C++프로그래밍 실습

Weekly-account book

진척 보고서 #1

제출일자:2023-11-25

제출자명:정찬

제출자학번:203804

1. 프로젝트 목표 (16 pt)

1) 배경 및 필요성 (14 pt)

요즘은 카드결제와 온라인 결재로 인해 자신이 얼마나 소비하고 있는지 인지하기 어려운 환경임. 또한 급격한 물가 상승으로 인한 무지출 챌린지가 유행중임. 현재 가계부 앱은 달 기준으로 설정이 되어있어서 예산관리가 어렵다는 문제점이 있음. 이러한 문제점을 해결하기 위해 주단위로 가계부를 작성할 수 있도록 하는 프로그램이 필요함

2) 프로젝트 목표

고객들의 1주일에 얼마나 사용했는지 파악하고 각각의 소비패턴을 고객들에게 알려줌. 또한 전주에 얼마나 사용했는지 비교파악과 더불어 최고 소비액을 어디에다가 사용했는지 출력해줌. 그리고 요즘 유행하는 무지출 챌린지에 맞춰서 하루동안 지출내역이 0이면 무지출한 총 날짜를 알려줌

3) 차별점

주 단위로 예산을 계획할 수 있어서 예산을 유동적으로 계획할 수 있음. 또한 결과값이 달보다는 주로 받으면 직관적이기 때문에 실제 자신이 계획을 잘 지키고 있는지 쉽게 판단 할 수 있음. 무지출을 한 일수를 알려주기 때문에 기존의 프로그램과 차별점이 있음

2. 기능 계획

1) 기능 1 :사용자에게 1주일 지출 내역을 받음

- 설명: 1주일동안 특정한 카테고리에 돈을 얼마나 사용했는지 거래목록과 지출 내역을 사용자에게 직접 받는다

(1) 세부 기능:각 요일마다 입력할 값이 있는지 확인을 한다.

- 설명 (지출이 없는날이 있을수도 있으므로 O/X로 입력을 받는다.)

(2) 세부 기능:카테고리를 입력 받고 거래처와 금액을 입력 받는다.

- 설명 (카테고리는 개발자 임의로 선정을 한 부분에서 숫자로 선택을 하고 거래처와 금액은 사용자가 입력을 받는다.)

(3) 세부 기능:추가적으로 지출이 있을 수 있으므로 더 입력하는지 사용자에게 확인한다.

- 설명 (예:X를 입력하면 다음날로 넘어가서 다시 세부기능(1)을 확인한다.)

2) 기능 가계부 결과 표시

- 설명: 일주일 동안 소비했던 금액중에서 가장 큰 소비액의 거래목록(구매내역)과 금액을 표시하고, 소비한 금액이 각 카테고리별로 몇퍼센트인지 알려준다. 또한 계획된 예산으로 사용했는지 알려주고,무지출인 날짜가 얼마인지 알려준다.

(1) 세부기능: 가장 큰 소비액의 거래목록(구매내역)과 금액을 표시함

(2) 세부기능: 사용자에게 각 카테고리에서 입력받은 금액을 총액으로 나누어서 백분율로 표시함

(3) 세부기능: 사용자가 입력한 예산과 총지출 내역을 비교해서 초과할경우 초과금액을 알려주고 남을경우 남은 금액을 사용자에게 알려줌

(4) 세부기능: 무지출 일수에서 제외할 카테고리를 사용자에게 입력을 받고 그 카테고리를 제외하고 무지출일수를 사용자에게 알려줌

-설명 (사용자가 제외한 카테고리를 지출을 무시하고 나머지 카테고리에 지출이 없을경우 무지출로 인정을함)

3) 기능:1주일전 지출내역 비교

- 설명:1주일전에 입력되어있던 지출내역과 무지출일수를 비교해서 보여줌

(1) 세부기능:일주일전에 사용된 총지출내역과 이번 일주일 총 지출내역을 비교

(2) 세부기능: 일주일전의 무지출일수와 이번 일주일 무지출일수를 비교

4) 기능: 거래내용 수정 및 삭제

(1) 세부기능:사용자가 원하는 날짜의 거래내역을 선택하도록 함

(2) 세부기능:사용자가 거래내역을 수정 및 삭제를 가능하도록 함

3. 진척사항

1) 기능 구현

(1) 1-(1) 세부 기능:각 요일마다 입력할 값이 있는지 확인을 한다.

