|  |  |  |
| --- | --- | --- |
|  |  |  |
| Class Menus; | Menu | To create menus; |
|  |  |  |
| Class Clients | Client | To create client databases; |
|  |  |  |
|  |  |  |
|  |  |  |

8. client.csv 파일, 불러오기; 추가하기;

#include <iostream>

#include <fstream>

#include <sstream>

#include <vector>

#include <string>

using namespace std;

int main() {

vector<vector<string>> data;

ifstream infile("client.csv");

string line;

// 1. 기존 파일 데이터를 읽어 벡터로 저장

while (getline(infile, line)) {

stringstream ss(line);

string cell;

vector<string> row;

while (getline(ss, cell, ',')) {

row.push\_back(cell);

}

data.push\_back(row);

}

infile.close();

// 2. 사용자로부터 입력받아 새 데이터 추가

string name, age, email;

cout << "이름 입력: ";

getline(cin, name);

cout << "나이 입력: ";

getline(cin, age);

cout << "이메일 입력: ";

getline(cin, email);

vector<string> new\_row = { name, age, email };

data.push\_back(new\_row);

// 3. 전체 데이터를 파일에 다시 저장

ofstream outfile("client.csv");

for (const auto& row : data) {

for (size\_t i = 0; i < row.size(); ++i) {

outfile << row[i];

if (i != row.size() - 1) outfile << ",";

}

outfile << endl;

}

outfile.close();

cout << "새로운 사용자가 추가되었습니다." << endl;

return 0;

}

7. client.csv 파일, 만들기; id, name, address, phone, hobby 추가;

#include <iostream>

#include <fstream>

#include <vector>

#include <string>

#include <cstdlib>

#include <ctime>

using namespace std;

string generateKoreanName() {

vector<string> lastNames = { "김", "이", "박", "최", "정", "조", "강", "윤", "장", "임", "한", "오", "서", "신" };

vector<string> firstNames = { "민수", "서연", "지후", "지민", "하은", "도윤", "예은", "수빈", "현우", "지원",

"시우", "유진", "하윤", "건우", "예린", "소율", "지안", "태윤", "은우", "서준" };

return lastNames[rand() % lastNames.size()] + firstNames[rand() % firstNames.size()];

}

string generateAddress() {

vector<string> cities = {

"서울", "부산", "대구", "인천", "광주", "대전", "울산", "세종",

"수원", "고양", "성남", "용인", "청주", "천안", "전주", "포항",

"창원", "김해", "평택", "안산", "안양", "남양주", "화성", "제주"

};

vector<string> districts = {

"중구", "서구", "남구", "북구", "동구", "강남구", "강서구", "송파구", "성동구", "성북구",

"해운대구", "수성구", "달서구", "유성구", "팔달구", "덕양구", "일산동구", "분당구", "기흥구",

"흥덕구", "서북구", "덕진구", "진해구", "장유동", "신제주"

};

return cities[rand() % cities.size()] + " " + districts[rand() % districts.size()] + " 123-45";

}

string generatePhoneNumber() {

int second = rand() % 9000 + 1000;

int third = rand() % 9000 + 1000;

return "010-" + to\_string(second) + "-" + to\_string(third);

}

string generateHobby() {

vector<string> hobbies = {

"독서", "영화감상", "운동", "게임", "요리", "여행", "사진촬영", "음악감상", "등산", "수영",

"뜨개질", "보드게임", "유튜브 보기", "자전거 타기", "드로잉", "일기 쓰기", "웹서핑", "카페 투어"

};

return hobbies[rand() % hobbies.size()];

}

int main() {

srand(static\_cast<unsigned int>(time(nullptr)));

ofstream file("client.csv");

if (!file.is\_open()) {

cerr << "파일 열기 실패!" << endl;

return 1;

}

file << "id,name,age,address,cell phone,hobby\n";

for (int i = 1; i <= 20000; ++i) {

string name = generateKoreanName();

int age = rand() % 60 + 20;

string address = generateAddress();

string phone = generatePhoneNumber();

string hobby = generateHobby();

file << i << "," << name << "," << age << "," << address << "," << phone << "," << hobby << "\n";

}

file.close();

cout << "✅ client.csv 파일이 20,000명 데이터로 생성되었습니다!" << endl;

return 0;

}

6. 테스트용 .csv 파일로 저장하기;

#include <iostream>

#include <fstream>

using namespace std;

int main() {

ofstream file("testdb.csv");

file << "number, id, name, phone\n";

file << 1 << "," << 2323 << "," << "cherlhee" << "," << "010-3333-4444" << endl;

file << 2 << "," << 1111 << "," << "jcradar" << "," << "010-1111-2222" << endl;

file.close();

return 0;

}

5. csv 파일로 저장하기;

#include <iostream>

#include <fstream>

#include <sstream>

using namespace std;

class Clients {

int id;

string name;

string address;

string tel;

public:

