

수업명:c프로그래밍및실습

나의 지출 알아보기

프로젝트

진척 보고서 #2

제출일자:12월10일

제출자명:장예성

1. 프로젝트 목표

1) 배경 및 필요성

최근들어 물가상승과 무분별한 소비를 하는사람의 증가에 따라 2020년 기준 대학생 월 평균 생활비가 59만원으로 상당히 큰 금액을 인당 소비하는것으로 나타났습니다.(출처:잡코리아X알바몬 통계센터)이 프로젝트를 통해 프로그램을 만들게 된 배경은 스스로 알바를 하며 합리적인 소비를 하는것은 괜찮지만 아무런 제약과 생각을 하지않고 부모님에게 받은 용돈으로만 무분별하게 소비를 하게 되니돈이 물흐르듯이 빠져 나가게 되는걸 깨닫게 되었고 이제부터라도 스스로 내가쓴돈을 기록하며 왜 이 돈을 지출하게 되었는지에 대한 이유를 함께 적어놓으며한달 동안 쓴 돈들을 보며 쓰지 말아야 할 돈을 쓰지는 않았나 생각해보고 반성하기위해 이 프로그램을 만들게 된 배경이 되었습니다.

2) 프로젝트 목표

내가 한달동안 쓴 돈들의 카테고리과 가격 및 돈을 쓰게된 이유등을 입력하고 한눈에 지출 내역을 확인하는것을 목표로 함.

3) 차별점

기존의 가계부들은 지출에 대해 카테고리과 가격등만 입력하지만 이와달리 지출내용에 대해 자세하게 설명할수있는 메모등을 추가하여 지출의 이유에 대해 나중에 자세하게 알아볼수 있게 합니다. 지출의 카테고리를 입력하기 때문에 카테고리 별로 어떠한 범주에서 지출이 제일 많았는지 그래프등으로 확인할수 있다는차별점 또한 존재합니다. 또한 한달 예산을 미리 입력해 두고 지출이 예산을 초과하였을 경우에 경고 알림또한 주어 경각심을 주는것도 차별화 된 점이라고 할수

있습니다.또한 소비방법에 대한 동향을 알아볼수있는 방법도한 존재합니다.카드현금 그 외의 것으로 지출을 하게 되는데 자신이 어떠한 방법으로 가장 많은지출을 하게 되는지 또한 알아볼수 있다는 점에서 차별점이라고 할수있습니다.

2. 기능 계획

1) 기능 1 자신의 지출의 카테고리 및 가격 입력

- 설명 :자신의 지출에 대해 주어진 카테고리에서 선택하고 가격을 입력한다.

(1) 세부 기능- 설명:코딩을 통해 지출에 대한 여러가지 카테고리를 제시한 후 사용자에게 선택하라고한다. 카테고리의 예시는 식비,옷,필기구 등을 예시로 들수있다.

(2) 세부 기능- 설명:카테고리를 사용자가 선택하게 되면 (예를들어 식비 라는 카테고리 선택) 식비라는 카테고리를 선택하게 되면 세부적인 사항을 입력하세요라는 문구가 뜨게 되고 (햄버거 세트:9500원)과 같이 세부적인 사항들을 입력을한다.

2) 기능 2 지출 내역에 대한 메모

(1) 세부 기능- 설명:지출을 하게 되면 나중에 볼때는 왜 이런 소비를 하였는지 이유를 알수없기 때문에 지출한 내역에 대해 간단하게 혹은 자세하게 메모를 입력한다.

3) 기능 3 예산 설정 및 알림

(1) 세부 기능- 설명:한달동안 사용할 예산을 초기에 설정 해 두고 한달동안의 지출이 예산을 초과 한경우 최종 화면에서 경고를 해 주어 경각심을 주는 기능이다.

