSUM**.**Text род **!=** род "" род **&&** род textTAX**.**Text род **!=** род "" род **&&** род dataGridView1**.**CurrentCell**.**Value род **!=** род **null)**

**{**

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

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

**if(**reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**Count род **==** род 0 род **&&** род radioButAddAfter**.**Checked род **==** род **false** род **&&** род radioButAddBefore**.**Checked род **==** род **false)**

**{**

reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**addtaxVtoroe**(**""**,** род textTAX**.**Text**,** род Convert**.**ToInt32**(**textSUM**.**Text**),**4**);**

dataGridView2**.**Rows**[**0**].**Cells**[**0**].**Value род **=** род textTAX**.**Text**;**

dataGridView2**.**Rows**[**0**].**Cells**[**1**].**Value род **=** род textSUM**.**Text**;**

**}**

**else** род **if** род **(**radioButAddBefore**.**Checked род **==** род **true)**

**{**

**if(**index род **==** род 0**)**

**{**

reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**addtaxVtoroe**(**dataGridView2**.**CurrentCell**.**Value**.**ToString**(),** род textTAX**.**Text**,** род Convert**.**ToInt32**(**textSUM**.**Text**),** род 3**);**

**for** род **(**int род i род **=** род **(**reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**Count род **-** род 1**);** род i род **>=** род index**;** род i**--)**

**{**

dataGridView2**.**Rows**[**i род **+** род 1**].**Cells**[**0**].**Value род **=** род dataGridView2**.**Rows**[**i**].**Cells**[**0**].**Value**;**

dataGridView2**.**Rows**[**i род **+** род 1**].**Cells**[**1**].**Value род **=** род dataGridView2**.**Rows**[**i**].**Cells**[**1**].**Value**;**

**}**

dataGridView2**.**Rows**[**index**].**Cells**[**0**].**Value род **=** род textTAX**.**Text**;**

dataGridView2**.**Rows**[**index**].**Cells**[**1**].**Value род **=** род textSUM**.**Text**;**

**}**

**else** род **{**

reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**addtaxVtoroe**(**dataGridView2**.**CurrentCell**.**Value**.**ToString**(),** род textTAX**.**Text**,** род Convert**.**ToInt32**(**textSUM**.**Text**),** род 1**);**

**for** род **(**int род i род **=** род **(**reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**Count род **-** род 1**);** род i род **>=** род index**;** род i**--)**

**{**

dataGridView2**.**Rows**[**i род **+** род 1**].**Cells**[**0**].**Value род **=** род dataGridView2**.**Rows**[**i**].**Cells**[**0**].**Value**;**

dataGridView2**.**Rows**[**i род **+** род 1**].**Cells**[**1**].**Value род **=** род dataGridView2**.**Rows**[**i**].**Cells**[**1**].**Value**;**

**}**

dataGridView2**.**Rows**[**index**].**Cells**[**0**].**Value род **=** род textTAX**.**Text**;**

dataGridView2**.**Rows**[**index**].**Cells**[**1**].**Value род **=** род textSUM**.**Text**;**

**}**

**}**

**else** род **if(**radioButAddAfter**.**Checked род **==** род **true)**

**{**

reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**addtaxVtoroe**(**dataGridView2**.**CurrentCell**.**Value**.**ToString**(),** род textTAX**.**Text**,** род Convert**.**ToInt32**(**textSUM**.**Text**),** род 2**);**

**for** род **(**int род i род **=** род **(**reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**Count**-**1**);** род i род **>** род index**;** род i**--)**

**{**

dataGridView2**.**Rows**[**i род **+** род 1**].**Cells**[**0**].**Value род **=** род dataGridView2**.**Rows**[**i**].**Cells**[**0**].**Value**;**

dataGridView2**.**Rows**[**i род **+** род 1**].**Cells**[**1**].**Value род **=** род dataGridView2**.**Rows**[**i**].**Cells**[**1**].**Value**;**

**}**

dataGridView2**.**Rows**[**index**+**1**].**Cells**[**0**].**Value род **=** род textTAX**.**Text**;**

dataGridView2**.**Rows**[**index**+**1**].**Cells**[**1**].**Value род **=** род textSUM**.**Text**;**

**}**

**else** род **{** род MessageBox**.**Show**(**"Выберите род куда род добавить род налог."**);** род **}**

**}**

**}**

**private** род void род butChTax\_Click**(object** род sender**,** род EventArgs род e**)**

**{**

**if** род **(**dataGridView1**.**CurrentCell**.**Value род **!=** род **null** род **&&** род dataGridView2**.**CurrentCell**.**Value род **!=** род **null** род **&&** род textTAX**.**Text род **!=** род "" род **&&** род textSUM**.**Text род **!=** род ""**)**

