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-ИБС11

очной формы обучения

Гринимаев Артем Вадимович

Проверил: ассистент каф. ИБС

Романчук С. П.

Саратов 2020

# Аннотация

Данная расчетно - графическая работа является реализацией телефонного справочника, написанного на языке С++ .

Описание программы

Планируется, что программа будет содержать класс «контакт», содержащий поля: «номер», «имя» , «организация».

Класс будет содержать методы get и set для каждого поля, а также перегруженные операторы ввода и вывода.

Все контакты будут содержаться в контейнере типа vector с названием directory, а сохраняться в файл directory.txt.

Для работы с контактами будет реализовано главное меню со следующей функциональностью:

* Просмотр всех контактов;
* Открытие контакта для его просмотра, или для его редактирования, или для его удаления;
* Поиск по всем контактам по выбранной характеристике;
* Создание нового контакта;
* Удаление всех контактов.

Реализация пунктов меню планируется с помощью отдельных функций.

Файл directory.txt планируется использовать для считывания контактов при запуске программы и для сохранения контактов при её завершении.

Содержание

[Аннотация 2](#_Toc43137534)

[Введение 4](#_Toc43137535)

[Теоретическая часть 5](#_Toc43137536)

[Практическая часть 10](#_Toc43137537)

[Заключение 10](#_Toc43137538)

[Приложения 11](#_Toc43137539)

# Введение

Тема работы:

«Телефонный справочник».

Цель работы:

Разработать приложение с функционалом телефонного справочника

Задачи работы:

1. Разработать класс для работы с одним контактом.
2. Организовать контейнер типа vector для хранения контактов, реализовать его загрузку и сохранение в файл.
3. Разработать главное меню с возможностью создания, редактирования, удаления контактов, а также их поиска.

# Теоретическая часть

Язык разработки.

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

Поддерживает такие парадигмы программирования как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование, обеспечивает модульность, раздельную компиляцию, обработку исключений, абстракцию данных, объявление типов (классов) объектов, виртуальные функции. Стандартная библиотека включает, в том числе, общеупотребительные контейнеры и алгоритмы. 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.

C++ широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также развлекательных приложений. Существует множество реализаций языка C++, как бесплатных, так и коммерческих и для различных платформ.

Необъектно-ориентированные особенности

Типы:

Символьные: char, wchar\_t (char16\_t и char32\_t, в стандарте C++11).

Целочисленные знаковые: signed char, short int, int, long int (и long long int, в стандарте C++11).

Целочисленные беззнаковые: unsigned char, unsigned short int, unsigned int, unsigned long int(и unsigned long long int, в стандарте C++11).

С плавающей точкой: float, double, long double.

Логический: bool.

Операции сравнения возвращают тип bool. Выражения в скобках после if, while приводятся к типу bool.

Функции могут принимать аргументы по ссылке. Функции могут возвращать результат по ссылке. Cсылки сходны с указателями, со следующими особенностями: перед использованием ссылка должна быть инициализирована; ссылка всегда указывает на один и тот же адрес; в выражении ссылка обозначает непосредственно тот объект или ту функцию, на которую она указывает, обращение же к объекту или функции через указатель требует разыменование указателя.

Объектно-ориентированные особенности

C++ добавляет к C объектно-ориентированные возможности. Он вводит классы, которые обеспечивают три самых важных свойства ООП: инкапсуляцию, наследование и полиморфизм.

Методы класса — это функции, которые смогут применяться к экземплярам класса. Грубо говоря, метод — это функция объявленная внутри класса и предназначенная для работы с его объектами. Методы объявляются в теле класса. Описываться могут там же, но могут и за пределами класса (внутри класса в таком случае достаточно представить прототип метода, а за пределами класса определять метод поставив перед его именем — имя класса и оператор ::). Методы и поля входящие в состав класса называются членами класса. При этом методы часто называют функциями-членами класса.

Наследование

В C++ при наследовании одного класса от другого наследуется реализация класса, плюс класс-наследник может добавлять свои поля и функции или переопределять функции базового класса. Множественное наследование разрешено. Конструктор наследника вызывает конструкторы базовых классов, а затем конструкторы нестатических членов-данных, являющихся экземплярами классов. Деструктор работает в обратном порядке. Наследование бывает публичным, защищённым и закрытым.

