Российский университет дружбы народов  
Инженерная академия  
Кафедра механики космического полёта

  
Курсовая работа по дисциплине: «Языки и методы программирования»  
«Разработка программы для создания базы данных тестовых заданий»

Выполнили: студенты 1 курса  
группы ИКБ-101  
Бондаренко Юрий Александрович  
Яблоновский Павел Алексеевич

Москва

2016

Оглавление

Оглавление

[Базы данных и объектно-ориентированние программирование C++ 3](#_Toc468641635)

Как была решена данная задача...............................................................................................................5

[Исходный код 6](#_Toc468641637)

Контрольное тестирование программы.................................................................................................11

Базы данных C++

**База данных** — совокупность данных, организованных в соответствии с концептуальной структурой, описывающей характеристики этих данных и взаимоотношения между ними, причём такое собрание данных, которое поддерживает одну или более областей применения.

Одной из задач проектирования базы данных является обеспечение уникального определения каждого экземпляра сущности, т.е. необходимо сделать так, чтобы система могла извлечь из таблицы нужные ей строки. Иными словами, строки должны быть однозначно идентифицированы.

Одну строку от другой отличают по значениям первичного ключа. Под первичным ключом понимают поле или набор полей, однозначно (уникально) идентифицирующих запись. Первичный ключ должен быть минимально достаточным: в нем не должно быть полей, удаление которых из первичного ключа не отразится на его уникальности. Правила хорошего тона при разработке структур баз данных, и чисто практические соображения должны побудить разработчика всегда определять первичный ключ для таблицы базы данных.

Вместе с первичным ключом существует и так называемый уникальный ключ. Уникальные ключи несут аналогичную нагрузку - они также служат для однозначной идентификации записей в таблице. Отличие первичных ключей от уникальных состоит в том, что первичный ключ может быть в таблице только один, а уникальных ключей - несколько.

Первичный и уникальный ключи задаются при создании таблицы, когда соответствующие поля относят к первичному или уникальному ключу. Если поле помечено как уникальное, то в него для каждой строки можно вводить только различные данные. Если в некоторую строку поля ввести значение, которое уже встречалось в какой-либо другой строке, то система выдаст сообщение об ошибке. Так должна работать любая база данных.

**Структура:**

Структура объектной модели описываются с помощью трех ключевых понятий:

**Инкапсуляция** - каждый объект обладает некоторым внутренним состянием (хранит внутри себя запись данных), а также набором методов - процедур, с помощью которых (и *только* таким образом) можно получить доступ к данным, определяющим внутреннее состояние объекта, или изменить их. Таким образом, объекты можно рассматривать как самостоятельные сущности, отделенные от внешнего мира.

**Наследование** - подразумевает возможность создавать из классов объектов новые классы объекты, которые наследуют структуру и методы своих предков, добавляя к ним черты, отражающие их собственную индивидуальность. Наследование может быть простым (один предок) и множественным (несколько предков).

**Полиморфизм** - различные объекты могут по разному реагировать на одинаковые внешние события в зависимости от того, как реализованы их методы.

Для поддержания целостности объектно-ориентированный подход предлагает использовать следующие средства:

* автоматическое поддержание отношений наследования
* возможность объявить некоторые поля данных и методы объекта как "скрытые", не видимые для других объектов; такие поля и методы используются только методами самого объекта
* создание процедур контроля целостности внутри объекта

Появление объектно-ориентированных СУБД вызвано потребностями программистов на ОО-языках, которым были необходимы средства для хранения объектов, не помещавшихся в оперативной памяти компьютера. Также важна была задача сохранения состояния объектов между повторными запусками прикладной программы. Поэтому, большинство ООСУБД представляют собой библиотеку, процедуры управления данными которой включаются в прикладную программу.

**Объектно-ориентированное программирование (ООП)** — методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования. Основой ООП является абстракция, Фундаментальная идея которой состоит в разделении несущественных деталей реализации подпрограммы и характеристик, существенных для ее корректного использования.

Основой объектно-ориентированного программирования является идея классов. **Класс** – тип данных, определяемый пользователем. В классе задаются свойства и поведение какого-либо предмета или процесса в виде полей данных и функций (методов) для работы с ними. Существенным свойством класса является то, что детали его реализации скрыты от пользователя интерфейсом. Интерфейсом класса являются заголовки его методов.