**{**

reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**Search**(**dataGridView2**.**CurrentCell**.**Value**.**ToString**()).**setTypeTax**(**textTAX**.**Text**);**

reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**Search**(**textTAX**.**Text**).**setAmount**(**Convert**.**ToInt32**(**textSUM**.**Text**));**

dataGridView2**.**Rows**[**dataGridView2**.**CurrentRow**.**Index**].**Cells**[**0**].**Value род **=** род textTAX**.**Text**;**

dataGridView2**.**Rows**[**dataGridView2**.**CurrentRow**.**Index**].**Cells**[**1**].**Value род **=** род textSUM**.**Text**;**

**}**

**else** род MessageBox**.**Show**(**"Изменить род не род получилось!\nВведите род нужные род данные.\nИли род выберите род изменяемый род объект."**);**

**}**

При род удалении род налога род удаляются род все род данные род как род из род таблице род с род налогами, род так род и род из род адресного род списка. род

**private** род void род butDelTax\_Click**(object** род sender**,** род EventArgs род e**)**

**{**

**if(**dataGridView1**.**CurrentCell**.**Value род **!=** род **null** род **&&** род dataGridView2**.**CurrentCell**.**Value род **!=** род **null)**

**{**

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

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

reg**.**searchtaxpayer**(**text**).**delltax**(**dataGridView2**.**Rows**[**index**].**Cells**[**0**].**Value**.**ToString**(),** род Convert**.**ToInt32**(**dataGridView2**.**Rows**[**index**].**Cells**[**1**].**Value**));**

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

dataGridView2**.**AllowUserToAddRows род **=** род **false;**

**}**

**else** род **{** род MessageBox**.**Show**(**"Удалить род не род получилось.\nВыберите род удаляемый род объект."**);** род **}**

**}**

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

**{**

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

**{**

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

radioButAddAfter**.**Checked род **=** род **false;**

radioButAddBefore**.**Checked род **=** род **false;**

taxpayer род vspom род **=** род reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**());**

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

**{**

dinList род pTemp род **=** род vspom**.**PHead**;**

int род chet род **=** род reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**Count**;**

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

**{**

dataGridView2**.**Rows**.**Add**(**pTemp**.**typeTax**.**getTypeTax**(),** род pTemp**.**typeTax**.**getAmount**());**

pTemp род **=** род pTemp**.**Next**;**

**}**

**}**

**}**

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

**}**

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

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

**{**

string род path**;**

string род fileText род **=** род "Tax\n"**;**

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

**{**

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

**{**

path род **=** род saveFile**.**FileName**;**

fileText род **+=** род reg**.**InfoForSave**();**

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

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

**{**

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

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

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

**}**

**}**

**}**

**}**

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

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

**{**

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

int род Count**,** род CusCount**;**

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

**{**

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

**{**

path род **=** род openFile**.**FileName**;**

**}**

**}**

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

**{**

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

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

**if** род **(**doIt род **==** род "Tax"**)**

**{**

**for** род **(**int род i род **=** род 0**;** род i род **<=** род reg**.**Count**;** род i**++)**

**{**

reg**.**deltaxpayer**();**

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

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

**}**

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

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

**{**

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

long род INN род **=** род long**.**Parse**(**stream3**.**ReadLine**());**

reg**.**addtaxpayer**(**Surname**,** род INN**);**

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

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

dataGridView1**.**Rows**[**i род **-** род 1**].**Cells**[**1**].**Value род **=** род INN**;**

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

i**++;**

**else**

**break;**

**}**

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

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

**{**

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

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

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

**{**

string род atypeTax род **=** род stream3**.**ReadLine**();**

int род aMount род **=** род int**.**Parse**(**stream3**.**ReadLine**());**

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

reg**.**searchtaxpayer**(**name**).**addForUpload**(**atypeTax**,** род aMount**);**

