**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

**КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ**

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

до курсової роботи

з дисципліни «Програмування»

Виконав: студент групи іт-23

Літвінов К. О.

Варіант: 15

Перевірив: Курінна В. В.

2023

**Зміст**

[1 Загальна постановка задачі 3](#_Toc135932213)

[ЗАВДАННЯ НА КУРСОВИЙ ПРОЕКТ 3](#_Toc135932214)

[2.Опис підпрограм користувача 5](#_Toc135932215)

[3.Тексти розроблених модулів 7](#_Toc135932216)

[4 Текст основної програми 13](#_Toc135932217)

[5 Контрольний приклад 14](#_Toc135932218)

[Список використаних джерел 15](#_Toc135932219)

1 Загальна постановка задачі

**ЗАВДАННЯ НА КУРСОВИЙ ПРОЕКТ**

Взявши за основу свій варіант структури запису, реалізувати роботу з найпростішою базою даних, використовуючи двійкові файли (файли структур).

У програмі реалізувати створення нової бази даних і додавання записів в уже створену. Вхідні дані повинні вводитися з перевіркою на область допустимих значень. Усі дії користувача повинні контролюватися і супроводжуватися обдуманими повідомленнями.

У програмі повинні бути передбачені відповідні функції, що коректно обробляють вхідні дані, виконують читання-запис файлів бази даних. Інформація повинна запам’ятовуватися у файлі і бути доступною для наступної (багаторазової) обробки.

Реалізувати у відповідності до свого варіанту:

запит (обчислення окремого значення);

пошук інформації за ключем.

Увага!! Це два окремі завдання.

Передбачити можливість виведення всіх записів бази даних на екран чи в окремий текстовий файл (за бажанням користувача).

Уся обробка бази даних повинна відбуватися шляхом вибору користувачем відповідного пункту з меню.

Таблиця №1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***№***  ***Вар.*** | | ***Ключ для пошуку*** | ***Ключ для пошуку*** | ***запит*** |
| 15 | Рисунок 3 | | за типом будівлі | Визначити невиплачену суму всіх клієнтів. | Визначити  невиплачену суму всіх  клієнтів |

Будівля

Клієнт

Тип будівлі

Фірма-підрядчик

Вартість

% закредит

сума вкладу

Невипла- чена сума

**Схема запису «Будівля»**

2.Опис підпрограм користувача

Наводиться опис всіх  функцій (тип результату ,ідентифікатор, список параметрів), що були розроблені в ході виконання курсової роботи.  Для кожної функції зазначити суть її роботи.  Опис можна оформити у вигляді списку або таблиці, на розсуд студента

Таблица №3 **Таблиця позначень**

|  |  |  |
| --- | --- | --- |
| **Позначення** | **Пояснення** | |
| KYRS\_Litvinov\_Rel.cpp | | |
| int main() | | |
| controlMenu(); | | Виклик функції "controlMenu()", яка містить меню керування програмою. |
| struk.h | | |
| const int N = 20; | | Оголошення константи "N" зі значенням 20. |
| struct Customer {  float creditInterest;  float depositAmount;  float creditBalance;  }; | | Оголошення структури "Customer", яка містить поля для даних про клієнта. |
| struct Building {  Customer cust;  int buildingType;  float price;  char subcontra[N];  }; | | Оголошення структури Building, яка містить поля для будівлі та пов'язаного з нею клієнта. |
| functions.h | | |
| void createDatabaseFile() | | |
| createDatabaseFile(): Ця функція створює текстовий файл "database.txt", який буде використовуватися для запису даних про будівлі | | |
| void createDatabase() | | |
| ofstream file("database.bin", ios::binary); | | створюється об'єкт file класу ofstream для запису даних у файл з ім'ям "database.bin" |
| createDatabase(): Ця функція створює бінарний файл "database.bin", який буде використовуватися для зберігання даних про будівлі. | | |
| void addRecord() | | |
| Building record; | | створюється змінна record типу Building. Building є визначеною структурою даних у файлі struk.h. |
| ofstream file("database.bin", ios::binary | ios::app); | | створюється об'єкт file класу ofstream для запису даних у файл з ім'ям "database.bin" |
| addRecord(): Ця функція дозволяє додавати нові записи про будівлі до бази даних. | | |
| void query() | | |
| float totalUnpaidAmount = 0.0; | | Змінна totalUnpaidAmount використовується для зберігання загальної суми невиплачених коштів за кредитом |
| Building record; | | створюється змінна record типу Building. |
| ifstream file("database.bin", ios::binary); | | створюється об'єкт file класу ifstream для читання даних із файлу з ім'ям "database.bin". |
| query(): Ця функція виконує запит до бази даних та обчислює загальну невиплачену суму всіх клієнтів за кредитом. | | |
| void searchByKey() | | |
| int key; | | оголошується змінна key типу int. |
| Building record; | | Змінна record є об'єктом структури Building, яка містить інформацію про будівлю та клієнта. |
| bool found = false; | | оголошується змінна found типу bool та ініціалізується значенням false. |
| searchByKey(): Ця функція дозволяє шукати записи за певним ключем (тип будівлі) в базі даних. | | |
| void writeRecordToDatabaseFile(const Building& record) | | |
| writeRecordToDatabaseFile(): Ця функція записує дані про будівлю до текстового файлу "database.txt". Вона отримує об'єкт будівлі в якості параметра і відкриває файл для запису. Дані записуються у вигляді текстового формату. | | |
| void displayDatabase() | | |
| Building record; | | Змінна record є об'єктом структури Building, яка містить інформацію про будівлю та клієнта. |
| void controlMenu() | | |
| char choice; | | використовується для зберігання вибору користувача в контексті програми. |

