

수업명 : C++ 프로그래밍 및 실습

# 대학 생활 가계부

진척 보고서 #02

제출일자: 2023-12-15

제출자명:신승환

제출자학번:192280

## **1. 프로젝트 목표**

### **1) 배경 및 필요성**

대학교에 복학하면서, 평소 씀씀이가 늘어남에 따라 지출과 수입의 흐름을 파악할 필요성을 느꼈고 이를 정리하여 월별 합계를 파악한다면 경제적으로 힘든 옷끈 상황이 오는 것을 미리 방지할 수 있다.

### **2) 프로젝트 목표**

자신이 월에 얼마 정도를 사용했는지 파악하여 스스로 지출에 대해 관리를 할 수 있도록 만드는 것이 목표

### **3) 차별점**

기존 프로그램들은 그냥 언제 얼마를 사용하였는지 기록하는 프로그램이었다면 이번에 만드는 프로그램은 어떤 일로 인해 수입이 생겼는지 혹은 어떤 것에 지출을 하였는지 설명을 기록하여 추후 확인하였을 때 훨씬 좋을 것이다. 추가적으로 일별로 기록되는 표를 만들어 보기에 편하게 만들 예정

## **2. 기능 계획**

### **1) 기능 1 (가계부 프로그램에 어떤 것들이 있는지 출력)**

- 설명) : 사용자가 프로그램에서 선택할 수 있는 내역 추가, 조회, 월별 합계 조회, 내역 수정, 내역 삭제, 내역 보기(표를 출력해 정리된 내역 보기)

#### **(1) 세부 기능 1**

- 항목들을 출력하여 사용자에게 입력 받기

### **2) 기능 2 (수입 / 지출 내역 추가)**

- 설명 : 항목 중에서 내역 추가를 선택하면 1. 수입 또는 지출 2.금액 3.설명 4.날짜

(YYYY-MM-DD)를 입력받는다.

-(1) 세부기능 1

월 테이블(표)를 생성한다.

-(2) 세부기능 2

위에서 내역 추가된 내용 들을 월 테이블(표)에 추가한다

### **3)기능 3 (내역 목록 조회)**

- 설명 : 내역 조회를 선택하면 년도와 월 , 일을 출력하여 선택받고 그 날짜의 내역 목록을 조회

### **4) 기능 4 (월별 합계 조회)**

- 설명 : 월별 합계 조회를 선택하면 년도와 월을 출력하여 선택받고 그에 해당하는 월의 수입과 지출의 합인 합계를 출력

### **5)기능 5 (특정 내역 수정 및 삭제)**

-설명 : 내역 수정 및 삭제를 선택하면 년도와 월 일을 출력하여 선택받고 내역에 대해 수정 및 삭제를 할 수 있도록 해준다

### **6)기능 6 (월 테이블 출력)**

-설명 : 내역이 저장된 월 테이블을 년도와 월을 입력 받아 출력

## **3. 진척사항**

### **1) 기능 구현**

(1) 기능 1 (가계부 프로그램에 어떤 것들이 있는지 출력)

- 입출력 : 7 가지의 선택지를 출력한다. 그리고 사용자에게 원하는 숫자를 입력받는다.
- 설명 : 메인메뉴라는 함수를 이용하여 7 가지의 출력할 내용을 함수화하여 사용하고 메인함수에서 do-while 문과 switch 문을 이용하여 각 기능에 대한 함수를 호출한다.
- 적용된 배운 내용 : 함수, 반복문, 조건문

- 코드 스크린샷:

```
#include <iostream>
#include <vector>
#include <string>
using namespace std;

// 메인 메뉴 출력 및 사용자 선택
int mainMenu() {
    int choice;
    cout << "1. 내역 추가\n";
    cout << "2. 내역 조회\n";
    cout << "3. 월별 합계 조회\n";
    cout << "4. 내역 수정\n";
    cout << "5. 내역 삭제\n";
    cout << "6. 내역 보기\n";
    cout << "7. 종료\n";
    cout << "선택: ";
    cin >> choice;
    return choice;
}

// 메인 함수 반복문
int main() {
    int choice;
    do {
        choice = mainMenu();
        switch (choice) {
            // 각 기능에 대한 함수 호출
            case 1:
                addEntry();
                break;
            case 2:
                viewEntry();
                break;
            case 3:
                viewMonthTotal();
                break;
            case 4:
                editEntry();
                break;
        }
    } while (choice != 7);
}
```