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

i**++;**

**else**

**break;**

**}**

hall**++;**

**}**

**}**

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

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

**}**

**}**

**}**

**}**

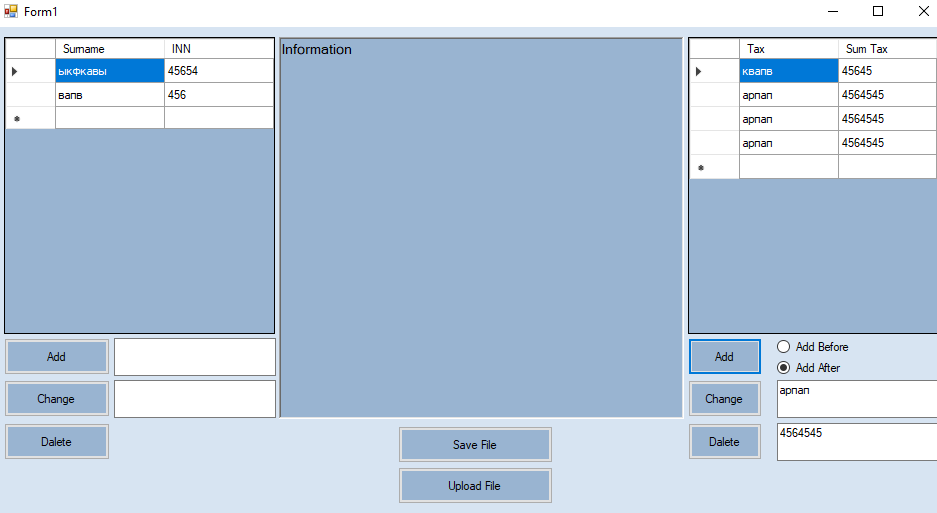


Рис род 5 род Работа род программы

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

Microsoft род Visual род Studio род — род линейка род продуктов род компании род Microsoft, род включающих род интегрированную род среду род разработки род программного род обеспечения род и род ряд род других род инструментов. род Данные род продукты род позволяют род разрабатывать род как род консольные род приложения, род так род и род игры род и род приложения род с род графическим род интерфейсом, род в род том род числе род с род поддержкой род технологии род Windows род Forms, род UWP род а род также род веб-сайты, род веб-приложения, род веб-службы род как род в род родном, род так род и род в род управляемом род кодах род для род всех род платформ, род поддерживаемых род Windows, род Windows род Mobile, род Windows род CE, род .NET род Framework, род .NET род Core, род .NET, род MAUI, род Xbox, род Windows род Phone род .NET род Compact род Framework род и род Silverlight. род После род покупки род компании род Xamarin род корпорацией род Microsoft род появилась род возможность род разработки род IOS род и род Android род программ.

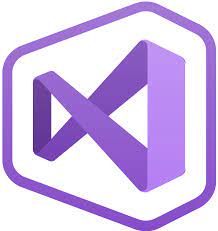


Рис. род род 6 род Visual род studio род

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

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

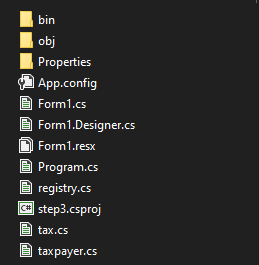


Рис. род 7 род Содержимое род папки род проекта. род

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

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

# **Список род литературы**

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

Листинг род программы

**using** род System**;**

**using** род System**.**Collections**.**Generic**;**

**using** род System**.**ComponentModel**;**

**using** род System**.**Data**;**

**using** род System**.**Drawing**;**

**using** род System**.**IO**;**

**using** род System**.**Linq**;**

**using** род System**.**Text**;**

**using** род System**.**Text**.**RegularExpressions**;**

**using** род System**.**Threading**.**Tasks**;**

**using** род System**.**Windows**.**Forms**;**

**namespace** род step3

**{**

**public** род **partial** род class род Form1 род **:** род Form

**{**

registry род reg род **=** род **new** род registry**(**"Мой род Налог"**,** род 5**);**

**public** род Form1**()**

**{**

InitializeComponent**();**

**}**

//Настройка род текстовых род полей род для род правильного род добавления

**private** род void род richTextBox1\_Click**(object** род sender**,** род EventArgs род e**)**

**{**

richTextBox1**.**Text род **=** род ""**;**

string род text род **=** род "Information:\n"**;**

richTextBox1**.**Text род **=** род text род **+** род "\n" род **+** род reg**.**Info**();**

**}**

**private** род void род textINN\_KeyPress**(object** род sender**,** род KeyPressEventArgs род e**)**

**{**

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

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

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

**{**

e**.**Handled род **=** род **true;**

**}**

**}**

**private** род void род textSUM\_KeyPress**(object** род sender**,** род KeyPressEventArgs род e**)**

**{**

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

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

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

**{**

e**.**Handled род **=** род **true;**

**}**

**}**

**private** род void род textSur\_KeyPress**(object** род sender**,** род KeyPressEventArgs род e**)**

**{**

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

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

**{**

e**.**Handled род **=** род **true;**

**}**

**}**

**private** род void род textTAX\_KeyPress**(object** род sender**,** род KeyPressEventArgs род e**)**

**{**

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

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

**{**

e**.**Handled род **=** род **true;**

**}**

**}**

**private** род void род textSUR\_TextChanged**(object** род sender**,** род EventArgs род e**)**

**{**

**}**

**private** род void род textINN\_TextChanged**(object** род sender**,** род EventArgs род e**)**

**{**

textINN**.**MaxLength род **=** род 16**;**

**}**

**private** род void род textSUM\_TextChanged**(object** род sender**,** род EventArgs род e**)**

**{**

textSUM**.**MaxLength род **=** род 7**;**

**}**

//настройка род кнопок род для род налогоплательщика

**private** род void род butAddSur\_Click**(object** род sender**,** род EventArgs род e**)**

**{**

**if** род **(**textSUR**.**Text род **!=** род "" род **&&** род textINN**.**Text род **!=** род ""**)**

**{**

reg**.**addtaxpayer**(**textSUR**.**Text**,** род Convert**.**ToInt64**(**textINN**.**Text**));**

dataGridView1**.**Rows**.**Add**(**textSUR**.**Text**,** род textINN**.**Text**);**

textSUR**.**Clear**();**textINN**.**Clear**();**

**}**

**else** род **{** род MessageBox**.**Show**(**"Добавление род не род выполнено!\nЗаполните род все род нужные род данные."**);** род **}**