- 입출력: 사용자가에게 O(X를 제외한 문자) or X 로 입력 받아서 출력여부를 결정

- 설명: 사용자가 O(X를 제외한 문자)를 입력할경우 카테고리 선택,거래처와 금액을 입력하는 코드로 넘어가지만 X를 입력할 경우 다음날로 넘어간다.

- 적용된 배운 내용

  클래스,함수,조건문,벡터,배열등이 사용되었음.

- 코드 스크린샷

```
//사용자에게 입력을 여부를 확인함
void BudgetTracker::ListBudget() {
    for (int i = 0; i < WEEK_DAY; i++) {
        cout << week[i] << endl;
        cout << "입력할게 있으시면 0, 입력할게 없으면 X이라고 적으세요 " << endl;
        string check1;
        cin >> check1;
        if (check1 == "x" or check1 == "X") {
            continue;
        }
```

(2) 1-(2) 세부 기능: 카테고리를 입력 받고 거래처와 금액을 입력 받는다.

(3) 세부 기능: 추가적으로 지출이 있을 수 있으므로 더 입력하는지 사용자에게 확인한다.

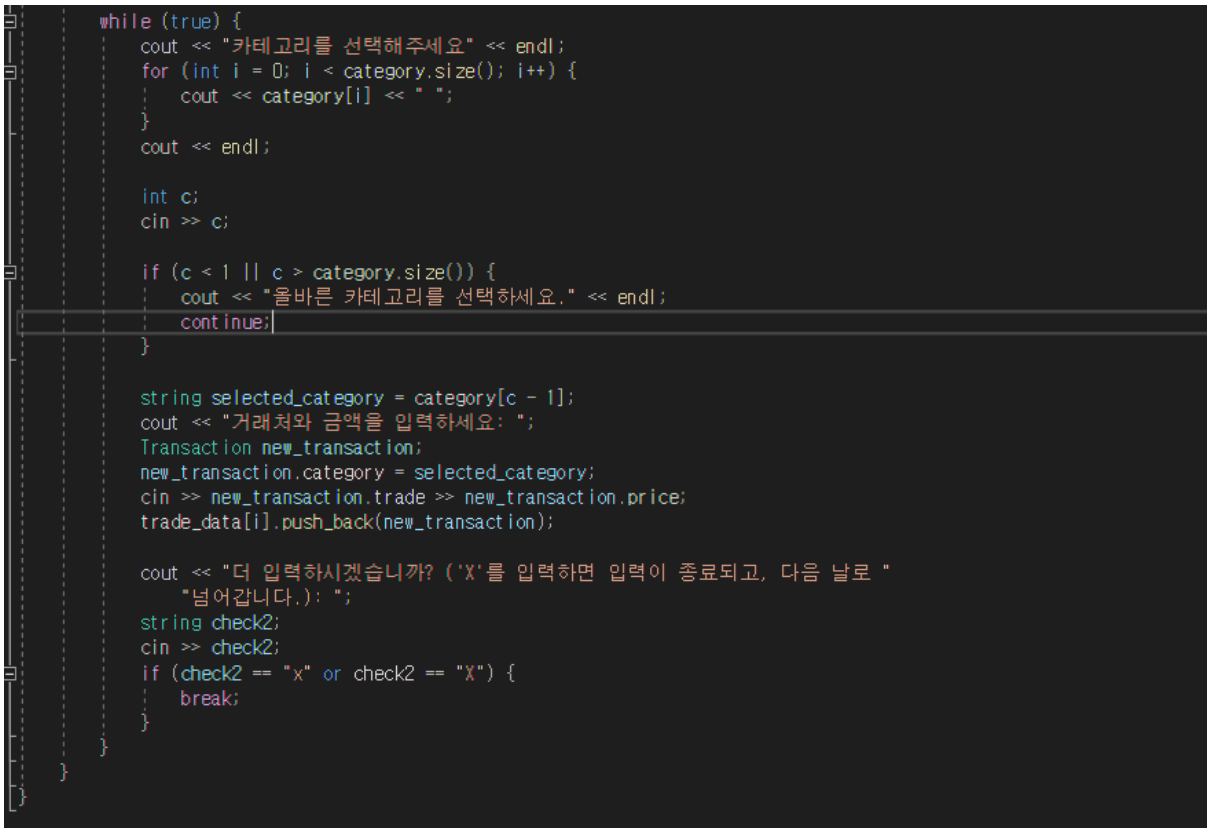
입출력:출력된 카테고리를 사용자가 번호로 선택을 함, 거래처와 금액을 입력, 사용자가 X를 입력할 때 까지 무한반복