Полиморфизм

Целью полиморфизма, применительно к объектно-ориентированному программированию, является использование одного имени для задания общих для класса действий. Выполнение каждого конкретного действия будет определяться типом данных. Преимуществом полиморфизма является то, что он помогает снижать сложность программ, разрешая использование того же интерфейса для задания единого класса действий. Выбор же конкретного действия, в зависимости от ситуации, возлагается на компилятор. Полиморфизм может применяться также и к операторам.

Инкапсуляция

Основным способом организации информации в C++ являются классы. В отличие от структуры (struct) языка C, которая может состоять только из полей и вложенных типов, класс (class) C++ может состоять из полей, вложенных типов и функций-членов. Инкапсуляция в С++ реализуется через указание уровня доступа к членам класса: они бывают публичными (public), защищёнными (protected) и закрытыми (private). В C++ структуры отличаются от классов тем, что по умолчанию члены и базовые классы у структуры публичные, а у класса — собственные.

Стандартная библиотека

В языке программирования C++ термин Стандартная Библиотека означает коллекцию классов и 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.

Достоинства:

* Высокая совместимость с языком Си
* Вычислительная производительность
* Поддержка различных стилей программирования: структурное, объектно-ориентированное, обобщённое программирование, функциональное программирование, порождающее метапрограммирование.
* Автоматический вызов деструкторов объектов (в порядке обратном вызову конструкторов) упрощает и повышает надёжность управления памятью и другими ресурсами (открытыми файлами, сетевыми соединениями, т. п.).
* Перегрузка операторов
* Шаблоны (дают возможность построения обобщённых контейнеров и алгоритмов для разных типов данных)
* Возможность расширения языка для поддержки парадигм, которые не поддерживаются компиляторами напрямую
* Доступность. Для С++ существует огромное количество учебной литературы, переведённой на всевозможные языки

Недостатки:

* Плохо продуманный синтаксис сужает спектр применимости языка
* Язык не содержит многих важных возможностей
* Язык содержит опасные возможности
* Производительность труда программистов на языке оказывается неоправданно низка
* Громоздкость синтаксиса
* Тяжелое наследие
* Необходимость следить за памятью

**Code::Blocks** — свободная кроссплатформенная среда разработки. Code::Blocks написана на С++ и использует библиотеку wxWidgets. Имея открытую архитектуру, может масштабироваться за счёт подключаемых модулей. Поддерживает языки программирования С, С++, D (с ограничениями), Fortran.

Code::Blocks разрабатывается для Windows, Linux и Mac OS X. Среду можно собрать из исходников практически под любую Unix-подобную систему, например FreeBSD[2], PC-BSD

Возможности

Возможности компиляции

* Поддержка множества компиляторов (не включены)
* MinGW / GCC C/C++
* GNU ARM GCC Compiler
* GNU AVR GCC Compiler
* GNU GCC Compiler for PowerPC
* Microsoft Visual C++ 6[3]
* Microsoft Visual C++ Toolkit 2003
* Microsoft Visual C++ 2005/2008 (с некоторыми ограничениями)
* Microsoft Visual C++ 2010 (без поддержки отладчика, требуется DDK)

Возможности интерфейса

* Подсветка синтаксиса
* Сворачивание блоков кода
* Автодополнение кода
* Браузер классов
* Планировщик для нескольких пользователей
* Система проверки правописания (только для комментариев)
* Автоформатирование кода AStyle настраиваемый Code Style
* Утилита для разработки регулярных выражений (использует wxWidgets regexp parser, синтаксис pcre)
* Поиск по проекту с подсветкой найденных совпадений. Поддерживает регулярные выражения
* Поиск места декларации или реализации идентификатора (функции, макроса, класса и т. д.), включая подключенные заголовки из внешних библиотек
* Переход \*.h <-> \*.cpp (F11)

# Практическая часть

В данной работе я использую библиотеки:

* «string» - для работы со строками;
* «vector» - для работы с контейнерами типа «vector»;
* «fstream» - для работы с файловым вводом-выводом.
* «algorithm» - для использования алгоритмов обработки контейнера типа «vector»;
* «iterator» - для создания итераторов на контейнер типа «vector»

Программа состоит из 4 файлов: main.cpp, contact.h, mainMenu.h, directory.txt.

