Федеральное государственное бюджетное

образовательное учреждение высшего образования

«Саратовский государственный технический университет

имени Гагарина Ю.А.»

Институт прикладных информационных

технологий и коммуникаций

Кафедра Информационная безопасность

автоматизированных систем

Направление подготовки 10.03.01 Информационная безопасность

Расчётно-графическая работа

по дисциплине «Языки программирования»

**«Телефонный справочник»**

Выполнил: студент 1 курса

Учебной группы с-ИБС12

Очной формы обучения

Шаповалов Д. В.

Проверил: ассистент каф. ИБС

Романчук С. П.

Саратов 2020

# Аннотация

**Краткое описание работы**

**Телефонный справочник:** приложение, эмулирующее функционал телефонного справочника.

**Возможности программы:**

1. добавление, редактирование и удаление контактов, или очищение всего списка,
2. просмотр всех контактов и поиск по различным характеристикам,
3. загрузка и сохранение телефонного справочника в файл.

Реализован класс моделирующий запись в телефонной книге.

Оглавление

[**Аннотация** 1](#_Toc42656440)

[**Введение** 3](#_Toc42656441)

[Цель работы 3](#_Toc42656442)

[Задачи 3](#_Toc42656443)

[Описание приложения 4](#_Toc42656444)

[**Теоретическая часть** 5](#_Toc42656445)

[C++ 5](#_Toc42656446)

[**Практическая часть** 7](#_Toc42656447)

[Библиотеки 7](#_Toc42656448)

[Классы 7](#_Toc42656449)

[Методы TelephoneDirectory.cpp 9](#_Toc42656450)

[**Заключение** 10](#_Toc42656451)

[**Код программы** 11](#_Toc42656452)

[**Литература** 17](#_Toc42656453)

# Введение

**Тема**

«Телефонный справочник»

# ****Цель работы****

Разработать Телефонный справочник, используя полученные в ходе обучения знания о языке C++, в частности, об объектно-ориентированном программировании.

# ****Задачи****

1. Реализация взаимодействие с контактами телефонного справочника.
2. Реализация методов взаимодействия с файлами непосредственно (сохранение/ открытие/ редактирование/ удаление справочника).
3. Доработка дизайна приложения для потенциального пользователя.
4. Коррекция кода с последующим тестовым запуском программы.

# Описание приложения

Приложение представляет собой текстовое меню с нумерованным списком действий.

Взаимодействие с программой осуществляется путём нажатия соответствующих символов на клавиатуре.

Пользователю предлагается широкий функционал взаимодействия с телефонным справочником (далее - ТС), включающее в себя:

1. Возможность создания и переименования ТС (одновременно возможно взаимодействие только с одним ТС);
2. Возможность добавления контактов к уже существующему ТС;
3. Возможность поиска конкретного контакта по имени/ фамилии/ номеру телефона;
4. Возможность сохранения созданного ТС в файл;
5. Возможность открытия созданного ТС из файла (программа взаимодействует только с файлами, находящимися в папке собственно программы).

# Теоретическая часть

C++ — компилируемый, статически типизированный язык программирования общего назначения.

Поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование. Язык имеет богатую стандартную библиотеку, которая включает в себя распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности. C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков. В сравнении с его предшественником — языком  
C, — наибольшее внимание уделено поддержке объектно-ориентированного и обобщённого программирования.

Стандарт C++ состоит из двух основных частей: описание ядра языка и описание стандартной библиотеки.

Первое время язык развивался вне формальных рамок, спонтанно, по мере встававших перед ним задач. Развитию языка сопутствовало развитие кросс-компилятора [cfront](https://ru.wikipedia.org/w/index.php?title=Cfront&action=edit&redlink=1). Новшества в языке отражались в изменении номера версии кросс-компилятора. Эти номера версий кросс-компилятора распространялись и на сам язык, но применительно к настоящему времени речь о версиях языка C++ не ведут. Лишь в 1998 году язык стал стандартизированным.

C++ поддерживает как комментарии в стиле C (/\* комментарий \*/), так и однострочные (// вся оставшаяся часть строки является комментарием), где // обозначает начало комментария, а ближайший последующий символ новой строки, который не предварён символом \ (либо эквивалентным ему обозначением ??/), считается окончанием комментария. Плюс этого комментария в том, что его не обязательно заканчивать, то есть обозначать окончание комментария.

Спецификатор inline для функций. Функция, определённая внутри тела класса, является inline по умолчанию. Данный спецификатор является подсказкой компилятору и может встроить тело функции в код вместо её непосредственного вызова.

Квалификаторы const и volatile. В отличие от С, где const обозначает только доступ на чтение, в C++ переменная с квалификатором const должна быть инициализирована.

В языке программирования C++ термин **Стандартная Библиотека** означает коллекцию классов функций, написанных на базовом языке. Стандартная Библиотека поддерживает несколько основных контейнеров, функций для работы с этими контейнерами, объектов-функции, основных типов строк и потоков (включая интерактивный и файловый ввод-вывод), поддержку некоторых языковых особенностей, и часто используемые функции для выполнения таких задач, как, например, нахождение квадратного корня числа. Стандартная Библиотека языка C++ также включает в себя спецификации стандарта ISO C90 стандартной библиотеки языка Си. Функциональные особенности Стандартной Библиотеки объявляются внутри пространства имен std. Пространства имён (namespace).

Для работы с памятью введены операторы new, new [ ], delete и delete [ ]. В отличие от библиотечных malloc и free, пришедших из C, данные операторы производят инициализацию объекта. Для классов это вызов конструктора, для POD типов инициализацию можно либо не проводить (new Pod;), либо провести инициализацию нулевыми значениями (new Pod ( ); new Pod{ };).

# Практическая часть

Проект «RGR2» (название проекта ТС) состоит из пяти файлов, включающих в себя 2 независимых друг от друга класса. Первый класс состоит из 7-ми функций, второй – из 12-ти.

# Библиотеки

**Стандартная библиотека языка программирования C++**

*Используемые классы библиотеки:*

**iostream –** (от англ. input/output stream **–** поток ввода-вывода) – заголовочный файл с классами, функциями и переменными для организации ввода-вывода в языке программирования C++. Он включён в стандартную библиотеку C++.

**conio.h** – (от англ. console input-output – консольный ввод-вывод) – заголовочный файл, используемый в старых компиляторах, работающих в операционных системах MS-DOS, для создания текстового интерфейса пользователя.

**fstream** – (от англ. filestream – файловый поток) – заголовочный файл из стандартной библиотеки C++, включающий набор классов, методов и функций, которые предоставляют интерфейс для чтения/записи данных из/в файл. Для манипуляции с данными файлов используются объекты, называемые потоками («stream»).

**string** – (от англ. string – строка) –класс с методами и переменными для организации работы со строками в языке программирования C++.