3.Тексти розроблених модулів

1. struk.h

const int N = 20;

struct Customer {

float creditInterest;

float depositAmount;

float creditBalance;

};

struct Building {

Customer cust;

int buildingType;

float price;

char subcontra[N];

};

----------------------------------------------------------------------------------------------------

2. funcions.h

#pragma once

#include <struk.h>

using namespace std;

// базу в файл

void createDatabaseFile() {

ofstream file("database.txt");

if (file.is\_open()) {

cout << " Файл бази даних успішно створено." << endl;

file.close();

}

else {

cout << " Помилка при відкритті файлу бази даних." << endl;

}

}

void createDatabase() {

ofstream file("database.bin", ios::binary);

if (file.is\_open()) {

cout << "Створено нову базу даних." << endl;

file.close();

}

else {

cout << "Помилка при створенні бази даних." << endl;

}

}

void addRecord() {

Building record;

cout << "Введіть дані для нового запису:" << endl;

cout << "Будівля:(вводяться символи )";

cin.ignore();

cin.getline(record.subcontra, N);

cout << "Тип будівлі:(приклад<1,2>)";

cin >> record.buildingType;

cout << "Ціна: ";

cin >> record.price;

cout << "Відсоткова ставка по кредиту: ";

cin >> record.cust.creditInterest;

cout << "Сума внеску: ";

cin >> record.cust.depositAmount;

// Перевірка, чи введені від'ємні числа

if (record.buildingType < 0 || record.price < 0 || record.cust.creditInterest < 0 || record.cust.depositAmount < 0) {

cout << "Помилка: Від'ємні числа не допускаються." << endl;

return;

}

// Розрахунок невиплаченої суми

if (record.cust.depositAmount < record.price) {

record.cust.creditBalance = record.price - record.cust.depositAmount;

}

else {

record.cust.creditBalance = 0.0;

}

ofstream file("database.bin", ios::binary | ios::app);

if (file.is\_open()) {

file.write(reinterpret\_cast<const char\*>(&record), sizeof(Building));

cout << "Запис успішно додано до бази даних." << endl;

file.close();

}

else {

cout << "Помилка при відкритті бази даних." << endl;

}

}void query() {

float totalUnpaidAmount = 0.0;

Building record;

ifstream file("database.bin", ios::binary);

if (file.is\_open()) {

while (file.read(reinterpret\_cast<char\*>(&record), sizeof(Building))) {

if (record.cust.creditBalance > 0.0) {

totalUnpaidAmount += record.cust.creditBalance \* record.cust.creditInterest;

}

}

cout << "Загальна невиплачена сума всіх клієнтів по кредиту: " << totalUnpaidAmount << endl;

file.close();

}

else {

cout << "Помилка при відкритті бази даних." << endl;

}

}

void searchByKey() {

int key;

cout << "Введіть ключ для пошуку за типом будівлі: ";

cin >> key;