설명:1~14까지 개발자가 제공한 카테고리를 숫자로 직접 선택을 함 만약 벗어난 숫자를 입력할 경우 다시 선택하라고 문구가 뜸 카테고리를 입력후 거래처와 금액을 입력함 추가적으로 입력할 거래처가 있을 경우 x를 누르지 않으면 다시 새로운 카테고리와 거래처를 입력할 수있음

- 적용된 배운 내용

  클래스,함수,조건문,벡터,배열,구조체등이 사용되었음.

- 코드 스크린샷



```
        while (true) {
            cout << "카테고리를 선택해주세요" << endl;
            for (int i = 0; i < category.size(); i++) {
                cout << category[i] << " ";
            }
            cout << endl;

            int c;
            cin >> c;

            if (c < 1 || c > category.size()) {
                cout << "올바른 카테고리를 선택하세요." << endl;
                continue;
            }

            string selected_category = category[c - 1];
            cout << "거래처와 금액을 입력하세요: ";
            Transaction new_transaction;
            new_transaction.category = selected_category;
            cin >> new_transaction.trade >> new_transaction.price;
            trade_data[i].push_back(new_transaction);

            cout << "더 입력하시겠습니까? ('X'를 입력하면 입력이 종료되고, 다음 날로 "
                "넘어갑니다.): ";
            string check2;
            cin >> check2;
            if (check2 == "x" or check2 == "X") {
                break;
            }
        }
    }
}
```

(3) 2-(1) 세부기능: 가장 큰 소비액의 거래목록(구매내역)과 금액을 표시함

입출력: 거래처와 거래금액을 입력으로 받아 가장 각 지출을 비교하여서 가장 큰 소비액을 출력한다

. - 적용된 배운 내용

조건문이 사용되었음

- 코드 스크린샷

```
            total_consumption += transaction.price;
            if (transaction.price > max_price) {
                max_price = transaction.price;
            }
        }

        if (allCategoriesNoExpenditure) {
            no_expenditure++;
        }
    }
}

average_consumption = CalculateAverageConsumption();
category_percentage = AnalyzeCategoryPercentage(total_consumption);

cout << "무지출 일수는 " << no_expenditure << " 입니다" << endl;
cout << "총 소비금액는 " << total_consumption << "원 입니다" << endl;
cout << "최대 소비금액은 " << max_price << "원 입니다" << endl;
```

(4) (2) 세부기능: 사용자에게 각 카테고리에서 입력받은 금액을 총액으로 나누어서 백분율로 표시함

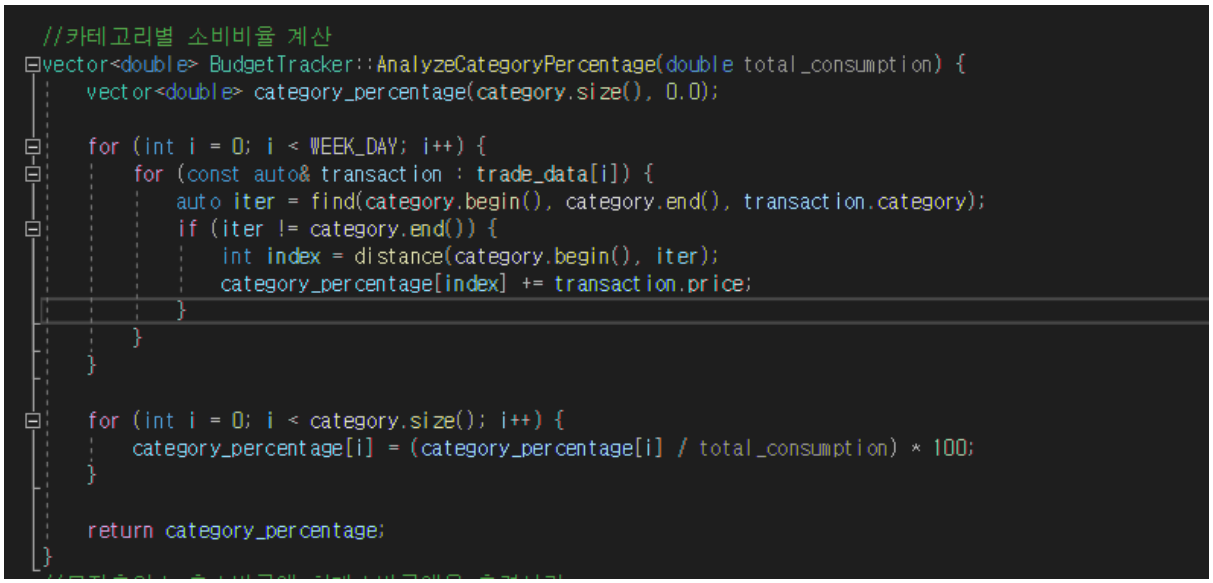
입출력: 총 소비액을 입력받아 각 카테고리별로 소비비율을 출력한다.