# Классы

**Account** – (англ. account – учётная запись) – содержит в себе всю информацию, связанную с учётными записями.

Состав класса:

1. *Name* – поле типа string, хранящее имя учётной записи;
2. *Surname* – поле типа string хранящее фамилию учётной записи;
3. *Number* – поле типа string хранящее номер учётной записи;
4. *Account ()* – конструктор по умолчанию (создаёт пустую учётную запись);
5. «*Гетеры*» и «*сеттеры*» вышеперечисленных полей.
6. *Print* – метод, реализующий вывод имеющихся данных об учётной записи на экран.

**TelephoneDirectory** – (англ. telephone directory – телефонный справочник) – содержит в себе всю информацию, связанную со справочником как таковым.

Состав класса:

1. *Name* – поле типа string, хранящее имя ТС;
2. *Size* – поле типа int, хранящее количество учётных записей;
3. *TD* – поле типа данных *Account\**, хранящее информацию об учётной записи в ТС.
4. *TelephoneDirectory ()* – конструктор по умолчанию (создаёт пустую ТС);
5. *TelephoneDirectory (string Name, int Size)* – конструктор с параметром (используется для создания и заполнения учётных записей);
6. *~TelephoneDirectory* – деструктор (уничтожает открытую в программе ТС).
7. *print, insert, edit, del, clearing, search, fout, fin* – методы, описывающие выполнение программы в целом (более подробно далее).

# Методы TelephoneDirectory.cpp

1. *print* – осуществляет вывод текущего состояния ТС на экран;
2. *insert* – осуществляет добавление новой учётной записи в ТС.

Выполняется в следующем порядке: создаётся временный pTD, копирующий размер TD, производится очищение памяти TD, поле Size TD увеличивается на 1, выделяется память под TD, на размер одной учётной записи больше предыдущего, содержимое из временного pTD копируется в TD, временный pTD удаляется.

1. *edit* – осуществляет изменение существующей учётной записи в ТС.

Выполняется в следующем порядке: с помощью ввода объекта *n* типа int с клавиатуры определяется номер изменяемой учётной записи, объявляется *pS* типа string, служащий временным хранилищем для вводимых символов, выполняются соответствующие set-методы (setName, setSurname, setNumber), актуальная информация выводится на экран.

1. *del* – осуществляет удаление учётной записи из ТС.

Выполняется в следующем порядке: Size TD уменьшается на единицу, объявляется временный pTD типа *Account\**, копирующий размер TD, производится очищение памяти TD, содержимое из временного pTD копируется в TD, временный pTD удаляется.

1. *clearing* – осуществляет удаление открытого ТС.

Выполняется в следующем порядке: посредством ввода с клавиатуры подтверждается удаление существующего ТС, в случае подтверждения память TD очищается, выделяется память под TD, равная «пустому» TD, поле Size обнуляется.

1. *search* – осуществляет поиск аккаунта по имени/ фамилии/ номеру телефона.

Выполняется в следующем порядке: объявляется *pS* типа string, служащий временным хранилищем для вводимых символов, объявляется временный pTD типа *Account\**, посредством ввода с клавиатуры происходит выбор требуемого параметра поиска (по имени/ по фамилии/ по номеру телефона), пользователь вводит необходимое значение, происходит проверка выбранного параметра на совпадение с имеющимися в учётных записях, все подошедшие учётные записи выводятся на экран.

1. *fout* – осуществляет сохранение ТС в файл с текущим названием ТС.

Выполняется в следующем порядке: объявляется *path* типа string, служащий временным хранилищем для имени файла, к имени дописывается “.txt”, что делает файл пригодным для сохранения в текстовом формате, проверяет текущий ТС на наличие учётных записей, в случае их отсутствия выводит соответствующее сообщение на экран, производится форматированная запись в файл.