**Решение поставленной задачи**

Для решения поставленной задачи мы использовали таки функции, как:

void delete\_ex(std::string);//Удаление задания

void show\_test(std::string)//Демонстрации задания

void rewrite\_string (std::string);//Перезапись строки

void add\_quest(std::string);//Добавление задания

exam (std::string);//Конструктор

**Конструктор** осуществлял создание таких объектов, как: exercise - Заданеие, answers - Ответы, corr\_answer- Правильный ответ, hint - Подсказка, add\_inform - Дополнительная информация и запись всех этих объектов, введенных пользователем, в файл. Название файла также вводил пользователь.

**Демонстрация задания** была выполнена с помощью функции getline (test,line);   
Так же учитывается, что если файл не открыт, то выводит ошибку, означающую что файл не открыт. Это же условие используется и в дальнейших функциях

**Удаление задания** было выполнено через класс шаблона ifstream.  
ifstream определяет поток, используемый для чтения однобайтовой последовательности символьных данных из файла. **ifstream** — это определение typedef, которое специализирует класс шаблона **basic\_ifstream** для **char и string**.

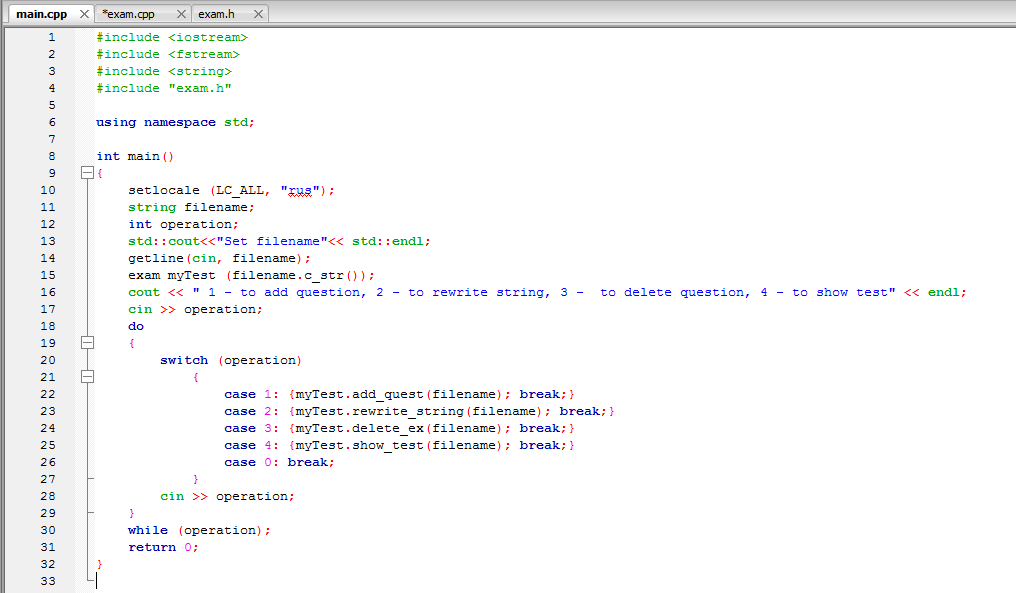
Также существует **wifstream** — определение typedef, которое специализирует **basic\_ifstream** для чтения двухбайтовых символов**wchar\_t**.Дополнительные сведения см. в описании [wifstream](https://msdn.microsoft.com/ru-ru/library/3b5w4311.aspx).  
Также шаблонный класс ifstream был использован в других функциях.