설명: **category_percentage**라는 벡터를 초기화시킨후 각 요일마다 거래내역을 확인함. find함수로 인덱스를 찾으면 **category_percentage의 값을 거래 금액만큼 증가시킴 모든 거래 내역을 확인했으면 전체 소비금액으로 나눠 비율 계산후 *100을하여 실제 퍼센티지로 변환을 하고 벡터값을 반환함.**

- 적용된 배운 내용

클래스,함수,조건문,벡터,배열,구조체등이 사용되었음.

- 코드 스크린샷



```
//카테고리별 소비비율 계산
vector<double> BudgetTracker::AnalyzeCategoryPercentage(double total_consumption) {
    vector<double> category_percentage(category.size(), 0.0);

    for (int i = 0; i < WEEK_DAY; i++) {
        for (const auto& transaction : trade_data[i]) {
            auto iter = find(category.begin(), category.end(), transaction.category);
            if (iter != category.end()) {
                int index = distance(category.begin(), iter);
                category_percentage[index] += transaction.price;
            }
        }
    }

    for (int i = 0; i < category.size(); i++) {
        category_percentage[i] = (category_percentage[i] / total_consumption) * 100;
    }

    return category_percentage;
}
```

(4) (3) 세부기능: 사용자가 입력한 예산과 총지출 내역을 비교해서 초과할경우 초과금액을 알려주고 남을경우 남은 금액을 사용자에게 알려줌

입출력: 사용자에게 입력받은 week_buget과 total_consumption의 차이를 출력시킴

설명: 차이가 양수이면 "남은예산을 표시",차이가 음수라면 "초과한 소비금액"을 출력 시킴

- 적용된 배운 내용

클래스,함수,조건문,벡터,배열,문자열 함수 등이 사용되었음.