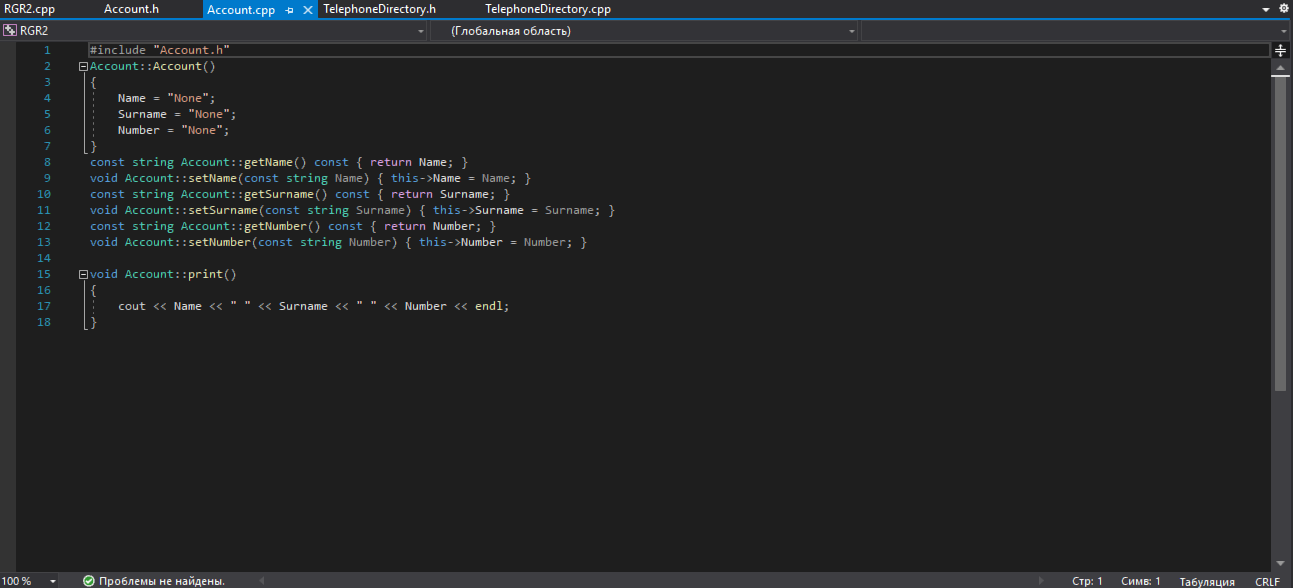
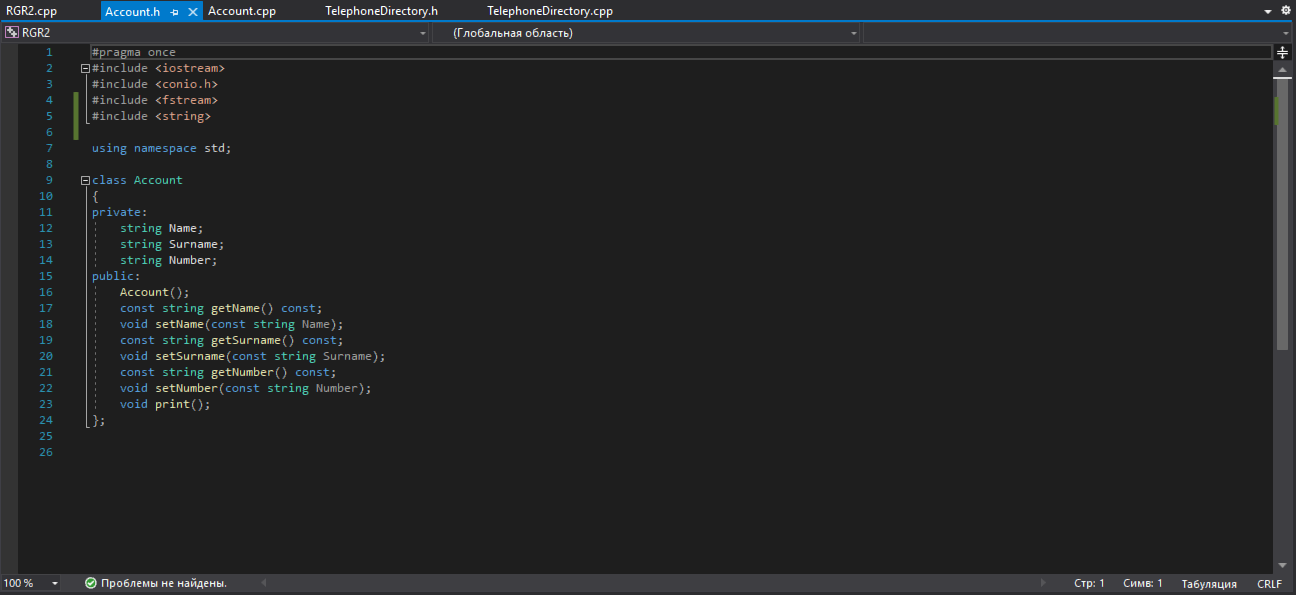
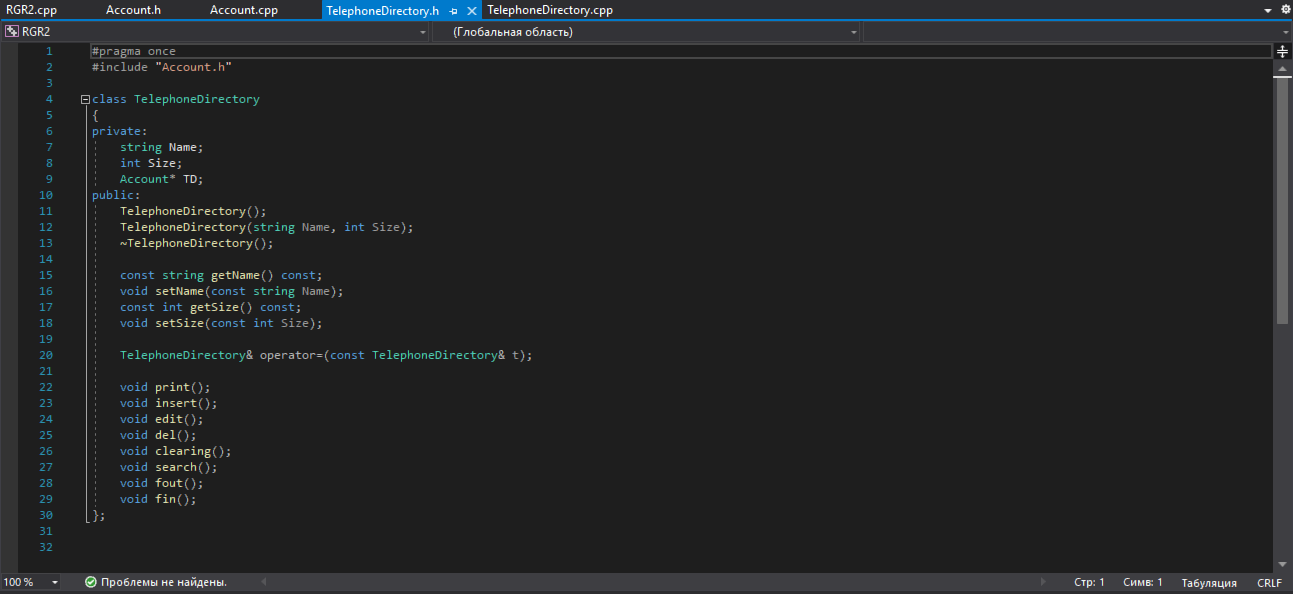
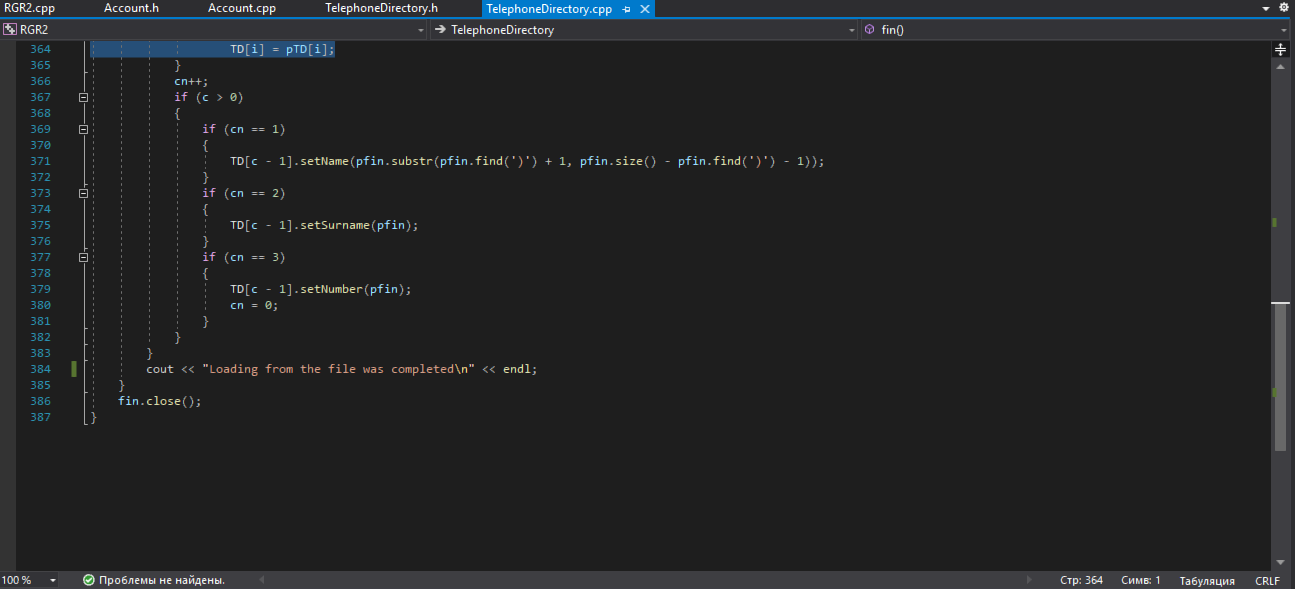
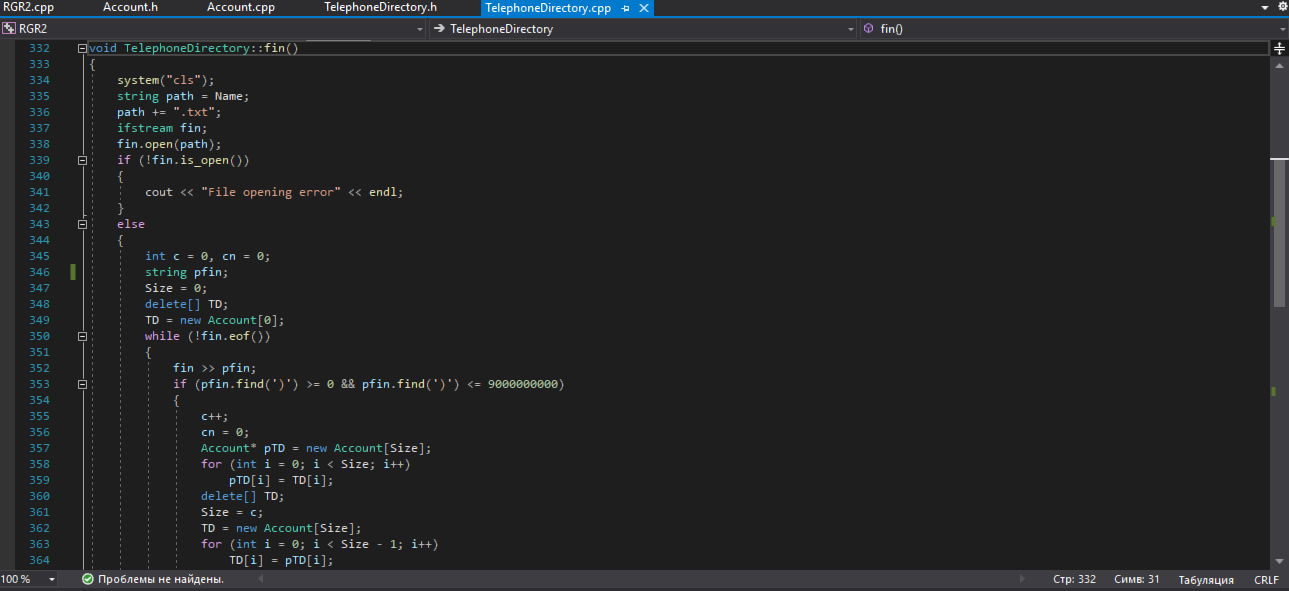
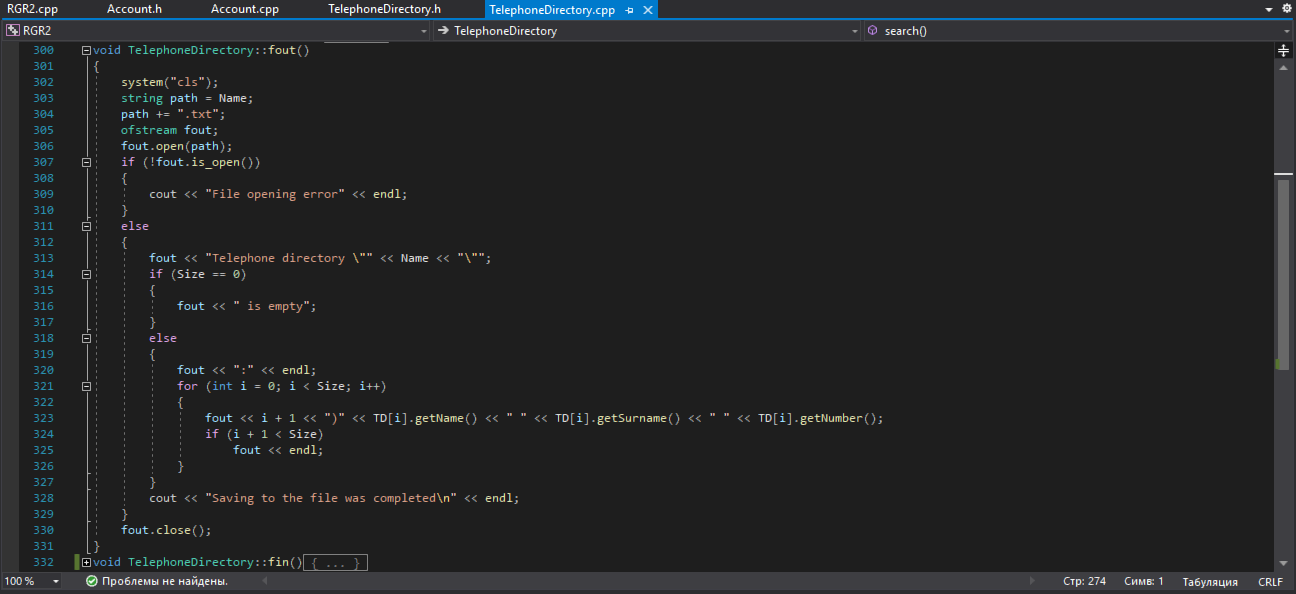
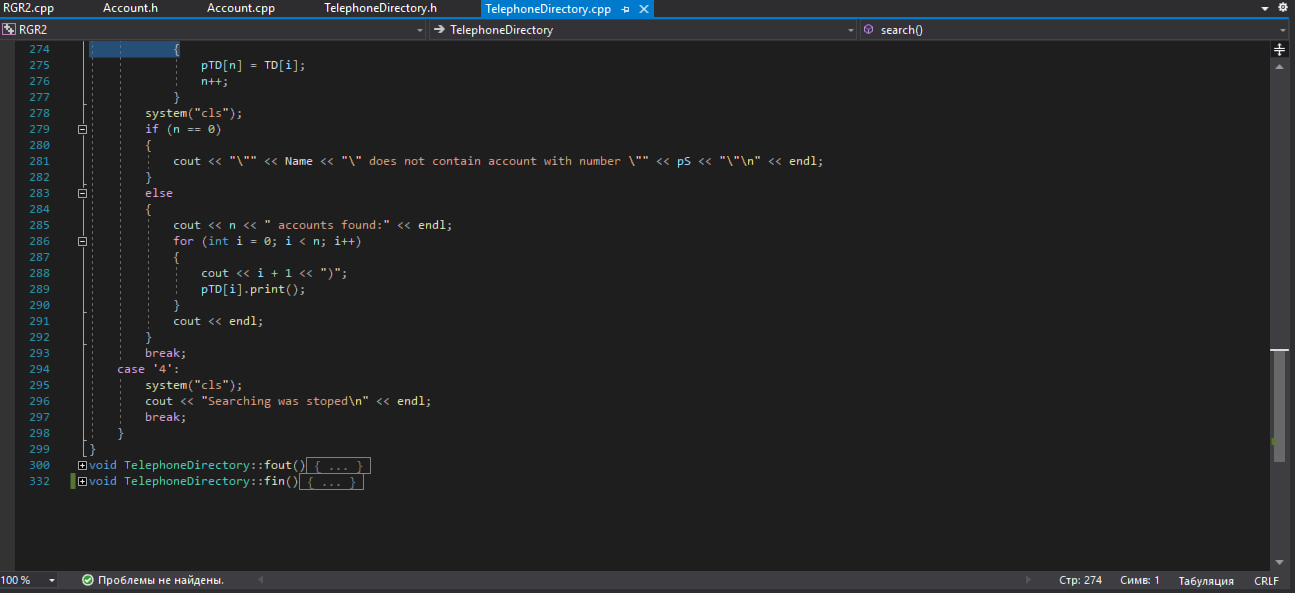
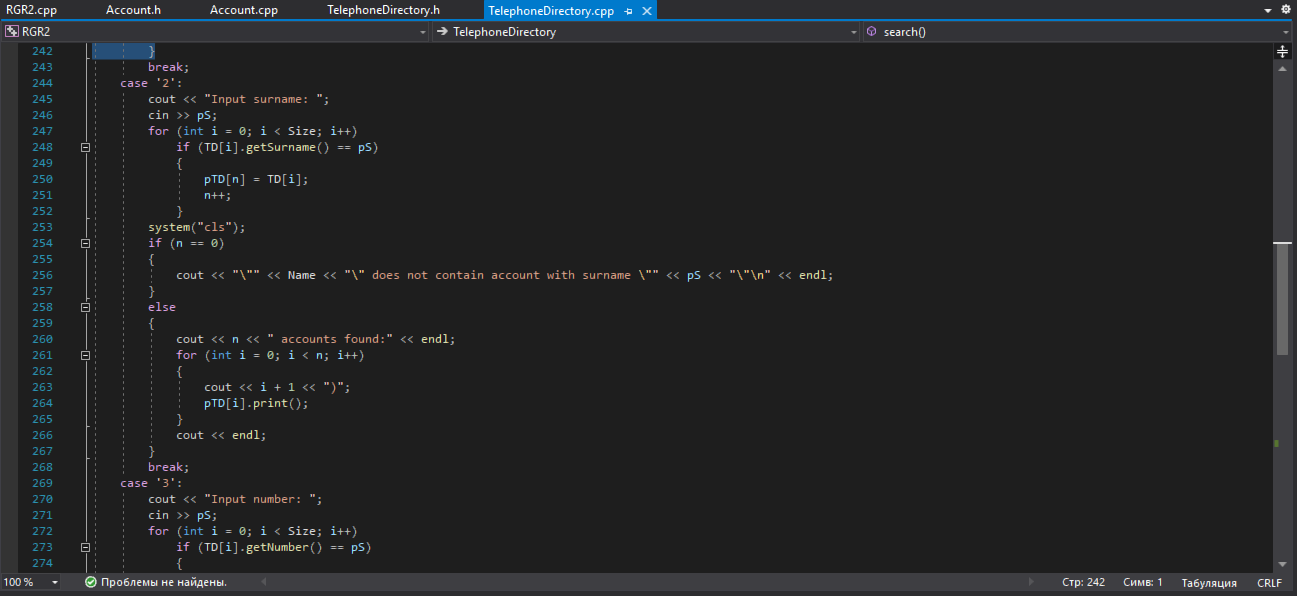