Clients();

void insertClient(int newId, string newName, string newTel, string newAddress);

void displayClient();

void saveToCSV(ofstream& file);

};

Clients::Clients() {}

void Clients::insertClient(int newId, string newName, string newTel, string newAddress) {

this->id = newId;

this->name = newName;

this->tel = newTel;

this->address = newAddress;

}

void Clients::displayClient() {

cout << "Client ID: " << id << ", Name: " << name

<< ", Tel: " << tel << ", Address: " << address << endl;

}

// Save client details to a CSV file (append mode)

void Clients::saveToCSV(ofstream& file) {

file << id << "," << name << "," << tel << "," << address << endl;

}

bool fileExists(const string& filename) {

ifstream file(filename);

return file.good();

}

int main() {

int id;

string name, tel, address;

const int maxnumClient = 2;

Clients client[maxnumClient];

string filename = "clients.csv";

// Open the file in append mode (ios::app)

ofstream outFile;

bool isNewFile = !fileExists(filename); // Check if the file exists

outFile.open(filename, ios::app);

if (!outFile) {

cerr << "Error opening file for writing!" << endl;

return 1;

}

// If it's a new file, add a CSV header

if (isNewFile) {

outFile << "ID,Name,Tel,Address\n";

}

// Input and append to CSV

for (int i = 0; i < maxnumClient; i++) {

cout << "Plz input ID: ";

cin >> id;

cout << "Plz input name: ";

cin.ignore();

getline(cin, name);

cout << "Plz input tel: ";

getline(cin, tel);

cout << "Plz input address: ";

getline(cin, address);

client[i].insertClient(id, name, tel, address);

client[i].saveToCSV(outFile); // Append new client data

}

outFile.close();

cout << "\n ------ Client Data Appended to CSV ------ \n";

return 0;

}

4. 텍스트 파일로 저장하기;

#include <iostream>

#include <fstream>

using namespace std;

int main() {

ofstream fout;

fout.open("song.txt");

if (!fout) {

cout << "file opening of song.txt failed;";

return 0;

}

int age = 21;

char singer[] = "kim";

char song[] = "yesterday";

fout << age << '\n';

fout << singer << endl;

fout << song << endl;

fout.close();

}

3. 배열로 고객정보 추가 클래스, 만들기;

|  |
| --- |
| ------ hello, client display ------  client id; 1, name: dsfsdclient tel; sdfsd, address; sdfsd  client id; 2, name: werwclient tel; werwe, address; werwe |

#include <iostream>

using namespace std;

class Clients {

int id;

string name;

string address;

string tel;

public:

Clients();

void insertClient(int newId, string newName, string newTel, string newAddress);

void displayClient();

};

Clients::Clients() {}

void Clients::insertClient(int newId, string newName, string newTel, string newAddress) {

this->id = newId;

this->name = newName;

this->tel = newTel;

this->address = newAddress;

}

void Clients::displayClient() {

cout << "client id; " << id << ", name: " << name;

cout << "client tel; " << tel << ", address; " << address << endl;

}

int main() {

int id;

string name, tel, address;

const int maxnumClient = 2;

Clients client[maxnumClient];

for (int i = 0; i < maxnumClient; i++) {

cout << "plz input id; ";

cin >> id;

cout << "plz input name; ";

cin >> name;

cout << "plz input tel; ";

cin >> tel;

cout << "plz input address; ";

cin >> address;

client[i].insertClient(id, name, tel, address);

}

cout << "\n ------ hello, client display ------ \n";

for (int i = 0; i < maxnumClient; i++) {

client[i].displayClient();

}

return 0;

}

2. 고객정보 추가 클래스, 만들기;

#include <iostream>

using namespace std;

class Clients {

int id;

string name;

string address;

string phone;

public:

Clients();

void insertClient(int newId, string newName);

void displayClient();

};

Clients::Clients() {}

void Clients::insertClient(int newId, string newName) {

this->id = newId;

this->name = newName;

}

void Clients::displayClient() {

cout << "client id; " << id << ", name: " << name << endl;

}

int main() {

Clients client;

int id;

string name;

cout << "plz input id; ";

cin >> id ;

cout << "plz input name; ";

cin >> name;

client.insertClient(id, name);

cout << "hello, project 2nd;" << endl;

client.displayClient();

return 0;

}

1. 메뉴만들기

#include <iostream>

using namespace std;

class Menus{

public:

Menus();

int display();

};

Menus::Menus() {}

int Menus::display() {

int selection;

cout << "menu" << endl;

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

cout << "1. insert clients" << endl;

cout << "2. delete clients" << endl;

cout << "3. display clients" << endl;

cout << "4. list clients" << endl;

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

cout << "please input selections (1~4); " ;

cin >> selection;

return selection;

}

int main() {

Menus menu;

menu.display();

return 0;

}