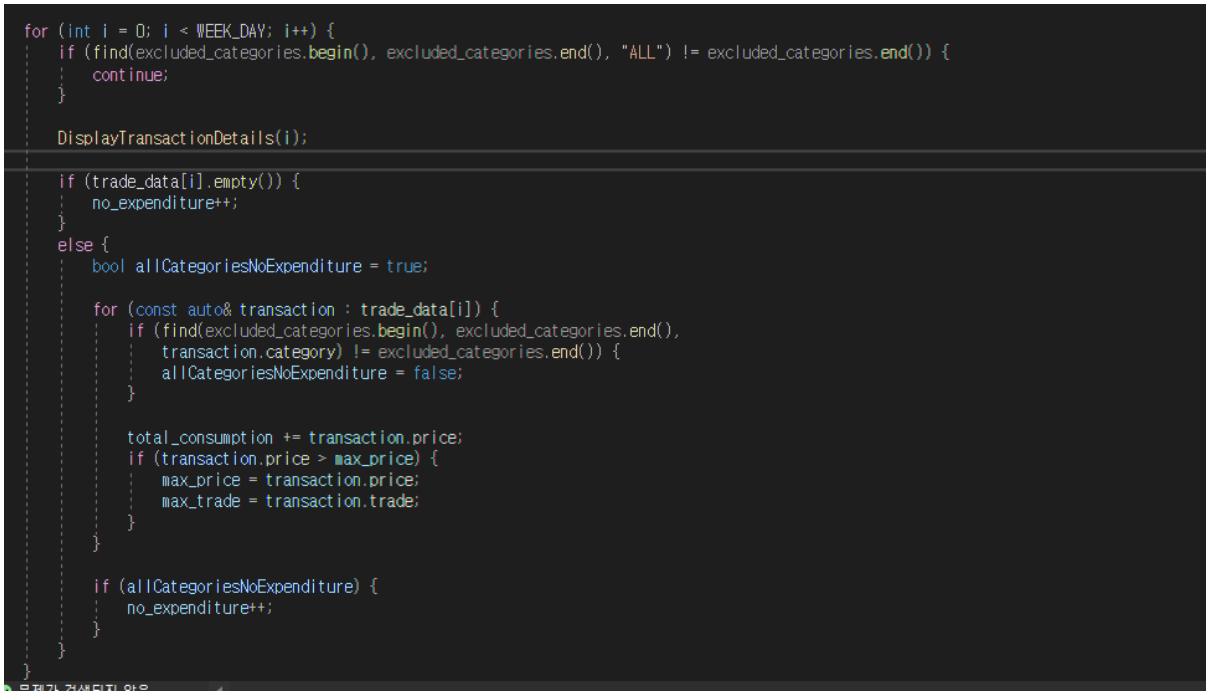
-코드 스크린샷

```
double average_consumption = 0;
vector<double> category_percentage;
double total_consumption =
    Result(excluded_categories, average_consumption, category_percentage);

if (week_budget > total_consumption) {
    cout << "남은 예산!" << endl;
    cout << "---------------------------" << endl;
    cout << week_budget - total_consumption << "원이 남았습니다" << endl;
}
else {
    cout << "예산 초과!!" << endl;
    cout << "---------------------------" << endl;
    cout << total_consumption - week_budget << "원을 초과했습니다" << endl;
}

cout << "일 평균 소비금액: " << average_consumption << endl;

cout << "카테고리별 소비 분석:" << endl;
for (int i = 0; i < category.size(); i++) {
    cout << category[i] << ": " << category_percentage[i] << "%" << endl;
}
```

(5) 2-4 세부기능: 무지출 일수에서 제외할 카테고리를 사용자에게 입력을 받고 그 카테고리를 제외하고 무지출일수를 사용자에게 알려줌

입출력:사용자에게 무지출 제외 카테고리를 받고 그것을 제외하고 출력

-코드 스크린샷



```
for (int i = 0; i < WEEK_DAY; i++) {
    if (find(excluded_categories.begin(), excluded_categories.end(), "ALL") != excluded_categories.end()) {
        continue;
    }

    DisplayTransactionDetails(i);

    if (trade_data[i].empty()) {
        no_expenditure++;
    }
    else {
        bool allCategoriesNoExpenditure = true;

        for (const auto& transaction : trade_data[i]) {
            if (find(excluded_categories.begin(), excluded_categories.end(),
                transaction.category) != excluded_categories.end()) {
                allCategoriesNoExpenditure = false;
            }

            total_consumption += transaction.price;
            if (transaction.price > max_price) {
                max_price = transaction.price;
                max_trade = transaction.trade;
            }
        }

        if (allCategoriesNoExpenditure) {
            no_expenditure++;
        }
    }
}
```