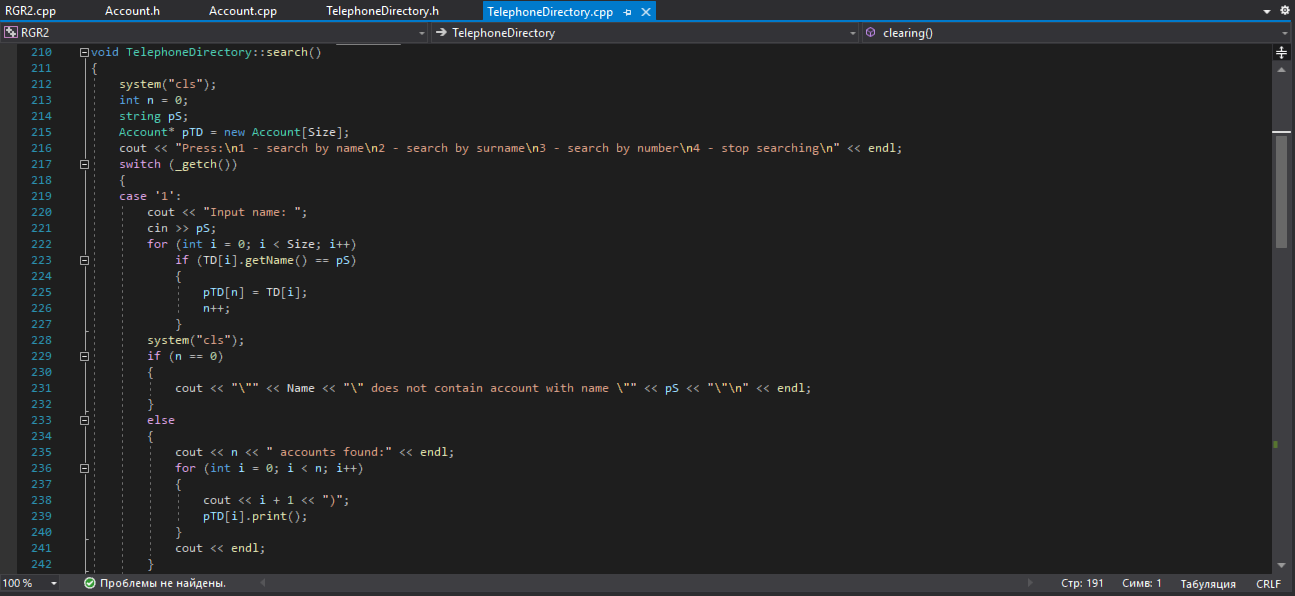
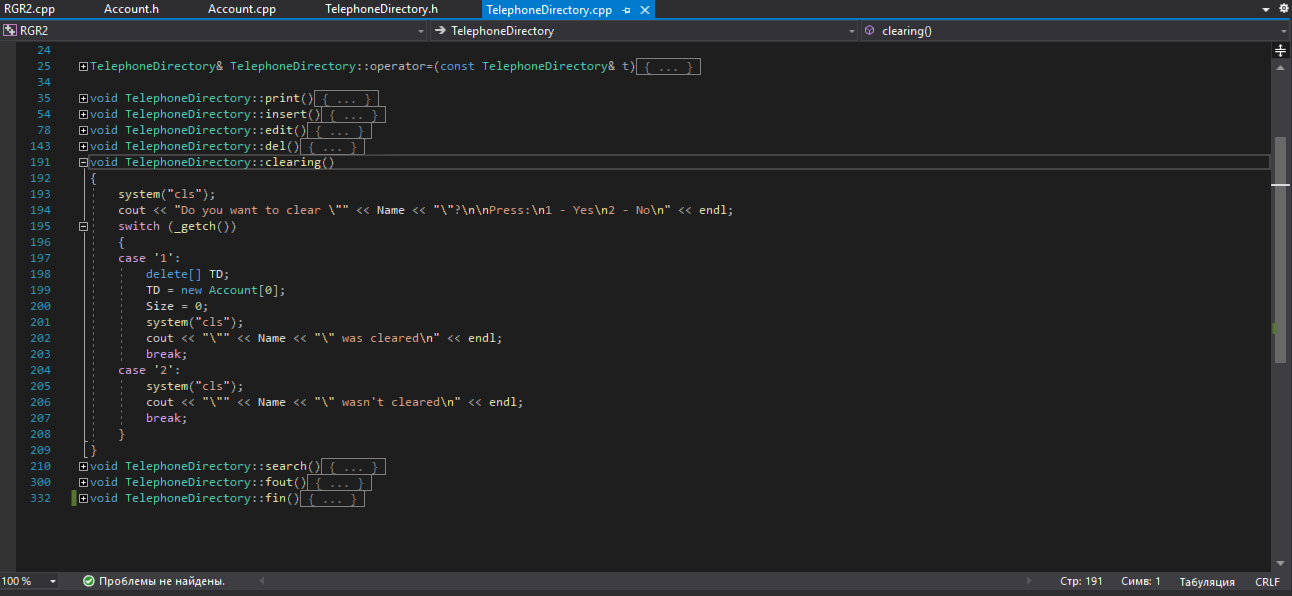
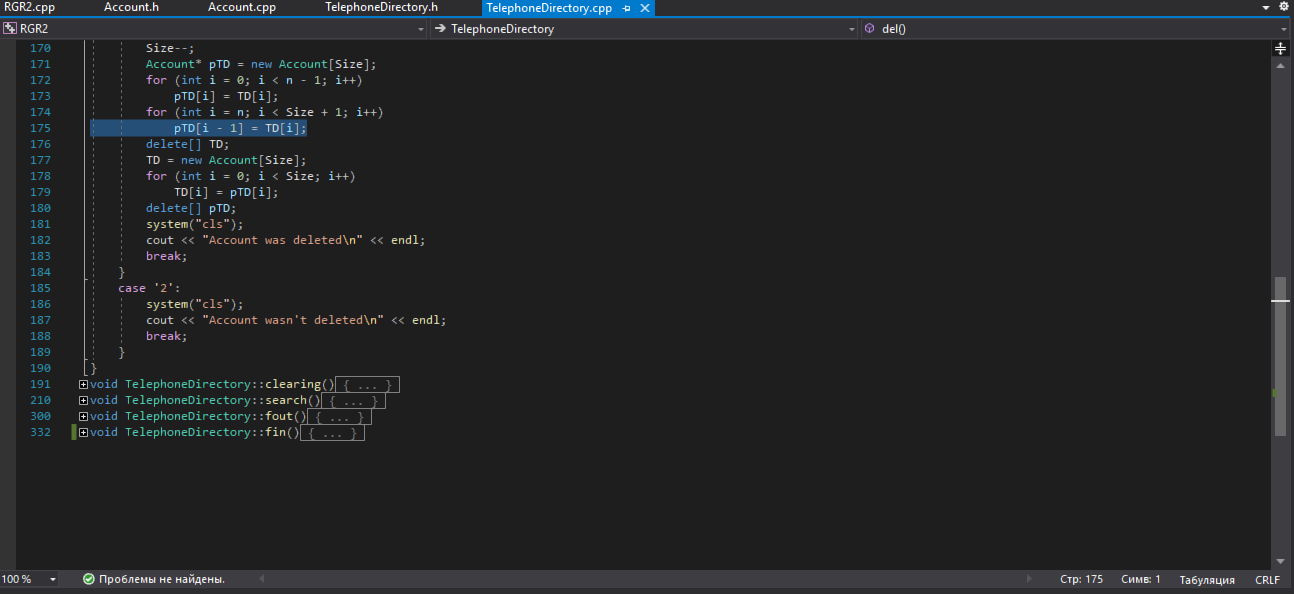
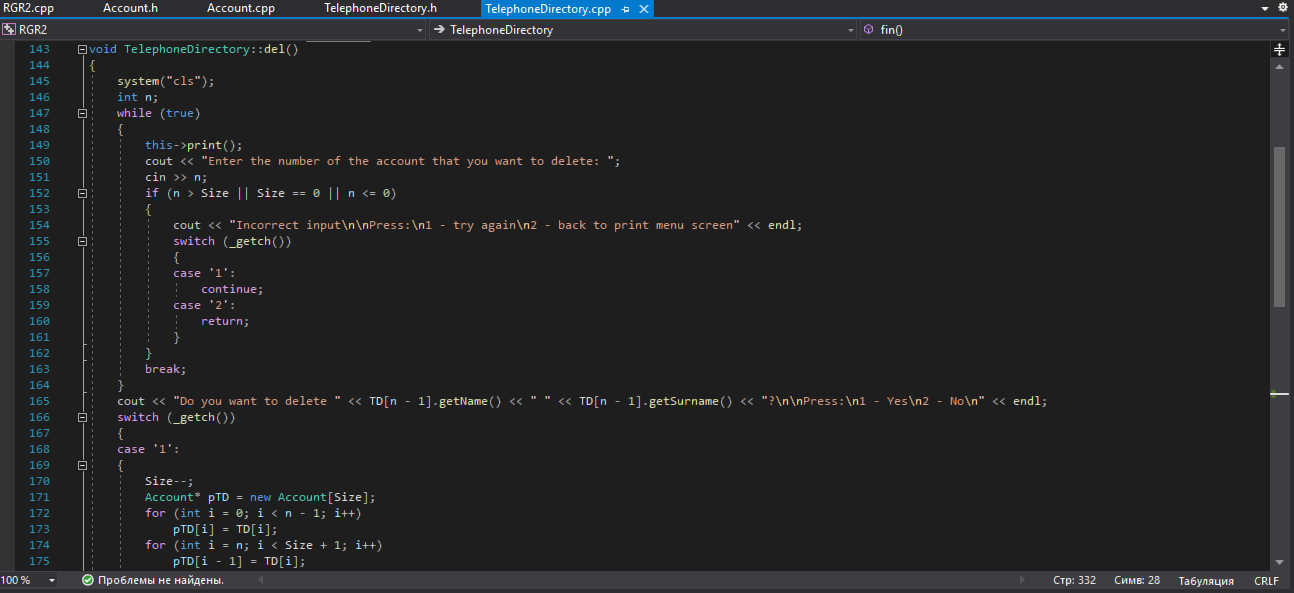
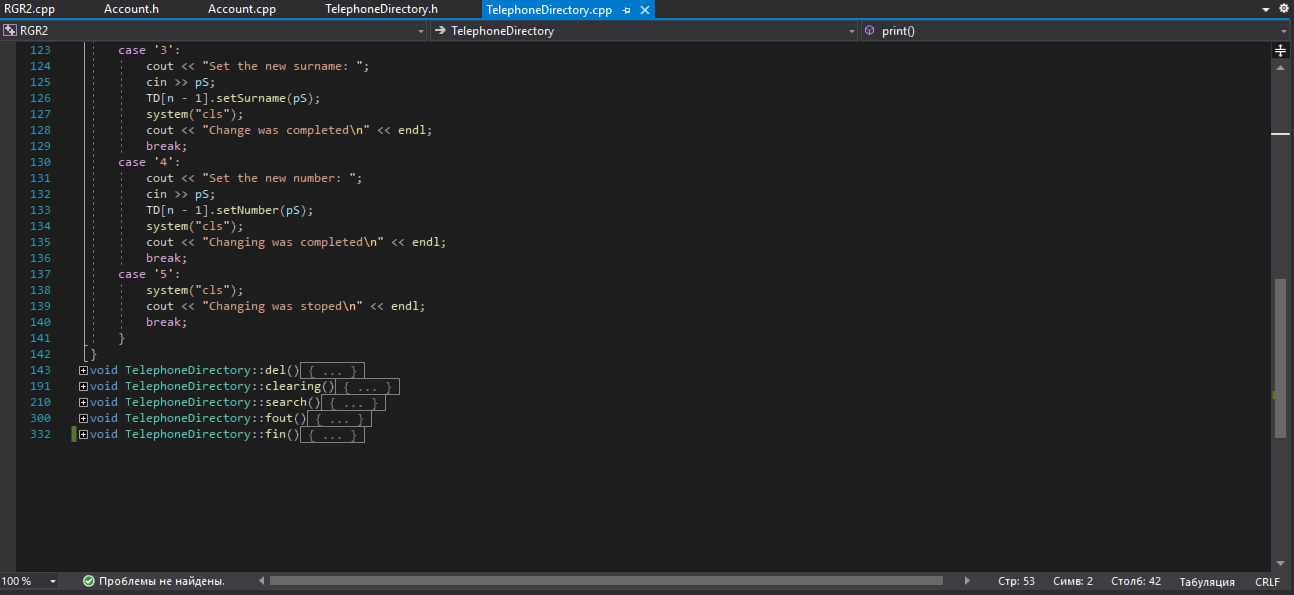
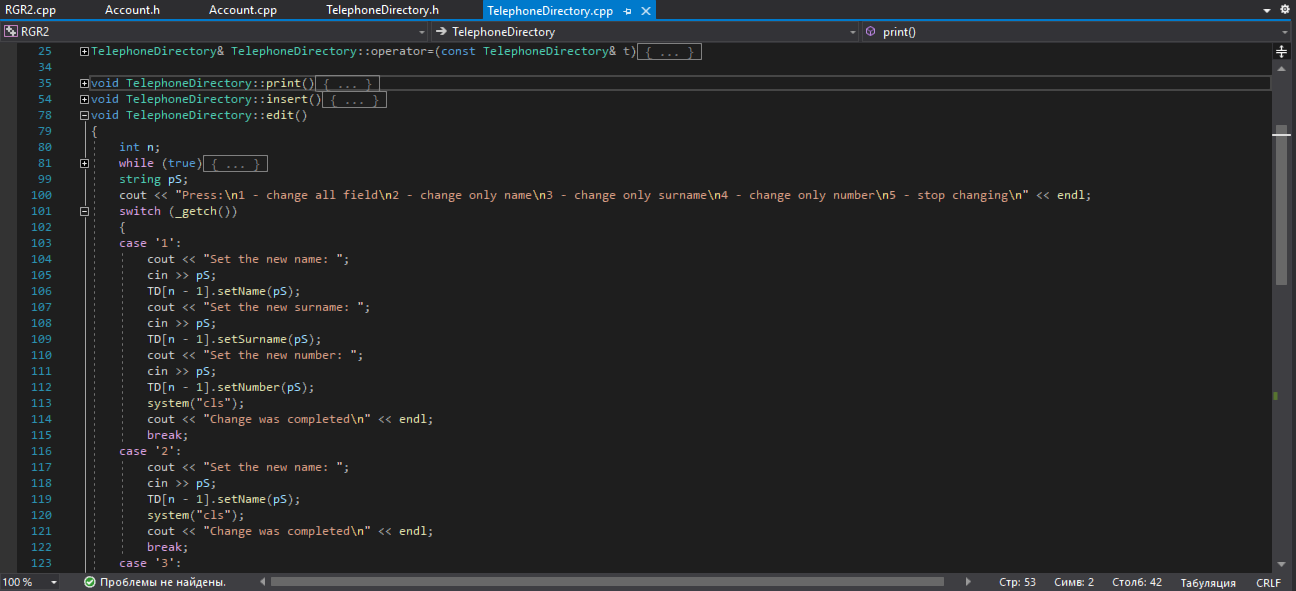
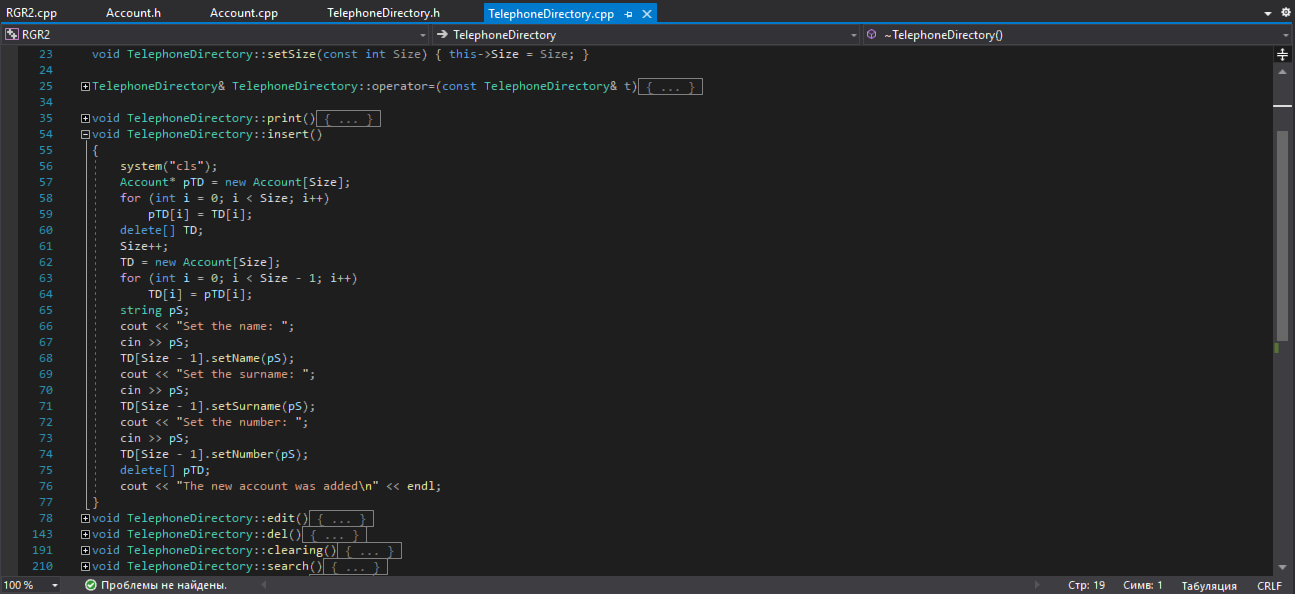
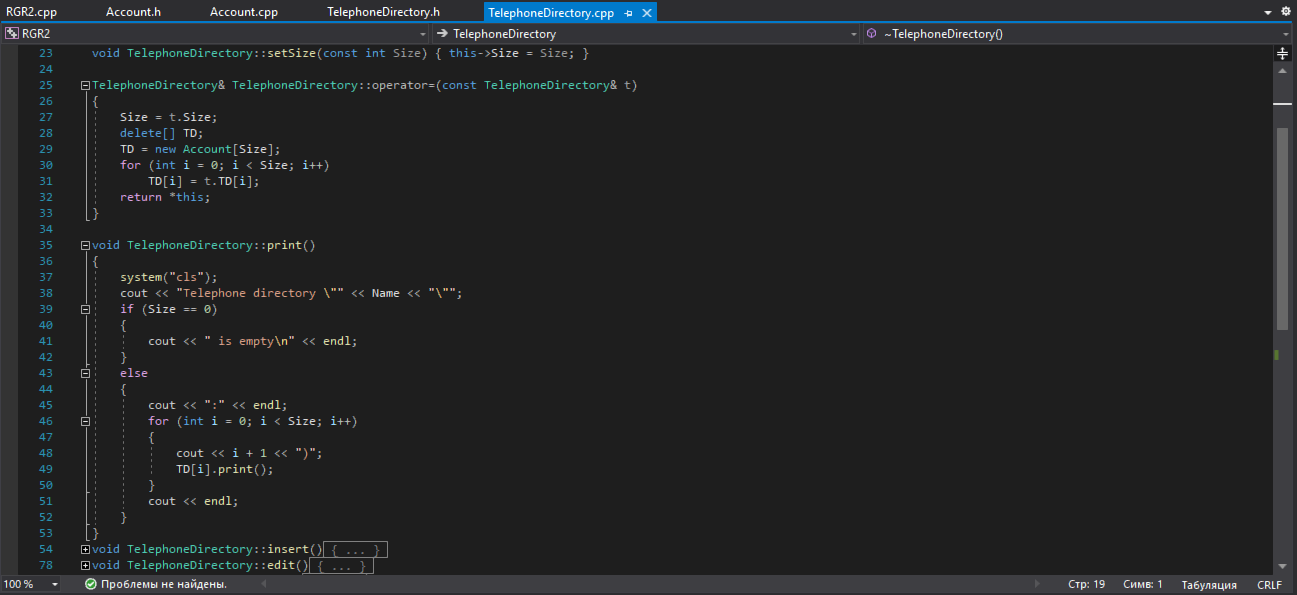
