**Московский государственный технический университет  им. Н.Э.Баумана**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| УТВЕРЖДАЮ: |  |
| Папшев И.С. | "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2020 Г. |

**Проектно-технологическая практика по дисциплине ПКШ.**

**«СУБД интернет-магазина музыкальных инструментов»**

Руководство пользователя

(вид документа)

писчая бумага

(вид носителя)

7

(количество листов)

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| Бригада группы ИУ5-21:  Агличеев М. С. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. |
|  |  |
|  |  |
|  |  |

Москва  -  2020

СОДЕРЖАНИЕ

1. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ ПРОГРАММЫ…………...3

2. ПОДКЛЮЧЕНИЕ СИСТЕМЫ КЛАССОВ И ФУНКЦИЙ………………….3

3. РАБОТА С СИСТЕМОЙ КЛАССОВ…………………………………………3

3.1. Общие сведения о системе классов и функций…………………………….3

3.2. Подключение системы классов и функций в программу………………….3

4. КЛАСС DBTableTxt……………………………………………………………4

4.1. Данные и переменные класса DBTableTxt ……………….………………..4

4.2. Конструкторы класса DBTableTxt…………………………………………..4

4.3. Деструктор класса DBTableTxt……………………………………………...4

4.4. Методы класса DBTableTxt………………………………………………….4

5. Класс DBTableSet...…………………………………………………………….5

6. ОТКЛЮЧЕНИЕ СИСТЕМЫ КЛАССОВ……………………………………..7

7. СООБЩЕНИЯ ОБ ОШИБКАХ И ИСКЛЮЧИТЕЛЬНЫЕ СИТУАЦИИ…..7

**1. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ ПРОГРАММЫ**

Система классов функций для обработки таблиц предназначена для автоматизации работы с объектами таблиц в программных проектах и предметной области, где необходимо это учитывать. В частности, система классов и функций предназначена для программирования задач, включающих действия: изменение содержимого таблиц в связи с поступлением новых данных от пользователя, обработка числовых данных таблиц (в данной программе обработка количества товара на складе).  Система классов обеспечивает удобную работу с этими объектами, высокий уровень надежности программ, функциональных возможностей, а также сокращение сроков разработки и реализации программных продуктов, где необходимо использовать подобные объекты.

**2. ПОДКЛЮЧЕНИЕ СИСТЕМЫ КЛАССОВ**

Для того, чтобы установить данную систему классов на компьютер следует произвести следующие действия:

1. Освободить на жестком диске как минимум 10 мегабайт свободного пространства.
2. Вставить флеш-накопитель с программным обеспечением.
3. Скопировать файлы с помощью системной функции копирования с носителя на жесткий диск в нужный каталог. Включить в проект библиотеку dbmsLib\_v1, E-Library (библиотека с разработанным классом) и заголовочный файл dbmsLib\_v1.h.

1. **РАБОТА С СИСТЕМОЙ КЛАССОВ И ФУНКЦИЙ**

**3.1. Общие сведения о системе классов и функций для обработки таблиц**

Система классов и функций описывает таблицы, в которых содержится информация о результатах сессии студентов кафедры.

Система классов позволяет программировать следующие операции и функции:

* Создание и задание свойств объектов типа таблица;
* Добавление строк в таблицу, их редактирование и удаление;
* Чтение/запись в файл/печать таблиц.

**3.2. Подключение системы классов и функций в программу**

В данном фрагменте подключаются заголовочные файлы библиотек: ввода вывода (**iostream, iomanip**), системы классов таблиц (**dbmsLib\_v1.h**);   
 Кроме того, объявлено использование именованного пространства стандартных функций (**std**). В проект подключается тестовая программа Testing.cpp.

#include <iostream>

#include <iomanip>

#include "dbmsLib\_v1.h"

using namespace std;

1. **КЛАСС DBTableTxt**

**Класс DBTableTxt.** Объекты данного типа содержат информацию о размерности таблицы, заголовках, файле, в котором она содержится. Предусматривается возможность изменения таблиц.