- 적용된 배운 내용

문자열함수,참조,배열등이 사용됨

(6) 3-1세부기능:일주일전에 사용된 총지출내역과 이번 일주일 총 지출내역을 비교

입출력: 사용자에게 입력받은 가계부 정보를 user_input의 txt파일로 기록을 연속적으로 해나감

설명: 이 입출력 데이터를 가지고 사용자의 가계부의 패턴을 분석할 수 있음

- 적용된 배운 내용

파일 입출력,예외처리(try-catch)이 사용되었다.

```
            string selected_category = category[c - 1];
            cout << "거래처와 금액을 입력하세요: ";
            Transaction new_transaction;
            new_transaction.category = selected_category;
            cin >> new_transaction.trade >> new_transaction.price;

            outputFile << week[day] << endl;
            outputFile << selected_category << " " << new_transaction.trade << " " << new_transaction.price << endl;

            // 입력 에러 처리
            if (cin.fail() || new_transaction.price < 0) {
                throw std::invalid_argument("올바른 금액을 입력하세요.");
            }

            trade_data[day].push_back(new_transaction);

            cout << "더 입력하시겠습니까? ('X'를 입력하면 입력이 종료되고, 다음 날로 "
                "넘어갑니다.): ";
            string check2;
            cin >> check2;
            if (check2 == "x" or check2 == "X") {
                break;
            }
        }
        catch (const std::exception& e) {
            cout << "예외 발생: " << e.what() << endl;
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
        }
    }

    outputFile.close();
```

```
if (check1 == "x" or check1 == "X") {
    continue;
}
ofstream outputFile{ "user_input.txt", ios::app };

if (!outputFile.is_open()) {
    cout << "파일을 열 수 없습니다." << endl;
    return;
}

while (true) {
    cout << "카테고리를 선택해주세요" << endl;
    for (int j = 0; j < category.size(); j++) {
        cout << category[j] << " ";
    }
    cout << endl;
    try {
        int c;
        cout << "카테고리 번호를 입력하세요: ";
        cin >> c;

        if (cin.fail()) {
            throw std::invalid_argument("숫자를 입력하세요.");
        }

        if (c < 1 || c > category.size()) {
            throw std::out_of_range("올바른 카테고리를 선택하세요.");
        }
```

2) 테스트 결과

(1) 1-(1) 세부 기능:각 요일마다 입력할 값이 있는지 확인을 한다

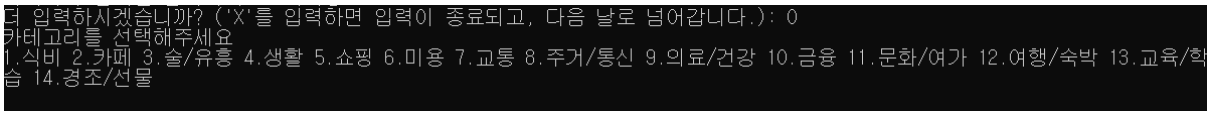
.- 설명

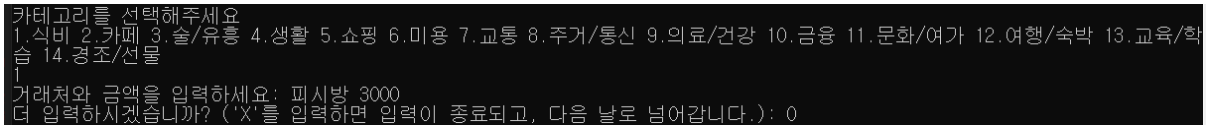
- 테스트 결과 스크린샷

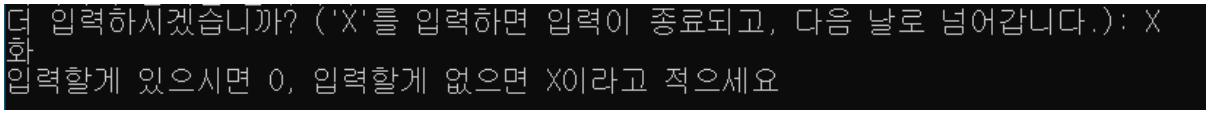
(2) 1-(2) 세부 기능: 카테고리를 입력 받고 거래처와 금액을 입력 받는다.

(3) 세부 기능: 추가적으로 지출이 있을 수 있으므로 더 입력하는지 사용자에게 확인한다.