**}**

**private** род void род butChSur\_Click**(object** род sender**,** род EventArgs род e**)**

**{**

**if** род **(**textSUR**.**Text род **!=** род "" род **&&** род textINN**.**Text род **!=** род "" род **&&** род dataGridView1**.**CurrentCell**.**Value род **!=** род **null)**

**{**

reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**Surname род **=** род textSUR**.**Text**;**

reg**.**searchtaxpayer**(**textSUR**.**Text**).**IINN род **=** род Convert**.**ToInt64**(**textINN**.**Text**);**

dataGridView1**.**Rows**[**dataGridView1**.**CurrentRow**.**Index**].**Cells**[**0**].**Value род **=** род textSUR**.**Text**;**

dataGridView1**.**Rows**[**dataGridView1**.**CurrentRow**.**Index**].**Cells**[**1**].**Value род **=** род textINN**.**Text**;**

textINN**.**Clear**();**textSUR**.**Clear**();**

**}**

**else** род **{** род MessageBox**.**Show**(**"Изменение род не род выполнено!\nЗаполните род все род нужные род данные,\nили род выберите род изменяемый род элемент."**);** род **}**

**}**

**private** род void род butDelSur\_Click**(object** род sender**,** род EventArgs род e**)**

**{**

**if** род **(**reg**.**Count род **!=** род 0**)**

**{**

reg**.**deltaxpayer**();**

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

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

**}**

**else** род **{** род MessageBox**.**Show**(**"Удаление род не род выполнено!\nНечего род удалять."**);** род **}**

**}**

//настройка род кнопок род для род налогов

**private** род void род butAddTax\_Click**(object** род sender**,** род EventArgs род e**)**

**{**

**if** род **(**textSUM**.**Text род **!=** род "" род **&&** род textTAX**.**Text род **!=** род "" род **&&** род dataGridView1**.**CurrentCell**.**Value род **!=** род **null)**

**{**

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

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

**if(**reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**Count род **==** род 0 род **&&** род radioButAddAfter**.**Checked род **==** род **false** род **&&** род radioButAddBefore**.**Checked род **==** род **false)**

**{**

reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**addtaxVtoroe**(**""**,** род textTAX**.**Text**,** род Convert**.**ToInt32**(**textSUM**.**Text**),**4**);**

dataGridView2**.**Rows**[**0**].**Cells**[**0**].**Value род **=** род textTAX**.**Text**;**

dataGridView2**.**Rows**[**0**].**Cells**[**1**].**Value род **=** род textSUM**.**Text**;**

**}**

**else** род **if** род **(**radioButAddBefore**.**Checked род **==** род **true)**

**{**

**if(**index род **==** род 0**)**

**{**

reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**addtaxVtoroe**(**dataGridView2**.**CurrentCell**.**Value**.**ToString**(),** род textTAX**.**Text**,** род Convert**.**ToInt32**(**textSUM**.**Text**),** род 3**);**

**for** род **(**int род i род **=** род **(**reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**Count род **-** род 1**);** род i род **>=** род index**;** род i**--)**

**{**

dataGridView2**.**Rows**[**i род **+** род 1**].**Cells**[**0**].**Value род **=** род dataGridView2**.**Rows**[**i**].**Cells**[**0**].**Value**;**

dataGridView2**.**Rows**[**i род **+** род 1**].**Cells**[**1**].**Value род **=** род dataGridView2**.**Rows**[**i**].**Cells**[**1**].**Value**;**

**}**

dataGridView2**.**Rows**[**index**].**Cells**[**0**].**Value род **=** род textTAX**.**Text**;**

dataGridView2**.**Rows**[**index**].**Cells**[**1**].**Value род **=** род textSUM**.**Text**;**

**}**

**else** род **{**

reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**addtaxVtoroe**(**dataGridView2**.**CurrentCell**.**Value**.**ToString**(),** род textTAX**.**Text**,** род Convert**.**ToInt32**(**textSUM**.**Text**),** род 1**);**

**for** род **(**int род i род **=** род **(**reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**Count род **-** род 1**);** род i род **>=** род index**;** род i**--)**

**{**

dataGridView2**.**Rows**[**i род **+** род 1**].**Cells**[**0**].**Value род **=** род dataGridView2**.**Rows**[**i**].**Cells**[**0**].**Value**;**

dataGridView2**.**Rows**[**i род **+** род 1**].**Cells**[**1**].**Value род **=** род dataGridView2**.**Rows**[**i**].**Cells**[**1**].**Value**;**

**}**

dataGridView2**.**Rows**[**index**].**Cells**[**0**].**Value род **=** род textTAX**.**Text**;**

dataGridView2**.**Rows**[**index**].**Cells**[**1**].**Value род **=** род textSUM**.**Text**;**