**4.1. Данные и переменные класса DBTableTxt**

В классе **DBTableTxt** объявлены следующие свойства доступные пользователю: 

|  |  |  |  |
| --- | --- | --- | --- |
| **Переменная** | **Тип свойства** | **Защита** | **Назначение** |
| **columnHeaders** | Header | private | Заголовок столбца |
| **tableName** | string | private | Имя таблицы |
| **primaryKey** | string | private | Идентификационный ключ |
| **fileName** | string | private | Имя файла |

**4.2. Конструкторы класса DBTableTxt**

В классе **DBTableTxt** описаны следующие конструкторы:

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Прототип** | **Тип возврата** | **Назначение /Параметры** |
|  | **DBTableTxt**() | **DBTableTxt &** | Нет |
|  | **DBTableTxt**(string tabName) | **DBTableTxt &** | Новый с параметром:  **tabName –** имятаблицы |

**4.3. Деструктор класса DBTableTxt**

Прототип деструктора:

~**DBTableTxt**();

**4.4. Методы класса DBTableTxt**

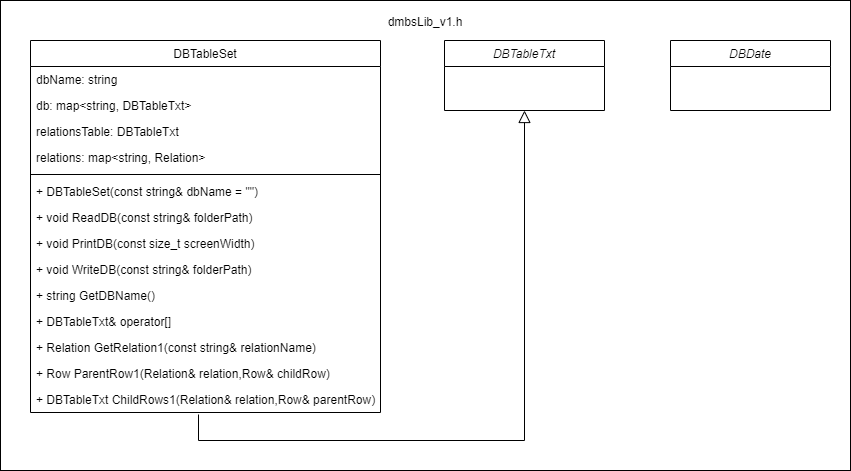
В таблице представлены методы класса **DBTableTxt** (часть из них оформлена как дружественные функции).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Прототип** | **Тип возврата** | **Назначение** | **Параметры** |
|  | friend void ReadDBTableTxt1 (DBTableTxt& tab, const string& fileName) | void | Чтение таблицы | **tab –** таблица, в которую считываются данные  **fileName -**  путь до файла с таблицей |
|  | friend void PrintDBTable1 (DBTableTxt& tab, const int screensize) | void | Печать таблицы | **screensize** – ширина экрана |
|  | friend void WriteDBTableTxt1 (DBTableTxt& tab, const string& fileName) | void | Запись таблицы в файл | **tab –** таблица, из которой считываются данные  **fileName -**  путь до файла, куда требуется записать таблицу |
|  | **int** GetSize() | int | Получить размер | Нет |
|  | **void** SetFileName(string path) | void | Задать имя файла | **path -** путь до папки с файлом с таблицей |
|  | **void** SetTableName(string tName) | void | Задать имя таблицы | **tName –** имя таблицы |
|  | string GetFileName(); | void | Получить имя файла | Нет |
|  | string GetTableName() | string | Получить имя таблицы | Нет |
|  | string GetPrimaryKey() | string | Получить ключ | Нет |
|  | Header GetHeader(); | Header | Получить заголовок | Нет |
|  | **void** SetHeader(Header& hdr); | void | Задать заголовок | **hdr** – заголовок |
|  | Row CreateRow(); | Row | Создать строку | Нет |
|  | Row& operator[](int ind); | Row& | Получить строку | **index** – индекс |
|  | **void** AddRow(Row row); | void | Добавить строку | **row** – строка |
|  | void RemoveRow(int rowIndex) | void | Удалить строку | **rowIndex** – индекс строки |
|  | void EditRow(int rowIndex, const string& columnName, string& value) | void | Измененить строку | **rowIndex** – индекс строки  **columnName** – название столбца  **value** – значение, на которое требуется сделать замену |
|  | int FindValue(const string& columnName, void\* value)  int FindValue(const string& columnName, const string& value)  int FindValue(const string& columnName, void\* value, const string& columnName2, void\* value2)  int FindValue(const string& columnName, const string& value, const string& columnName2, const string& value2); | int | Поиск строки, значение/я в которой соответствует/ют значению/ям, переданным в качестве аргументов | **columnName, columnName2 –** имена столбцов, в которых производится поиск нужного значения  **value, value2 –** искомые значения |
|  | DBTableTxt SelfRows(string columnName, Condition cond, void\* value) | DBTable | Выборка из таблицы строк, в которых значение поля colName и значение value удовлетворяют условию cond | **columnName** – значение поля;  **cond** – условие;  **value** – заданное зачение |
|  | string valueToString(Row& row, string columnName); | string | Получить строкового представления значения поля | **row** – строка с требуемым полем  **columnName –** название требуемого столбца |
|  | void CreateTableMaket(Strip\*& strips, int& nStrips, int screenWidth) | void | Создаие макета распечатки таблицы на экране соответствующей ширины. Поля таблицы разбиваются на полосы, выводимые поочерёдно. | **strips –** Описание полос вывода информации на экран  **nStrips –** количество полос  **screenWidth –** название требуемого столбца |

