**MFC 고객관리 프로그램 과제2 보고서**

경영학부

학번: 2020110016

이름: 오주혁

1. 개발 환경 및 목적

* **개발 환경:**
  + Visual Studio 2022
  + MFC 대화상자 기반 애플리케이션
  + C++ (객체지향, 파일 입출력, 메모리 관리) 활용1
* **목적:**
  + 고객 정보를 CSV 파일로 관리
  + GUI에서 고객 추가/삭제/상세조회 기능 구현
  + 리스트 컨트롤에 고객 이름만 표시

2. 주요 기능 및 동작 설명

2.1. 고객 정보 관리

* **CSV 파일 입출력:**
  + 프로그램 시작 시 customers.csv 파일에서 고객 데이터 불러오기
  + 고객 추가/삭제 시 CSV 파일 자동 업데이트2
* **데이터 포맷:**
  + 한 줄: ID,이름,생년월일,주소,메모
  + 예: C001,홍길동,19900101,동원로,VIP

2.2. 사용자 인터페이스

* **입력란:**
  + ID, 이름, 생년월일, 주소, 메모 입력 (CEdit 컨트롤)
* **버튼:**
  + 고객추가, 고객삭제, 상세조회 (CButton 컨트롤)
* **고객 리스트:**
  + 리스트 컨트롤(CListCtrl)에 고객 이름만 표시
  + 선택 후 삭제/상세조회 가능

2.3. 세부 동작

* **프로그램 시작 시:**
  + customers.csv 파일에서 모든 고객의 이름을 리스트에 표시
* **고객 추가:**
  + 입력란에 정보 입력 후 "고객추가" 클릭
  + 리스트에 이름 추가, CSV 파일에 전체 정보 저장
* **고객 삭제:**
  + 리스트에서 이름 선택 후 "고객삭제" 클릭
  + 리스트에서 삭제, CSV 파일에서 해당 고객 정보 삭제
* **상세조회:**
  + 리스트에서 이름 선택 후 "상세조회" 클릭
  + 해당 고객의 모든 정보가 입력란에 자동 표시

3. 코드 구조 및 주요 구현

3.1. 컨트롤 변수 선언 (assignment2Dlg.h)

cpp

CListCtrl m\_listCustomers;

CEdit m\_editID, m\_editName, m\_editBirth, m\_editAddress, m\_editMemo;

CButton m\_btnAdd, m\_btnDelete, m\_btnDetail;

HICON m\_hIcon;

**void** LoadCSVData();

3.2. CSV 파일 불러오기

cpp

**void** Cassignment2Dlg::LoadCSVData()

{

CStdioFile file;

**if** (!file.Open(\_T("customers.csv"), CFile::modeRead | CFile::typeText))

**return**;

m\_listCustomers.DeleteAllItems();

CString line; **int** nItem = 0;

**while** (file.ReadString(line))

{

CStringArray fields;

**int** start = 0;

**for** (**int** i = 0; i < 4; i++) {

**int** end = line.Find(',', start);

**if** (end == -1) end = line.GetLength();

fields.Add(line.Mid(start, end - start));

start = end + 1;

}

fields.Add(line.Mid(start));

m\_listCustomers.InsertItem(nItem, fields[1]); *// 이름만 리스트에 표시*

nItem++;

}

file.Close();

}

3.3. 고객 추가

cpp

**void** Cassignment2Dlg::OnBnClickedButton1()

{

CString id, name, birth, address, memo;

m\_editID.GetWindowText(id);

m\_editName.GetWindowText(name);

m\_editBirth.GetWindowText(birth);

m\_editAddress.GetWindowText(address);

m\_editMemo.GetWindowText(memo);

**if** (id.IsEmpty() || name.IsEmpty()) {

MessageBox(\_T("ID와 이름은 필수입니다."), \_T("입력 오류"), MB\_ICONWARNING);

**return**;

}

m\_listCustomers.InsertItem(0, name);

m\_editID.SetWindowText(\_T(""));

m\_editName.SetWindowText(\_T(""));

m\_editBirth.SetWindowText(\_T(""));

m\_editAddress.SetWindowText(\_T(""));

m\_editMemo.SetWindowText(\_T(""));

CStdioFile file;

**if** (file.Open(\_T("customers.csv"), CFile::modeWrite | CFile::modeNoTruncate | CFile::typeText))

{

file.SeekToEnd();

CString newLine;

newLine.Format(\_T("%s,%s,%s,%s,%s\n"), id, name, birth, address, memo);

file.WriteString(newLine);

file.Close();

}

}

3.4. 고객 삭제 및 상세조회

* **삭제:** 리스트에서 선택한 이름을 기준으로 CSV 전체를 다시 저장
* **상세조회:** 선택한 이름으로 CSV에서 전체 정보 찾아 입력란에 표시

4. 실행화면 예시

* **초기화면:** 고객 이름 리스트만 표시
* **입력/추가:** 입력란에 정보 입력 → "고객추가" 클릭
* **삭제/상세조회:** 리스트에서 이름 선택 → 해당 버튼 클릭

5. 느낀 점 및 개선사항

* **파일 입출력과 데이터 동기화의 중요성**을 체험
* **리스트 컨트롤 사용법, MFC 이벤트 처리**에 대한 이해도 향상
* 추후에는 **중복 고객 방지, 검색/정렬 기능, CSV 헤더 추가** 등 확장 가능