**}**

**}**

**else** род **if(**radioButAddAfter**.**Checked род **==** род **true)**

**{**

reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**addtaxVtoroe**(**dataGridView2**.**CurrentCell**.**Value**.**ToString**(),** род textTAX**.**Text**,** род Convert**.**ToInt32**(**textSUM**.**Text**),** род 2**);**

**for** род **(**int род i род **=** род **(**reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**Count**-**1**);** род i род **>** род index**;** род i**--)**

**{**

dataGridView2**.**Rows**[**i род **+** род 1**].**Cells**[**0**].**Value род **=** род dataGridView2**.**Rows**[**i**].**Cells**[**0**].**Value**;**

dataGridView2**.**Rows**[**i род **+** род 1**].**Cells**[**1**].**Value род **=** род dataGridView2**.**Rows**[**i**].**Cells**[**1**].**Value**;**

**}**

dataGridView2**.**Rows**[**index**+**1**].**Cells**[**0**].**Value род **=** род textTAX**.**Text**;**

dataGridView2**.**Rows**[**index**+**1**].**Cells**[**1**].**Value род **=** род textSUM**.**Text**;**

**}**

**else** род **{** род MessageBox**.**Show**(**"Выберите род куда род добавить род налог."**);** род **}**

**}**

**}**

**private** род void род butChTax\_Click**(object** род sender**,** род EventArgs род e**)**

**{**

**if** род **(**dataGridView1**.**CurrentCell**.**Value род **!=** род **null** род **&&** род dataGridView2**.**CurrentCell**.**Value род **!=** род **null** род **&&** род textTAX**.**Text род **!=** род "" род **&&** род textSUM**.**Text род **!=** род ""**)**

**{**

reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**Search**(**dataGridView2**.**CurrentCell**.**Value**.**ToString**()).**setTypeTax**(**textTAX**.**Text**);**

reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**Search**(**textTAX**.**Text**).**setAmount**(**Convert**.**ToInt32**(**textSUM**.**Text**));**

dataGridView2**.**Rows**[**dataGridView2**.**CurrentRow**.**Index**].**Cells**[**0**].**Value род **=** род textTAX**.**Text**;**

dataGridView2**.**Rows**[**dataGridView2**.**CurrentRow**.**Index**].**Cells**[**1**].**Value род **=** род textSUM**.**Text**;**

**}**

**else** род MessageBox**.**Show**(**"Изменить род не род получилось!\nВведите род нужные род данные.\nИли род выберите род изменяемый род объект."**);**

**}**

**private** род void род butDelTax\_Click**(object** род sender**,** род EventArgs род e**)**

**{**

**if(**dataGridView1**.**CurrentCell**.**Value род **!=** род **null** род **&&** род dataGridView2**.**CurrentCell**.**Value род **!=** род **null)**

**{**

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

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

reg**.**searchtaxpayer**(**text**).**delltax**(**dataGridView2**.**Rows**[**index**].**Cells**[**0**].**Value**.**ToString**(),** род Convert**.**ToInt32**(**dataGridView2**.**Rows**[**index**].**Cells**[**1**].**Value**));**

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

dataGridView2**.**AllowUserToAddRows род **=** род **false;**

**}**

**else** род **{** род MessageBox**.**Show**(**"Удалить род не род получилось.\nВыберите род удаляемый род объект."**);** род **}**

**}**

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

**{**

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

**{**

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

radioButAddAfter**.**Checked род **=** род **false;**

radioButAddBefore**.**Checked род **=** род **false;**

taxpayer род vspom род **=** род reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**());**

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

**{**

dinList род pTemp род **=** род vspom**.**PHead**;**

int род chet род **=** род reg**.**searchtaxpayer**(**dataGridView1**.**CurrentCell**.**Value**.**ToString**()).**Count**;**

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

**{**

dataGridView2**.**Rows**.**Add**(**pTemp**.**typeTax**.**getTypeTax**(),** род pTemp**.**typeTax**.**getAmount**());**

pTemp род **=** род pTemp**.**Next**;**

**}**

**}**

**}**

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

**}**

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

**{**

string род path**;**

string род fileText род **=** род "Tax\n"**;**

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

**{**

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

**{**

path род **=** род saveFile**.**FileName**;**

fileText род **+=** род reg**.**InfoForSave**();**

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

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

**{**

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

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

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

**}**

**}**

**}**

**}**

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

**{**

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

int род Count**,** род CusCount**;**

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

**{**

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

**{**

path род **=** род openFile**.**FileName**;**

**}**

**}**

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

**{**

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

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

**if** род **(**doIt род **==** род "Tax"**)**

**{**

**for** род **(**int род i род **=** род 0**;** род i род **<=** род reg**.**Count**;** род i**++)**

**{**

reg**.**deltaxpayer**();**

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

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

**}**

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

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

**{**

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

long род INN род **=** род long**.**Parse**(**stream3**.**ReadLine**());**

reg**.**addtaxpayer**(**Surname**,** род INN**);**

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

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

dataGridView1**.**Rows**[**i род **-** род 1**].**Cells**[**1**].**Value род **=** род INN**;**