**5. Класс DBTableSet**

**Цель**: Предоставить удобный интерфейс и множество функций для работы со студентами, книгами и абонементами.

**UML-схема класса:**



Атрибуты класса:

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Атрибут** | **Тип** | **Назначение** |
|  | dbName | string | Название БД |
|  | db | Map<string, DBTableTxt> | База данных – контейнер из отдельных таблиц |
|  | relationsTable | DBTableTxt | Таблица отношений между остальными таблицами (удобно записывать в файл) |
|  | relations | Map <string, Relation> | Контейнер с отношениями между таблицами |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Прототип** | **Тип возврата** | **Назначение** | **Параметры** |
|  | DBTableSet(const string& dbName = "") | DBTableSet& | Конструктор | **dbname –** название БД |
|  | void ReadDB(const string& folderPath) | void | Чтение БД из папки | **folderPath -** папка, в которой находится БД |
|  | void PrintDB(const size\_t screenWidth) | void | Вывод всех таблиц БД на экран | **screenWidth –** ширина экрана |
|  | void WriteDB(const string& folderPath) | void | Запись БД в папку | **folderPath –** папка, в которую нужно записать БД |
|  | string GetDBName() | string | Получение названия БД | Нет |
|  | DBTableTxt& operator[](string tableName) | DBTableTxt& | Получение конкретной таблицы | **tableName –** название искомой таблицы |
|  | Relation GetRelation1(const string& relationName) | Relation | Получение отношения между таблицами по его имени | **relationName –** название искомого отношения |
|  | Row ParentRow1(Relation& relation,Row& childRow) | Row | Получение строки из «родительской» таблицы, соответствующей строке из «детской» таблицы | **relation –** отношение между таблицами  **childRow –** строка из «детской» таблицы |
|  | DBTableTxt ChildRows1(Relation& relation,Row& parentRow) | DBTableTxt | Получение строк из «детской» таблицы, соответствующих строке из «родительской» таблицы | **relation –** отношение между таблицами  **parentRow –** строка из «родительской» таблицы |

**6. ОТКЛЮЧЕНИЕ СИСТЕМЫ КЛАССОВ**

Чтобы отключить данную систему классов, надо с помощью системной функции удаления (или файл менеджера) удалить файлы, перечисленные в пункте 2.3 данного документа, из каталогов, куда они были скопированы.

**7. СООБЩЕНИЙ ОБ ОШИБКАХ И ИСКЛЮЧИТЕЛЬНЫЕ СИТУАЦИИ**

В системе классов таблиц предусмотрена обработка исключительных ситуаций и выдача диагностических сообщений об ошибках. Выдаваемые системные диагностические сообщения и возникающие системные исключительные ситуации описаны в документации на систему программирования (MS VS 2012).