

**Министерство образования Республики Беларусь**

Учреждение образования «Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет инновационного непрерывного образования

Кафедра электронных вычислительных машин

Дисциплина: Системное программирование обеспечение  
вычислительных машин

К ЗАЩИТЕ ДОПУСТИТЬ

---

**ПОЯНИТЕЛЬНАЯ ЗАПИСКА**

к курсовой работе

на тему:

«Встраиваемая СУБД»

Студент

Д.А. Мисан

Руководитель

Л.П. Поденок

Минск 2020

## Содержание

Введение .....	3
1. Анализ предметной области.....	4
1.1. Теоретические сведения .....	4
1.2. Постановка задачи .....	6
2. Описание реализации .....	7
2.1. Выбор и обоснование средств реализации .....	7
2.2. Алгоритмы функций .....	9
3. Руководство пользователя.....	10
Заключение.....	12
Список использованной литературы .....	13

## **ВВЕДЕНИЕ**

Данное приложение является библиотекой-оберткой. Библиотека служит для встраивания в сложные системы для работы с БД.

В приложении используется не SQL база, а база данных с индексно-последовательными прямыми методами доступа (Berkeley DB).

На ее базе сделана библиотека-обертка (wrapper-library) на си, реализующая несколько простых функций:

- Add;
- Delete;
- Set;
- Get;
- Put;
- Next;
- Prev.

# 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1. Теоретические сведения

Berkeley DB (BDB) — высокопроизводительная встраиваемая система управления базами данных, реализованная в виде библиотеки. BDB является нереляционной базой данных — она хранит пары «ключ — значение» как массивы байтов и поддерживает множество значений для одного ключа. BDB может обслуживать тысячи процессов или потоков, одновременно манипулирующих базами данных размером в 256 терабайт, на разнообразном оборудовании под различными операционными системами, включая большинство UNIX-подобных систем и Windows, а также на операционных системах реального времени.

Первая версия Berkeley DB была разработана в Университете Беркли во время разработки BSD версии 4.3 (июнь 1986 года). Netscape попросила авторов Berkeley DB улучшить и расширить библиотеку — в то время версию 1.85, — чтобы она удовлетворяла их требованиям к использованию в сервере LDAP и в браузере Netscape. Этот запрос привёл к созданию Sleepycat Software (купленной корпорацией Oracle в феврале 2006 года). Berkeley DB распространяется под лицензией Sleepycat Public License (англ.), которая была одобрена OSI и FSF. Программа поставляется с полным исходным кодом, средствами сборки, инструментами тестирования и документацией. Качество кода и практичность вместе со свободной лицензией привели к использованию Berkeley DB во многих свободных и открытых программах. В рамках техники двойного лицензирования Oracle также распространяет проприетарную лицензию на использование библиотеки в закрытых проектах.

Berkeley DB примечательна своей простой архитектурой в сравнении с другими системами баз данных, такими как, например Microsoft SQL Server и Oracle Database. Например, в ней отсутствует сетевой доступ — программы используют базу данных через вызовы внутрипроцессного API. Она поддерживает SQL в качестве одного из интерфейсов, начиная с версии 5.0, хотя и не поддерживает столбцы в таблицах в традиционном понимании на уровне внутренней архитектуры. Berkeley DB предполагает работу с парами ключ-значение, где ключ и значение могут иметь фиксированную или переменную длину, а функция сравнения ключей может быть написана и назначена прикладным программистом. Программа, которая использует БД, сама решает, как данные сохраняются в записи; БД не налагает ограничений на данные, хранимые в записях. Запись и её ключ оба могут иметь размер до четырёх гигабайт.

Berkeley DB поддерживает необходимые возможности баз данных, такие как ACID-транзакции, детальные блокировки, интерфейс распределённых

транзакций XA, горячее резервное копирование и репликацию. Berkeley DB может использоваться как средство для построения хранимых индексов, так и в качестве хранилища данных.

Oracle предлагает BDB в трёх вариантах:

- Berkeley DB — собственно библиотека на языке «С»;
- Berkeley DB Java — библиотека, переписанная на Java (поддержка Google Android, Apache Maven);
- Berkeley DB XML — библиотека на Си, реализующая XML-СУБД на основе Berkeley DB со средствами работы с XML (Xerces, XPath, XQuery, XQilla).

Berkeley DB входит в состав большинства дистрибутивов Linux. Существуют средства для работы с Berkeley DB на языках Perl, Python и других.

## 1.2. Постановка задачи

На базе Berkeley DB создать библиотеку-обертку (wrapper-library), реализующую несколько простых функций:

- add – добавляет запись в БД;
- delete – удаляет запись из базы данных;
- set – устанавливает позицию в ключе;
- get – получает запись по значению ключа;
- put – сохраняет запись по значению ключа;
- next/prev – получает следующую/предыдущую запись в последовательности ключа.

## **2. ОПИСАНИЕ РЕАЛИЗАЦИИ**

### **2.1. Выбор и обоснование средств реализации**

Для разработки приложения используем официальную среду разработки - Microsoft Visual Studio.

Visual Studio более просторен в плане языков программирования, на которых можно писать приложения. Так же у Visual Studio есть просто огромный плюс в отладчике – панель, отображающая состояния объектов, значения переменных и т.д. в текущий момент выполнения программы.