## (2) 기능 2 (수입 / 지출 내역 추가)

- 입출력 : 1 을 입력하면 내역추가를 위해 수입,지출을 물어보고 금액과 설명, 해당 날짜에 대해 입력받고 그 값을 저장한다.
- 설명 : 내역추가 에 관한 함수를 만들고 벡터와 push\_back 을 이용하여 입력된 값을 해당 데이터에 추가해놓는다.(추후 사용할때 다시 이용)
- 적용된 배운 내용 : 함수, 벡터, push\_back
- 코드 스크린샷

```
//각 데이터 유형을 위해 벡터 선언
vector<string> types;
vector<double> amounts;
vector<string> descriptions;
vector<string> dates;
```

```
void addEntry() {
    string type;
    double amount;
    string description;
    string date;

    //입력 받은 것에 대해 해당하는 vector에 추가한다.
    //주주 조회, 수정, 삭제 등에 사용할 때 사용된다.

    cout << "1. 수입 또는 지출: ";
    cin >> type;
    types.push_back(type);

    cout << "2. 금액: ";
    cin >> amount;
    amounts.push_back(amount);

    // 입력 버퍼 클리어
    getchar();

    cout << "3. 설명: ";
    getline(cin, description);
    descriptions.push_back(description);

    cout << "4. 날짜 (YYYY-MM-DD): ";
    cin >> date;
    dates.push_back(date);

    cout << "내역이 추가되었습니다.\n";
}
```

### (3) 기능 3 (내역 목록 조회)

- 입출력 : 2 번을 입력하면 그에 해당하는 함수를 출력하여 년 월 일 을 입력받고 값을 해당하는 변수에 저장
- 설명 : 입력받고 그 값에 대해서 반복문과 조건문 그리고 자릿수 추출을 이용해 값을 뽑아낸다.
- 적용된 배운 내용 : 반복문, 조건문, 벡터
- 코드 스크린샷

```
void viewEntry() {
    string year;
    string month;
    string day;

    cout << "년도 입력: ";
    cin >> year;
    cout << "월 입력: ";
    cin >> month;
    cout << "일 입력: ";
    cin >> day;

    //벡터에 관해 반복문을 하기에 size_t, dates.size()로 적어서 반복문을 사용
    //날짜 문자열에서 해당하는 자릿수를 추출하여 년, 월, 일을 알아낸다.
    for (size_t i = 0; i < dates.size(); ++i) {
        if (dates[i].substr(0, 4) == year && dates[i].substr(5, 2) == month) {
            cout << dates[i] << " : " << types[i] << " " << amounts[i] << " " << descriptions[i] << "\n";
        }
    }
}
```

#### 4) 기능 4 (월별 합계 조회)

- 입출력 : 특정 년도와 월을 입력하면 해당하는 년도의 월에 해당하는 수입과 지출의 합계가 나온다.
- 설명 : 사용자가 입력한 월의 수입과 지출을 파악할 수 있도록 날짜를 기반으로 필터링하여 조건에 맞는 항목의 금액을 합산하는 방식
- 적용된 배운 내용 : 반복문, 조건문, substr 문자 추출,
- 코드 스크린

```
void viewMonthTotal() {
    string year, month;
    double incomeTotal = 0, expenseTotal = 0;

    cout << "년도 입력: ";
    cin >> year;
    cout << "월 입력: ";
    cin >> month;

    //반복문을 이용해서 각 항목의 날짜가 사용자가 입력한 년도와 월에 해당하는 지 확인한다.
    //조건문을 이용해서 항목의 날짜가 사용자가 입력한 년도와 월에 해당하면, 해당 항목의 type을 확인
    for (size_t i = 0; i < dates.size(); ++i) {
        if (dates[i].substr(0, 4) == year && dates[i].substr(5, 2) == month) {
            if (types[i] == "수입") {
                incomeTotal += amounts[i];
            }
            else if (types[i] == "지출") {
                expenseTotal += amounts[i];
            }
        }
    }

    cout << month << "월 수입 합계: " << incomeTotal << endl;
    cout << month << "월 지출 합계: " << expenseTotal << endl;
}
```