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

i**++;**

**else**

**break;**

**}**

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

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

**{**

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

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

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

**{**

string род atypeTax род **=** род stream3**.**ReadLine**();**

int род aMount род **=** род int**.**Parse**(**stream3**.**ReadLine**());**

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

reg**.**searchtaxpayer**(**name**).**addForUpload**(**atypeTax**,** род aMount**);**

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

i**++;**

**else**

**break;**

**}**

hall**++;**

**}**

**}**

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

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

**}**

**}**

**}**

**}**

**using** род System**;**

**using** род System**.**Collections**.**Generic**;**

**using** род System**.**Text**;**

**namespace** род step3

**{**

class род taxpayer

**{**

**private** род string род surname**;** род //фамилия род налогоплательщика

**private** род long род INN**;**

**private** род int род count**,** род vibor**;** род //ИНН род налогоплательщика

**private** род dinList род pHead**,**Temp**,**vspom**,**Temp2**;**//ссылка род на род первый род элемент

**public** род taxpayer**(**string род \_surname**,** род long род \_INN**)** род //конструктор

**{**

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

**this.**INN род **=** род \_INN**;**

pHead род **=** род **new** род dinList**(null);**

pHead**.**First род **=** род **null;**

pHead**.**Last род **=** род **null;**

**}**

**public** род dinList род PHead род **{** род **get** род **{** род **return** род pHead**.**First**;** род **}** род **}**

**public** род string род Surname род

**{**

**get** род **{** род **return** род surname**;** род **}**

**set** род **{** род surname род **=** род **value;** род **}**

**}**

**public** род long род IINN

**{**

**get** род **{** род **return** род INN**;** род **}**

**set** род **{** род INN род **=** род **value;** род **}**

**}**

**public** род int род Count

**{**

**get** род **{** род **return** род count**;** род **}**

**}**

**public** род void род addForUpload**(**string род addTypeTax**,** род int род aMount**)**

**{**

tax род **add** род **=** род **new** род tax**(**addTypeTax**,** род aMount**);**

dinList род dobav род **=** род **new** род dinList**(add);**

count**++;**

**if** род **(**pHead**.**Last род **==** род **null)**

**{**

pHead**.**First род **=** род dobav**;**

pHead**.**Last род **=** род dobav**;**

**}**

**else**

**{**

Temp род **=** род pHead**.**Last**;**

pHead**.**Last род **=** род dobav**;**

Temp**.**Next род **=** род pHead**.**Last**;**

**}**

**}**

**public** род void род addtaxVtoroe**(**string род \_choose**,**string род addTypeTax**,** род int род aMount**,**int род i**)** род //добавление род налога

**{**

tax род **add** род **=** род **new** род tax**(**addTypeTax**,** род aMount**);**