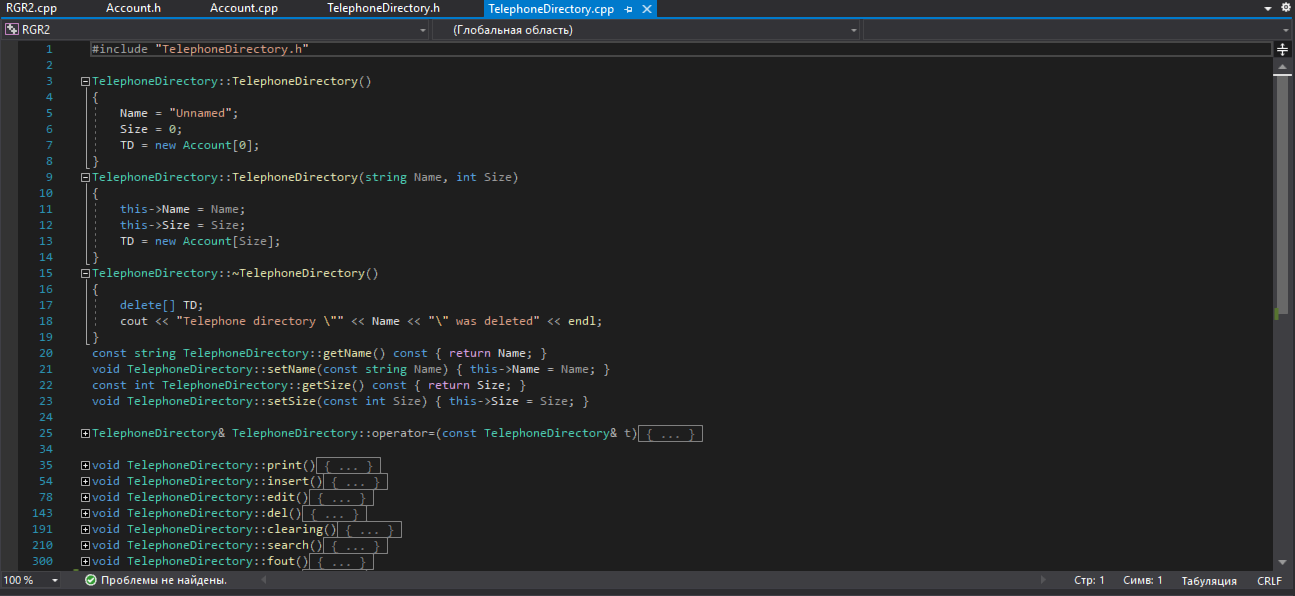
1. *fin* – осуществляет открытие файла в случае его совпадения с именем текущего ТС.

Выполняется в следующем порядке: объявляется *path* типа string, служащий временным хранилищем для имени файла, к имени дописывается “.txt”, что делает файл пригодным для открытия, память TD очищается, выделяется память под TD, равная «пустому» TD, создаются *c* типа int (служит счётчиком учётных записей), *cn* типа int (служит счётчиком слов записи), *pfin* типа string (служит для записи участков файла).

# Заключение

Благодаря накопленным за 2 семестра знаниям в области программирования, в частности объектно-ориентированного, мне удалось успешно создать программу, позволяющую удобнейшим образом записывать абсолютно любые контакты, например, преподавателей. Полученные знания, в особенности в области работы с файлами, бесценны и непременно помогут мне на жизненном пути.

# Код программы

# Литература

* [https://ru.wikipedia.org/wiki/C++](https://ru.wikipedia.org/wiki/C%2B%2B)
* [https://ru.wikipedia.org/wiki/Стандартная\_библиотека\_языка\_C++](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%BD%D0%B0%D1%8F_%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_%D1%8F%D0%B7%D1%8B%D0%BA%D0%B0_C%2B%2B)