#### 5)기능 5 (특정 내역 수정 및 삭제)

- 입출력 : 수정할 날짜를 입력하면 일치하는 항목을 찾아 수정할 항목을 선택하여 수정한다. 만약 날짜가 없다면 그에 맞는 문장 출력
- 설명 : 날짜를 입력받고 해당 날짜 항목을 찾고 수정할 항목을 선택한다. 그 이후 해당하는 항목을 수정하고 완료 되면 해당 메시지를 출력한다. 만약 처음 일치하는 날짜가 없을 경우 그에 따른 메시지를 출력한다.
- 적용된 배운 내용 : 벡터, 반복문, 조건문, get
- 코드 스크린

```

void editEntry() {
    string date;
    // 사용자에게 수정할 내역의 날짜를 입력받음
    cout << "수정할 내역의 날짜를 입력하세요 (YYYY-MM-DD): ";
    cin >> date;

    // 반복문을 이용해서 dates 벡터를 사용자가 입력한 날짜와 일치하는 항목 찾기
    for (size_t i = 0; i < dates.size(); ++i) {
        if (dates[i] == date) {
            // 일치하는 항목을 찾았을 때 수정할 항목 선택
            cout << "수정할 항목을 선택하세요 (1. 수입/지출 2. 금액 3. 설명): ";
            int choice;
            cin >> choice;

            // 줄 끝 개행 문자 무시
            cin.ignore();

            // 선택된 항목에 따라 해당 내용을 수정
            switch (choice) {
                case 1: // 수입/지출 타입 수정
                    cout << "새로운 수입/지출 (수입 또는 지출): ";
                    getline(cin, types[i]);
                    break;
                case 2: // 금액 수정
                    cout << "새로운 금액: ";
                    cin >> amounts[i];
                    cin.ignore();
                    break;
                case 3: // 설명 수정
                    cout << "새로운 설명: ";
                    getline(cin, descriptions[i]);
                    break;
                default:
                    cout << "잘못된 선택입니다." << endl;
                    break;
            }

            // 수정이 완료되었음을 알림
            cout << "내역이 수정되었습니다." << endl;

            return;
        }
    }

    // 일치하는 날짜의 항목을 찾지 못한 경우
    cout << "해당 날짜의 내역이 없습니다." << endl;
}

```

## 2) 테스트 결과

### (1) 기능 1 (가계부 프로그램에 어떤 것들이 있는지 출력)

- 설명 : 첫번째 기능의 함수인 메인메뉴를 작성하고 메인함수의 틀을 정리
- 테스트 결과 스크린샷 : 아직 세부 함수들을 정의 안해서 출력 되지 않음

### (2) 기능 2 (수입 / 지출 내역 추가)

- 설명 : 첫번째 1 번에 해당하는 기능의 함수를 작성
- 테스트 결과 스크린샷 : 아직 세부 함수들을 정의 안해서 출력 되지 않음

### **(3) 기능 3 (내역 목록 조회)**

- 설명 : 2 번에 해당하는 기능의 함수를 작성
- 테스트 결과 스크린샷 : 아직 세부 함수들을 정의 안해서 출력 되지 않음

### **(4) 기능 4 (월별 합계 조회)**

- 설명 : 조건문과 반복문을 이용해 원하는 값을 합산하고 저장 그리고 출력
- 테스트 결과 스크린샷 : 아직 세부 함수들을 정의 안해서 출력 되지 않음

### **5)기능 5 (특정 내역 수정 및 삭제)**

- 설명 : 사용자가 특정 날짜의 기록을 쉽게 찾아 수정할 수 있도록 한다.
- 테스트 결과 스크린샷 : 아직 세부 함수들을 정의 안해서 출력 되지 않음

## **4. 계획 대비 변경 사항**

**없음**

## **5. 프로젝트 일정 (참고: 간트차트)**



업무		11/3	11/10	11/17	11/24
제안서 작성		완료			
기능 1	세부기능 1		완료		
기능 2	세부기능 1			완료	
	세부기능 2			완료	
업무		11/24	12/1	12/8	12/15
기능 3		완료			
기능 4			완료		
기능 5				완료	
기능 6					----->

## **5)기능 5 (특정 내역 수정 및 삭제)**

### **2) 테스트 결과**

#### **(1) 기능 1 (가계부 프로그램에 어떤 것들이 있는지 출력)**

- 설명 : 첫번째 기능의 함수인 메인메뉴를 작성하고 메인함수의 틀 을 정리
- 테스트 결과 스크린샷 : 아직 세부 함수들을 정의 안해서 출력 되지 않음

#### **(2) 기능 2 (수입 / 지출 내역 추가)**

- 설명 : 첫번째 1 번에 해당하는 기능의 함수를 작성
- 테스트 결과 스크린샷 : 아직 세부 함수들을 정의 안해서 출력 되지 않음

#### **(3) 기능 3 (내역 목록 조회)**

- 설명 : 2 번에 해당하는 기능의 함수를 작성
- 테스트 결과 스크린샷 : 아직 세부 함수들을 정의 안해서 출력 되지 않음

#### **(4) 기능 4 (월별 합계 조회)**

## **4. 계획 대비 변경 사항**

**없음**

## **5. 프로젝트 일정 (참고: 간트차트)**

업무		11/3	11/10	11/17	11/24
제안서 작성		완료			
기능 1	세부기능 1		완료		
기능 2	세부기능 1			완료	
	세부기능 2			완료	
업무		11/24	12/1	12/8	12/15
기능 3		완료			
기능 4		----->	----->		
기능 5				----->	
기능 6					----->