Building record;

ifstream file("database.bin", ios::binary);

if (file.is\_open()) {

bool found = false;

while (file.read(reinterpret\_cast<char\*>(&record), sizeof(Building))) {

if (record.buildingType == key && record.cust.depositAmount < record.price) {

cout << "Знайдено запис:" << endl;

cout << "Будівля: " << record.subcontra << endl;

cout << "Тип будівлі: " << record.buildingType << endl;

cout << "Вартість: " << record.price << endl;

cout << "Відсоток за кредит: " << record.cust.creditInterest << endl;

cout << "Сума вкладу: " << record.cust.depositAmount << endl;

cout << endl;

cout << "Невиплачена сума: " << record.cust.creditBalance << endl;

cout << endl;

found = true;

}

}

if (!found) {

cout << "Записів з таким ключем не знайдено." << endl;

}

file.close();

}

else {

cout << "Помилка при відкритті бази даних." << endl;

}

}

void writeRecordToDatabaseFile(const Building& record) {

ofstream file("database.txt", ios::app);

if (file.is\_open()) {

file << "Будівля: " << record.subcontra << endl;

file << "Тип будівлі: " << record.buildingType << endl;

file << "Вартість: " << record.price << endl;

file << "Відсоток за кредит: " << record.cust.creditInterest << endl;

file << "Сума вкладу: " << record.cust.depositAmount << endl;

file << "Невиплачена сума: " << record.cust.creditBalance << endl;

file << endl;

file.close();

}

else {

cout << " Помилка при відкритті файлу бази даних." << endl;

}

}

void displayDatabase() {

Building record;

ifstream file("database.bin", ios::binary);

if (file.is\_open()) {

cout << "Записи бази даних:" << endl;

// Заголовок таблиці

cout << "-----------------------------------------------------------------------------------------------------------------" << endl;

cout << "Будівля\t\tТип будівлі\tВартість\tВідсоток за кредит\tСума вкладу\tНевиплачена сума" << endl;

cout << "-----------------------------------------------------------------------------------------------------------------" << endl;

while (file.read(reinterpret\_cast<char\*>(&record), sizeof(Building))) {

// Вивід даних у вигляді таблиці

cout << record.subcontra << "\t\t"

<< record.buildingType << "\t\t"

<< record.price << "\t\t\t"

<< record.cust.creditInterest << "\t\t\t"

<< record.cust.depositAmount << "\t\t\t"

<< record.cust.creditBalance << endl;

// Записать данные в файл

writeRecordToDatabaseFile(record);

}

cout << "-----------------------------------------------------------------------------------------------------------------" << endl;

file.close();

}

else {

cout << "Помилка при відкритті бази даних." << endl;

}

}

void controlMenu() {

char choice;

do {

cout << "Меню:" << endl;

cout << "a. Створити нову базу даних" << endl;

cout << "b. Додати запис" << endl;

cout << "c. Виконати запит" << endl;

cout << "d. Пошук за ключем" << endl;

cout << "f. Вивести базу даних" << endl;

cout << "g. Створення txt файлу " << endl;

cout << "q. Вийти з програми" << endl;

cout << "Ваш вибір: ";

cin >> choice;

switch (choice) {

case 'a':

createDatabase();

break;

case 'b':

addRecord();

break;

case 'c':

query();

break;

case 'd':

searchByKey();

break;

case 'f':

displayDatabase();

break;

case 'g':

createDatabaseFile();

break;

case 'q':

cout << "Програма завершує роботу." << endl;

break;

default:

cout << "Невірний вибір. Спробуйте ще раз." << endl;

break;

}

} while (choice != 'q');

}

4 Текст основної програми

1. KYRS\_Litvinov\_Rel.cpp

#include <iostream>

#include <fstream>

#include <string>

#include <functions.h>

#include <Windows.h>

#include <iomanip>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(LC\_ALL, " Ukrainian");

controlMenu();

return 1;

}

5 Контрольний приклад

Зображення, що містить текст, електроніка, знімок екрана, комп’ютер

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, чорний

Автоматично згенерований опис

Зображення, що містить текст, Мультимедійне програмне забезпечення, знімок екрана

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис

**Зображення, що містить текст, знімок екрана

Автоматично згенерований опис**

**Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис**

**Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис**

Список використаних джерел

1. <https://ru.stackoverflow.com>

2. <https://mix.sumdu.edu.ua/study/course/7898>

3. <https://github.com>

4. <https://learn.microsoft.com/uk-ua/cpp/windows/latest-supported-vc-redist?view=msvc-170>

5. <https://uk.wikibooks.org>

6. <https://translate.google.com/?hl=ru&sl=en&tl=uk&op=translate>