В файле main.cpp содержится объявление контейнера типа «vector», считывание с файла, вызов главного меню и сохранение в файл.

В файле contact.h содержить класс contact, методы get, set, перегрузки операторов ввода и вывода и их реализация.

В файле mainMenu.h хранится реализация главного меню и его функций.

В файл directory.txt сохраняются контакты.

# Заключение

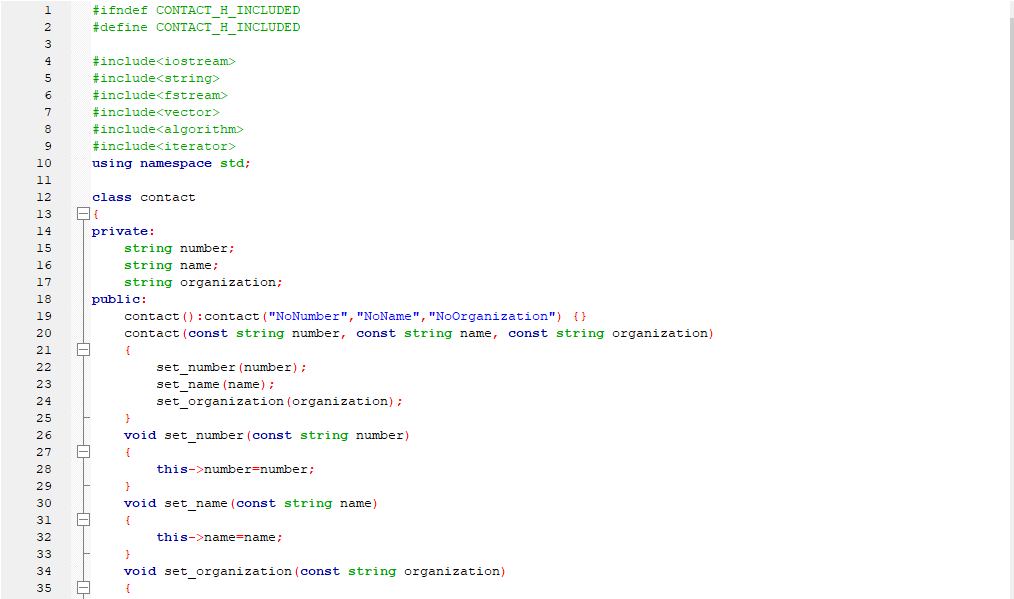
В результате мы получаем полноценное приложение, которое осуществляет всё ранее задуманное. В ходе проделывания данной работы я освоил объёмную часть теоретического материала по C++.