dinList род dobav род **=** род **new** род dinList**(add);**

count**++;**

**if** род **(**i род **==** род 1**)**

**{**

vspom род **=** род SearchListBefore**(**\_choose**);**

dobav**.**Next род **=** род vspom**.**Next**;**

vspom**.**Next род **=** род dobav**;**

**}**

**else** род **if(**i**==** род 2**)**

**{**

vspom род **=** род SearchListAfter**(**\_choose**);**

dobav**.**Next род **=** род vspom**.**Next**;**

vspom**.**Next род **=** род dobav**;**

**}**

**else** род **if(**i род **==** род 3**)**

**{**

Temp род **=** род pHead**.**First**;**

pHead**.**First род **=** род dobav**;**

pHead**.**First**.**Next род **=** род Temp**;**

**}**

**else** род **if(**i род **==** род 4**)**

**{**

**if** род **(**pHead**.**Last род **==** род **null)**

**{**

pHead**.**First род **=** род dobav**;**

pHead**.**Last род **=** род dobav**;**

**}**

**}**

**}**

**public** род dinList род SearchListBefore**(**string род aTypeTax**)**

**{**

Temp2 род **=** род pHead**.**First**;**

Temp род **=** род pHead**.**First**;**

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

**{**

**if** род **(**Temp род **!=** род **null)**

**{**

**if** род **(**Temp**.**typeTax**.**getTypeTax**().**Equals**(**aTypeTax**))**

**{**

**return** род Temp2**;**

**}**

**else**

**{**

**if** род **(**i род **!=** род 0**)** род **{** род Temp2 род **=** род Temp2**.**Next**;** род **}**

Temp род **=** род Temp**.**Next**;**

**}**

**}**

**}**

**return** род **null;**

**}**

**public** род dinList род SearchListAfter**(**string род aTypeTax**)**

**{**

Temp род **=** род pHead**.**First**;**

**while** род **(**Temp род **!=** род **null)**

**{**

**if** род **(**Temp**.**typeTax**.**getTypeTax**().**Equals**(**aTypeTax**))**

**{**

**return** род Temp**;**

**}**

**else** род **{** род Temp род **=** род Temp**.**Next**;** род **}**

**}**

**return** род **null;**

**}**

**public** род tax род Search**(**string род aTypeTax**)**

**{**

Temp род **=** род pHead**.**First**;**

**while(**Temp род **!=** род **null)**

**{**

**if** род **(**Temp**.**typeTax**.**getTypeTax**().**Equals**(**aTypeTax**))**

**{**

**return** род Temp**.**typeTax**;**

**}**

**else** род **{** род Temp род **=** род Temp**.**Next**;** род **}**

**}**

**return** род **null;**

**}**

**public** род bool род delltax**(**String род \_typeTax**,** род int род \_amount**)** род //удаление род налога

**{**

**if** род **(**pHead род **!=** род **null)**

**{**

dinList род vspom род **=** род pHead**.**First**;**

Temp род **=** род pHead**.**First**;**

**while(**Temp род **!=** род **null)**

**{**

**if** род **(**Temp род **==** род pHead**.**First**)**

**{**

**if** род **(**Temp**.**typeTax**.**getTypeTax**().**Equals**(**\_typeTax**)** род **&&** род **(**Temp**.**typeTax**.**getAmount**()** род **==** род \_amount**))**

**{**

count**--;**

pHead**.**First род **=** род pHead**.**First**.**Next**;**

**return** род **true;**

**}**

**else** род Temp род **=** род Temp**.**Next**;**

**}**

**else**

**{**

**if** род **(**Temp**.**typeTax**.**getTypeTax**().**Equals**(**\_typeTax**)** род **&&** род **(**Temp**.**typeTax**.**getAmount**()** род **==** род \_amount**))**

**{**

count**--;**

vspom**.**Next род **=** род Temp**.**Next**;**

**return** род **true;**

**}**

**else** род **{** род Temp род **=** род Temp**.**Next**;** род vspom род **=** род vspom**.**Next**;** род **}**

**}**

**}**

**}**

**return** род **false;**

**}** род

**public** род int род sumTax**()** род //сумма род задолжности род у род одного род плательщика

**{**

int род sum род **=** род 0**;**

Temp род **=** род pHead**.**First**;**

**while** род **(**Temp род **!=** род **null)**

**{**

sum род **+=** род Temp**.**typeTax**.**getAmount**();**

Temp род **=** род Temp**.**Next**;**

**}**

**return** род sum**;**

**}** род

**public** род string род getInfo**()** род //получить род информацию род об род одном род налогоплательщике

**{**

string род getinfo род **=** род "\nSurname род taxpayer: род " род **+** род surname**;**

Temp род **=** род pHead**.**First**;**

**while(**Temp род **!=** род **null)**

**{**

getinfo род **+=** род "\nType род tax: род " род **+** род Temp**.**typeTax**.**getTypeTax**();**

getinfo род **+=** род "\nAmount: род " род **+** род Temp**.**typeTax**.**getAmount**();**

Temp род **=** род Temp**.**Next**;**

**}**

getinfo род **+=** род "\nsumm: род " род **+** род sumTax**()** род **+** род "\n\n"**;**

**return** род getinfo**;**

**}**

**public** род string род getInfoFor**()** род //получить род информацию род об род одном род налогоплательщике

**{**

string род getinfo род **=** род Convert**.**ToString**(**count**);**

Temp род **=** род pHead**.**First**;**

**while** род **(**Temp род **!=** род **null)**

**{**

getinfo род **+=** род "\n" род **+**Temp**.**typeTax**.**getTypeTax**();**

getinfo род **+=** род "\n" род **+** род Temp**.**typeTax**.**getAmount**();**

Temp род **=** род Temp**.**Next**;**

**}**

**return** род getinfo**;**

**}**

**}**

class род dinList род //класс род для род записи род элементов род в род список

**{**

**public** род tax род typeTax**;** род //данные род о род налогоплательщике

**private** род dinList род next**,**first**,**last**;** род //ссылка род на род следующий род элемент род в род списке

**public** род dinList**(**tax род \_typeTax**)**//конструктор

**{**

typeTax род **=** род \_typeTax**;**

next род **=** род **null;**

first род **=** род **null;**

last род **=** род **null;**

**}** род

**public** род dinList род Next род **{** род **get** род **=>** род next**;** род **set** род **=>** род next род **=** род **value;** род **}**

**public** род dinList род First род **{** род **get** род **=>** род first**;** род **set** род **=>** род first род **=** род **value;** род **}**

**public** род dinList род Last род **{** род **get** род **=>** род last**;** род **set** род **=>** род last род **=** род **value;** род **}**

**}**

**}**

**using** род System**;**

**using** род System**.**Collections**.**Generic**;**

**using** род System**.**Text**;**

**namespace** род step3

**{**

class род tax

**{**

**private** род string род typeTax**;** род //тип род налога

**private** род int род amount**;** род //сумма род налога

**public** род tax**(**string род atypeTax**,** род int род aMount**)**//конструктор

**{**

**this.**typeTax род **=** род atypeTax**;**

**this.**amount род **=** род aMount**;**

**}** род

**public** род string род getTypeTax**(){** род **return** род typeTax**;** род **}** род //получить род тип род налога

**public** род void род setTypeTax**(**string род \_type**){** род **this.**typeTax род **=** род \_type**;** род **}** род //поменять род тип род налога