- 테스트 결과 스크린샷

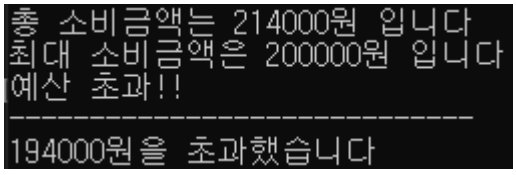






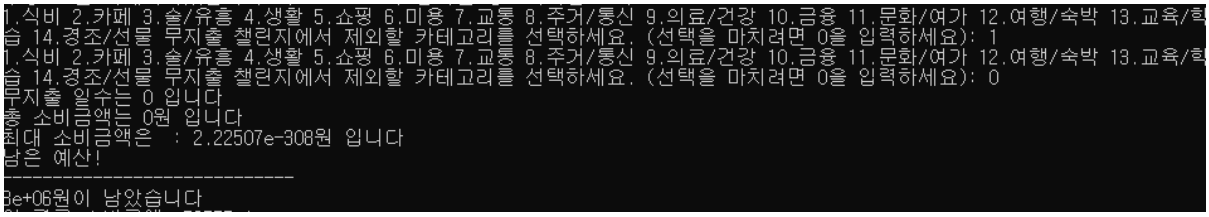
(3) 2-(1) 세부기능: 가장 큰 소비액의 거래목록(구매내역)과 금액을 표시함

입출력: 거래처와 거래금액을 입력으로 받아 가장 각 지출을 비교하여서 가장 큰 소비액을 출력한다



(4) (2) 세부기능: 사용자에게 각 카테고리에서 입력받은 금액을 총액으로 나누어서 백분율로 표시함

(5) 2-4 세부기능: 무지출 일수에서 제외할 카테고리를 사용자에게 입력을 받고 그 카테고리를 제외하고 무지출일수를 사용자에게 알려줌



(6) 3-1세부기능:일주일전에 사용된 총지출내역과 이번 일주일 총 지출내역을 비교

금
1.식비 식비 3000
목
1.식비 tlrqfl 2000
월
1.식비 tlfql 2000
월
1.식비 tlfql 2000
월
1.식비 피시방 2000
월
1.식비 sf 2000
목
1.식비 피시방 2000
금
1.식비 피시방 20000
금
2.카페 24 324
월
1.식비 학식 5000
화
2.카페 스타벅스 6900
금
10.금융 삼성전자 135000
토
11.문화/여가 CGV 15000
일
13.교육/학습 메가스터디 200000
일
3.술/유흥 역전할맥 34000
목
1.식비 vltlqkd 2000

4. 계획 대비 변경 사항

1) (2) 세부 기능:카테고리를 입력 받고 거래처와 금액을 입력 받는다.

- 이전: X

- 이후: 추가한 기능.

- 사유

카테고리별로 가계부를 입력받으면 사용자가 어디 부문에 돈을 많이 사용했는지 쉽게 파악할 수 있어서 추가하게 됨

2) (2) 세부기능: 사용자에게 각 카테고리에서 입력받은 금액을 총액으로 나누어서 백분율로 표시함

- 이전: X

- 이후: 추가한 기능.

-사유

사용자에게 어디 부문에 돈을 많이 사용했는지 수치적으로 제공하기 위해서 추가함

3) (4) 세부기능: 무지출 일수에서 제외할 카테고리를 사용자에게 입력을 받고 그 카테고리를 제외하고 무지출일수를 사용자에게 알려줌

- 이전: X

- 이후: 추가한 기능.

-사유

사용자가 어쩔 수 없이 매일 사용하는 지출이 있기 때문에 그걸 제외하고 무지출 일수를 계산하도록 도와주기 위해서 추가함

4) (1) 세부기능:사용자가 원하는 날짜의 거래내역을 선택하도록 함

- 이전: X

- 이후: 추가한 기능.

-사유

사용자가 잘못 입력을 했을 경우 수정하게 위해서 추가함

5) (1) 세부기능:사용자가 원하는 날짜의 거래내역을 선택하도록 함

- 이전: X

- 이후: 추가한 기능.

-사유

사용자가 잘못 입력을 했을 경우 수정하게 위해서 추가함

6) (2) 세부기능:사용자가 거래내역을 수정 및 삭제를 가능하도록 함

- 이전: X

- 이후: 추가한 기능.

-사유

사용자가 잘못 입력을 했을 경우 수정하게 위해서 추가함

5. 프로젝트 일정 (Red=데드라인,Green=기능 제작,BLACK=제

작완료,YELLO=보충)

업무		11/3	11/10	11/17	11/24	12/1	12/8	12/15	12/23
제안서 작성									
기능1	세부 기능1								
	세부 기능2								
	세부 기능3								
기능2	세부 기능1								
	세부 기능2								
	세부 기능3								
	세부 기능4								
기능3	세부 기능1								
	세부 기능2								
기능4	세부 기능1								
	세부 기능2								