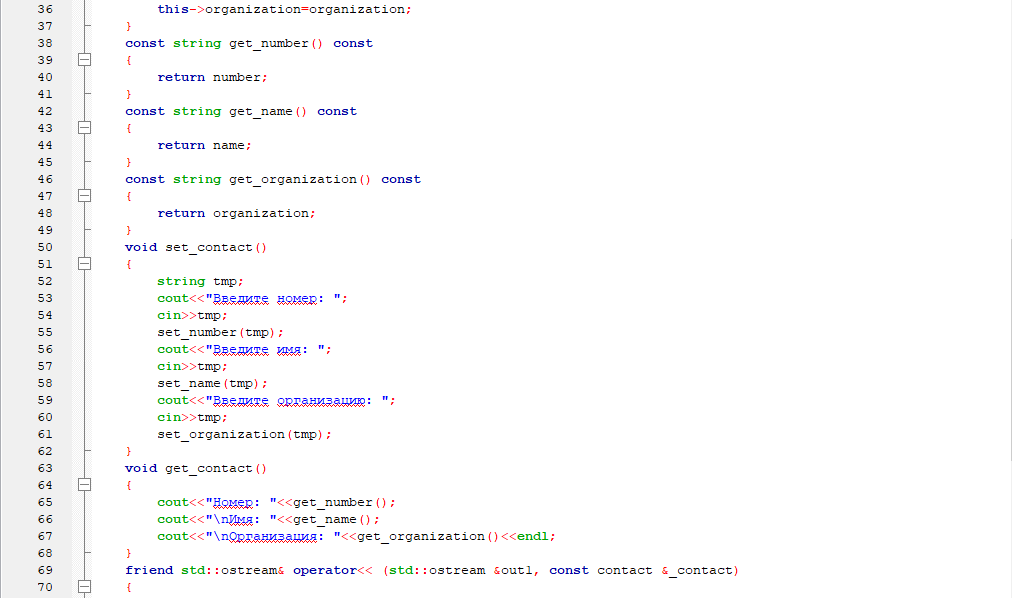
# Приложения

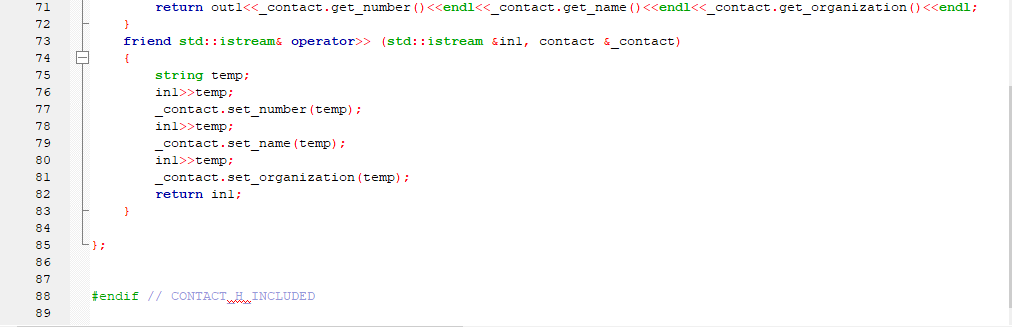
**main.cpp**

****

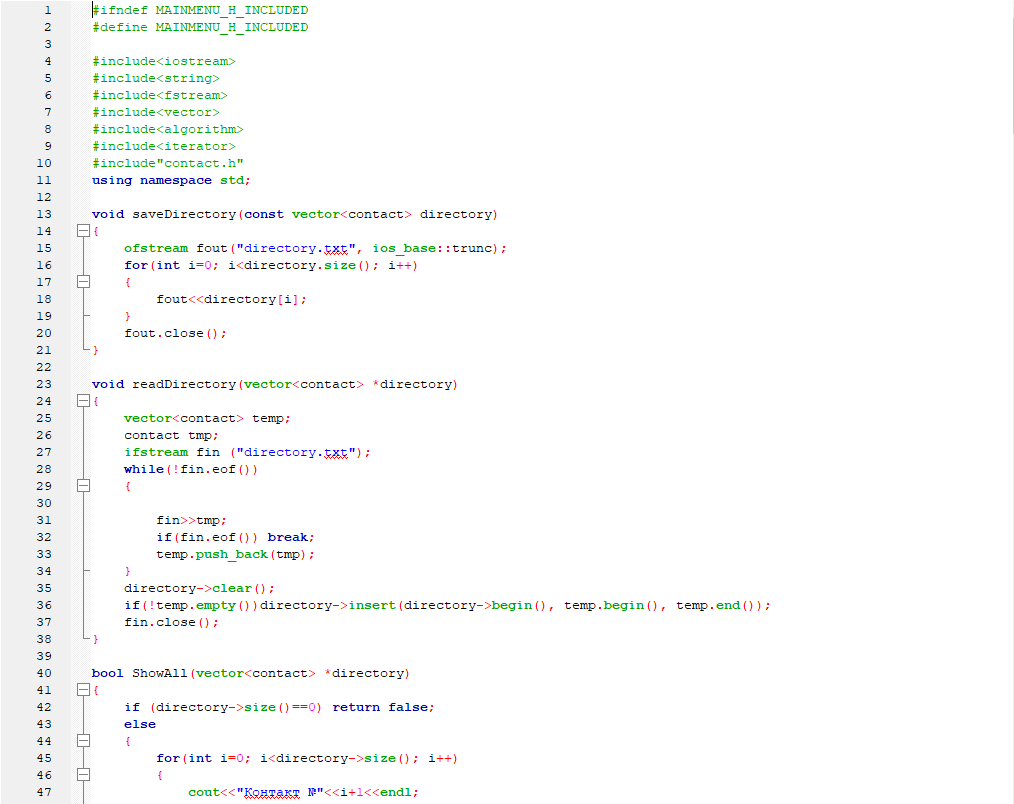
**contact.h**

****

****

****

**mainMenu.h**

****

****

****

****

****

****

****