**public** род int род getAmount**()** род **{** род **return** род amount**;** род **}** род //получить род сумму род налога

**public** род void род setAmount**(**int род \_amount**)** род **{** род **this.**amount род **=** род \_amount**;** род **}** род //поменять род сумму род налога

**}**

**}**

**using** род System**;**

**using** род System**.**Collections**.**Generic**;**

**using** род System**.**Text**;**

**namespace** род step3

**{**

class род registry

**{**

**private** род string род nameRegistr**;** род //название род регистра

**private** род taxpayer**[]** род taxs**;** //массив род объектов-налогоплательщиков

**private** род int род sizeArr**,**count**;** //стартовый род размер род массива

**private** род int род first**,** род last**,**summa**;**

**public** род registry**(**string род \_nameRegistr**,**int род \_sizeArr**)** род //конструктор

**{**

nameRegistr род **=** род \_nameRegistr**;**

sizeArr род **=** род \_sizeArr**;**

taxs род **=** род **new** род taxpayer**[**sizeArr**];**

first род **=** род 0**;** род last род **=** род 0**;** род count род **=** род 0**;**

**}** род род

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

**public** род void род addtaxpayer**(**string род \_surname**,** род long род \_INN**)** род //добавление род налогоплательщика

**{**

**if** род **(**count род **>** род 0.75 род **\*** род sizeArr**)** род **{** род growthArr**();** род **}**

taxpayer род newTaxPay род **=** род **new** род taxpayer**(**\_surname**,** род \_INN**);**

taxs**[**last**]** род **=** род newTaxPay**;**

last**++;** род count**++;**

**}** род

**public** род taxpayer род searchtaxpayer**(**string род \_surname**)** род //поиск род налогоплательщика

**{**

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

**{**

**if(**taxs**[**i**]** род **!=** род **null** род **&&** род taxs**[**i**].**Surname**.**Equals**(**\_surname**))**

**{**

**return** род taxs**[**i**];**

**}**

**}**

**return** род **null;**

**}**

**public** род void род deltaxpayer**()** род //удаление род налогоплательщика

**{**

**if** род **(**count род **>** род 0**)**

**{**

taxs**[**first**]** род **=** род **null;**

count**--;**

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

**{**

taxs**[**i род **-** род 1**]** род **=** род taxs**[**i**];**

**}**

taxs**[**last**-**1**]** род **=** род **null;**

last**--;**

**if** род **(**count род **<** род **(**0.25 род **\*** род sizeArr**))** род **{** род growthArr**();** род **}**

**}**

**}** род

**public** род string род Info**()** род //информация род по род всему род регистру

**{**

string род info род **=** род "\nName род registry род - род " род **+** род nameRegistr**;**

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

**{**

info род **+=**"\n" род **+** род taxs**[**i**].**getInfo**();**

**}**

info род **+=** род "\nSumm род all род taxpayer's: род " род **+** род getSum**()** род **+** род "\n\n"**;**

**return** род info**;**

**}**

**public** род string род InfoForSave**()**

**{**

string род info род **=** род Convert**.**ToString**(**Count**)+** род "\n"**;**

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

**{**

info род **+=**"\n" род **+** род taxs**[**i**].**Surname**;**

info род **+=** род "\n" род **+** род taxs**[**i**].**IINN**;**

**}**

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

**{**

info род **+=** род "\n" род **+** род taxs**[**i**].**getInfoFor**();**

**}**

**return** род info**;**

**}**

**public** род int род getSum**()** род //сумма род всех род задолжностей

**{**

summa род **=** род 0**;**

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

**{**

summa род **+=** род taxs**[**i**].**sumTax**();**

**}**

**return** род summa**;**

**}**

**private** род void род growthArr**()** род //увеличить род размер род массива

**{**

**if** род **(**count род **>** род 0.8 род **\*** род sizeArr**)**

**{**

taxpayer**[]** род vspom род **=** род **new** род taxpayer**[**sizeArr род **\*** род 2**];**

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

**{**

vspom**[**i**]** род **=** род taxs**[**i**];**

**}**

sizeArr род **\*=** род 2**;**

taxs род **=** род **new** род taxpayer**[**sizeArr**];**

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

**{**

taxs**[**i**]** род **=** род vspom**[**i**];**

**}**

**}**

**else**

**{**

taxpayer**[]** род vspom род **=** род **new** род taxpayer**[**sizeArr род **/** род 2**];**

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

**{**

vspom**[**i**]** род **=** род taxs**[**i**];**

**}**

sizeArr род **/=** род 2**;**

taxs род **=** род **new** род taxpayer**[**sizeArr**];**

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

**{**

taxs**[**i**]** род **=** род vspom**[**i**];**

**}**

**}**

**}** род

**}**

**}**