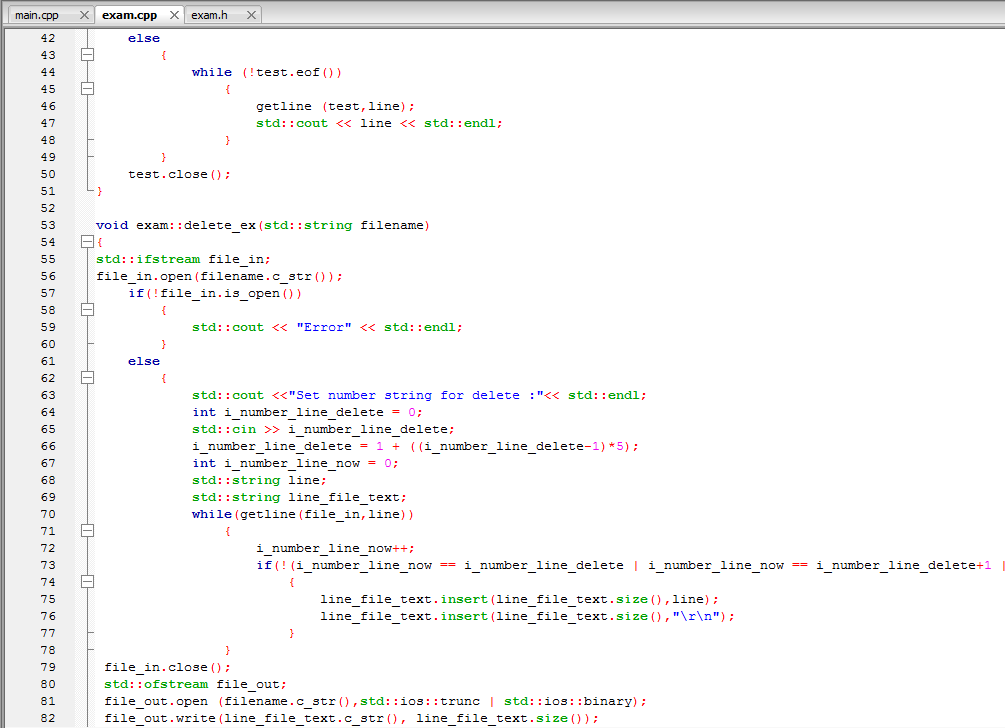
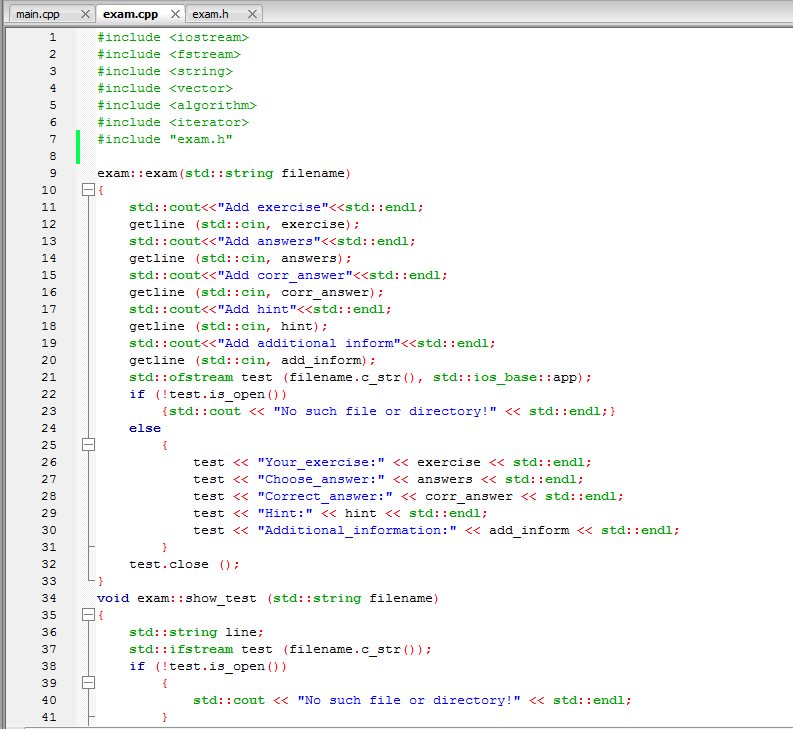
**Перезапись строки** была осуществлена через vector.Vector в C++ — это замена стандартному [динамическому массиву](https://code-live.ru/post/cpp-dynamic-arrays/), память для которого выделяется вручную, с помощью оператора new. Разработчики языка рекомендуют в использовать именно vector вместо ручного выделения памяти для массива. Это позволяет избежать утечек памяти и облегчает работу программисту.