Для реализации приложения была выбрана система программирования Microsoft Visual Studio от Microsoft, так как она предоставляет наиболее широкие возможности для программирования приложений под управлением операционной системы Windows.

Среда VS относится к классу инструментов ускоренной разработки программ и позволяет быстро и эффективно разработать приложение. Это достигается за счет двух характерных свойств VS: визуального конструирования форм и широкого использования библиотеки визуальных компонентов. Система программирования VS рассчитана на программирование различных приложений и предоставляет большое количество компонентов для этого.

Визуальное конструирование форм избавляет программиста от многих аспектов разработки интерфейса программы, так как VS автоматически готовит необходимые программные заготовки и соответствующий файл ресурсов.

Ключевой особенностью VS является возможность быстро создавать качественные приложения, не создавая при этом код большого размера. Таким образом, ускоряется работа по созданию приложения и в короткие сроки можно создать достаточно серьезные приложения.

Важной особенностью Visual Studio является возможность быстро создавать качественные приложения, не создавая при этом код большого размера. Таким образом, ускоряется работа по созданию приложения и в короткие сроки можно создать достаточно серьезные приложения.

Обёртка библиотеки является промежуточным слоем между прикладной программой и другой библиотекой или интерфейсом программирования приложений (API).

Целью написания обёртки библиотеки может быть обеспечение работоспособности библиотеки (API) в каком-либо (чаще скриптовом) языке, в котором прямой вызов функций этой библиотеки API затруднителен или невозможен.

Другой целью может быть обеспечение дополнительного удобства для прикладного программиста, например адаптация библиотеки к объектно-ориентированному стилю программирования, компенсация неудобного дизайна библиотеки и т.п.

Существуют также кроссплатформенные обёртки библиотек, скрывающие реализацию для разных операционных систем, например wxWidgets.



## 2.2. Алгоритмы функций

Функции приложения:

1. добавить запись в базу данных;
2. удалить запись из базы данных;
3. установить позицию в ключе;
4. получить запись по значению ключа;
5. сохранить запись по значению ключа;
6. получить следующую запись в последовательности ключа;
7. prev – получить предыдущую запись в последовательности ключа.

### 3. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Для использования библиотеки необходимо подключить заголовочный файл, который находится рядом с библиотекой. Далее в своем проекте необходимо вызывать функции из подключённого заголовочного файла.

Примерный вариант использования библиотеки представлен ниже (Рис 1.).

```
#include <iostream>
#include "MathLibrary.h"

using namespace Database;
int main()
{
    db db;
    db.init();
    std::cout << "add elem 1-9" << std::endl;
    db.add("1");
    db.add("2");
    db.add("3");
    db.add("4");
    db.add("5");
    db.add("6");
    db.add("7");
    db.add("8");
    db.add("9");
    db.view();
    std::cout << "delete first elem" << std::endl;
    db.del();
    db.view();
    std::cout << "delete elem pos 4" << std::endl;
    db.set(4);
    db.del();
    db.view();
    std::cout << "next elem after pos 4" << std::endl;
    db.next();
    std::cout << "next elem after pos 5" << std::endl;
    db.next();
    db.set(3);
    std::cout << "prev elem before pos 3" << std::endl;
    db.prev();
    std::cout << "prev elem before pos 2" << std::endl;
    db.prev();
    std::cout << "set pos 5 and put elem 10" << std::endl;
    db.set(5);
    db.put("10");
    db.view();
    db.set(1);

    system("PAUSE");

    return 0;
}
```

Рис. 1. Пример кода для использования библиотеки

- Функция add – добавляет запись в БД;
- Функция delete – удаляет запись из базы данных;
- Функция set – устанавливает позицию в ключе;
- Функция get – получает запись по значению ключа;
- Функция put – сохраняет запись по значению ключа;
- Функция next/prev – получает следующую/предыдущую запись в последовательности ключа;
- Функция view выводит на экран таблицу базы данных.

## **ЗАКЛЮЧЕНИЕ**

В результате выполнения курсовой работы было разработано мобильное приложение для платформы Android, позволяющее осуществлять обмен сообщениями в режиме реального времени. В качестве хранилища информации была использована облачная база данных, сконфигурировано и осуществлено подключение к API для интеграции. В приложении реализованы функции загрузки мультимедиа-файлов, а именно изображений в облачное хранилище, совместно с процессом сохранения нового сообщения в облачную базу данных. Производится обработка исключительных ситуаций, используются функции сортировки для сообщений таким образом, чтобы новые сообщения отображались в начале списка. В процессе работы программы активно используются возможности мобильного устройства к подключению к сети Интернет для загрузки и обновления информации, а также возможность отключения и подключения к сети в рамках мобильного приложения, посредством использования разрешений на доступ к системным функциям.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1 С. Макконелл «Совершенный код» / —Москва: Русская редакция, 2013.
- 2 Официальная документация Berkeley DB [Электронный ресурс]. — Режим доступа: [https://docs.oracle.com/cd/E17276\\_01/html/programmer\\_reference/index.html](https://docs.oracle.com/cd/E17276_01/html/programmer_reference/index.html). — Дата доступа: 22.05.2020
- 3 The C Book [Электронный ресурс]. — Режим доступа: [https://publications.gbdirect.co.uk/c\\_book/](https://publications.gbdirect.co.uk/c_book/). — Дата доступа: 22.05.2020