**Добавление вопроса** базируется на конструкторе, с небольшими изменениями в вызове самой функции.

**Исходный код программы**

main.cpp

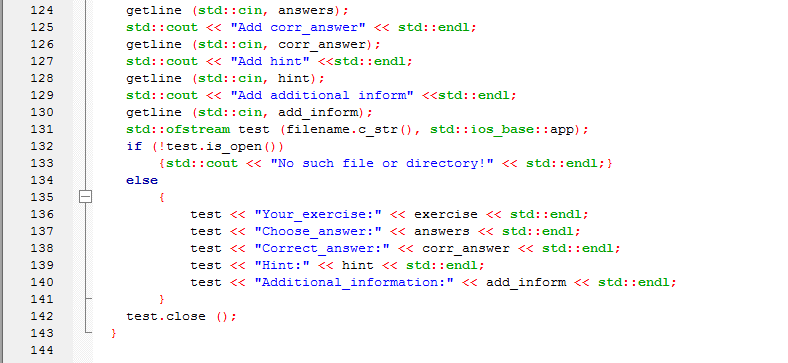
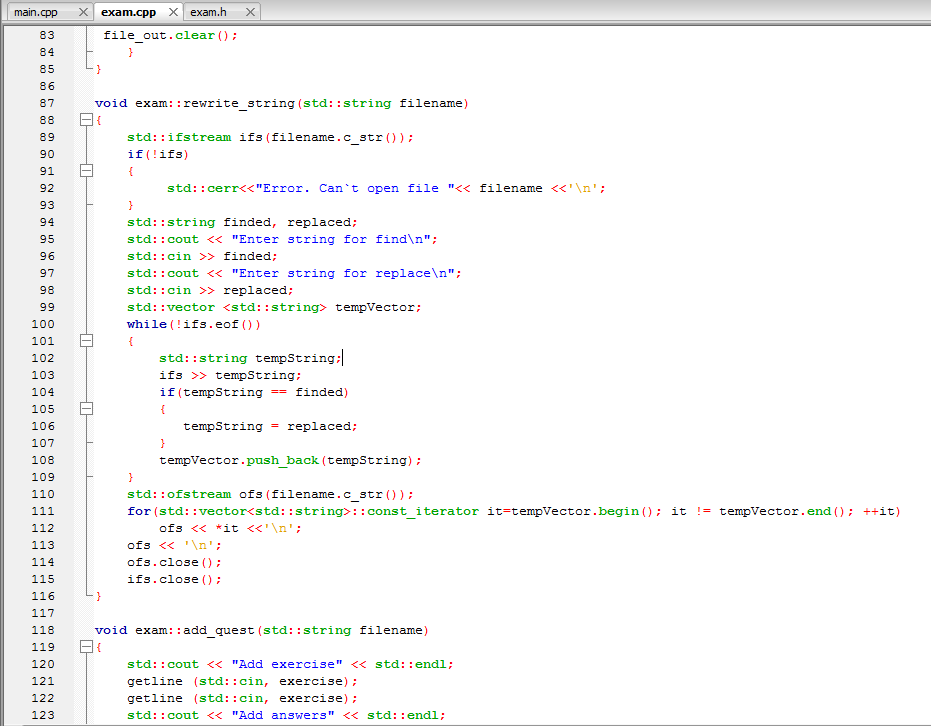


Exam.cpp

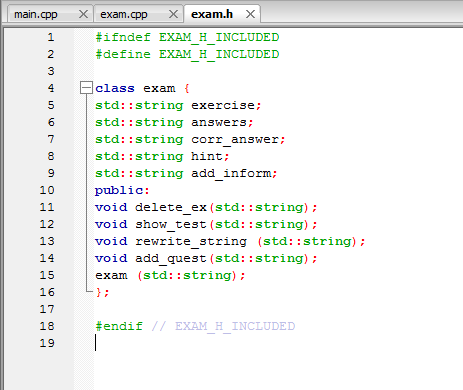
C:\Users\Gamer\Desktop\Безымянныыаываываывавыавыаывй.png

Условие If, которое не влезло в скриншот:

if(!(i\_number\_line\_now == i\_number\_line\_delete | i\_number\_line\_now == i\_number\_line\_delete+1 | i\_number\_line\_now == i\_number\_line\_delete+2 | i\_number\_line\_now == i\_number\_line\_delete+3 | i\_number\_line\_now == i\_number\_line\_delete+4))



Exam.h



**Контрольное тестирование программы**

Список использованной литературы

* Пахомов Б.И. С/С++ и MS Visual C++ 2012 для начинающих.
* http://informatic.ugatu.ac.ru/lib/office/Proekt.htm
* http://wm-help.net/lib/b/book/1940220047/42
* http://www.mstu.edu.ru/study/materials/zelenkov/ch\_6\_3.html
* C/C++. Программирование на языке высокого уровня/ Т. А. Павловская. – СПб.: Питер, 2003. – с. 173 – 176.
* https://msdn.microsoft.com/ru-ru/library/ae1k9a9f.aspx