4) 기능 4 지출 내역에 대한 표시

(1) 세부 기능- 설명: 지출 내역에 대해 입력 순대로 텍스트로 보여주는 기능이다. 앞서 나눈 카테고리별로 지출 내역을 날짜와 함께 보여주고 최종적으로 그 카테

고리 안에서 쓴 가격의 최종합까지 출력 한다.

(2) 세부 기능- 설명: 지출 내역에 대해 범주별로의 횟수를 그래프로 보여준다.

5) 기능 5 보안을 위한 비밀번호

(1) 세부 기능- 설명: 사전에 사용자가 사용할 비밀번호를 입력해 놓으면 나중에 비밀 번호를 입력해야만 가계부에 접속할수 있도록 한다.

6) 기능 6 지출방식의 동향

(1) 세부 기능- 설명: 카드로 소비하였는지 현금으로 소비하였는지 아니면 그 외의 방법으로 소비하였는지를 입력하여 최종 결과에서 자신이 어떤방식으로 지출을 많이하는지도 알려준다.

3. 진척사항

1) 기능 구현

(1) 구현한 기능 이름 :자신의 지출의 카테고리 및 가격 입력

- 입출력

- 설명:

1.char categories[MAX_CATEGORIES][20] = { "식비", "학교생활", "의류", "기타" };;

문자열을 저장하는 이차원 배열 categories를 선언하고 초기화합니다. 여기서 MAX_CATEGORIES는 4이며, 각 카테고리의 이름을 저장합니다.

2.struct Expense expenses[MAX_CATEGORIES];

Expense구조체 배열인 expenses를 선언합니다. 이 배열은 각 카테고리에 대한 지출을 추적합니다.

3.int cardCount = 0, cashCount = 0, otherCount = 0;;

cardCount, cashCount, otherCount변수를 초기화하여 카드, 현금, 기타 지출을 추적하는

데 사용됩니다.

4.for루프:expenses배열의 각 요소에 해당하는 카테고리와 연결된 Expense구조체의 초기 값을 설정합니다. strcpy를 사용하여 각 요소의 category멤버를 해당하는 카테고리로 복사하고, 카드, 현금, 기타 지출을 추적하기 위한 cardCount, cashCount, otherCount멤버를 0으로 초기화합니다.

5.카테고리 선택과 관련된 부분:사용자에게 카테고리를 선택하도록 요청하고, 선택한 카테고리를 표시합니다. 이때 사용자의 입력을 받아 올바른 범위인지 확인하고, 0을 입력하면 프로그램이 종료됩니다.

- 적용된 배운 내용:2차원 배열,구조체,함수,반복문(for문 while문)

-코드스크린샷

```
기타 파일 (진여 범위) main()
65 int main() {
66     char categories[MAX_CATEGORIES][20] = { "식비", "학교생활", "의류", "기타" };
67     struct Expense expenses[MAX_CATEGORIES];
68     int cardCount = 0, cashCount = 0, otherCount = 0;
69
70     for (int i = 0; i < MAX_CATEGORIES; i++) {
71         strcpy(expenses[i].category, categories[i]);
72         expenses[i].cardCount = 0;
73         expenses[i].cashCount = 0;
74         expenses[i].otherCount = 0;
75     }
76
77     printf("카테고리를 선택하세요:\n");
78     for (int i = 0; i < MAX_CATEGORIES; i++) {
79         printf("%d. %s\n", i + 1, categories[i]);
80     }
81
82     while (1) {
83         int category_choice;
84         printf("선택 (1-%d) (0을 입력하면 종료): ", MAX_CATEGORIES);
85         scanf("%d", &category_choice);
86
87         if (category_choice == 0) {
88             printf("프로그램을 종료합니다.\n");
89             break;
90         }
91
92         if (category_choice < 1 || category_choice > MAX_CATEGORIES) {
93             printf("잘못된 선택입니다.\n");
94             continue;
95         }
96
97         printf("%s를 선택하셨습니다.\n", categories[category_choice - 1]);
98
99         char detail[MAX_DETAIL_LENGTH];
100         printf("세부 내역 및 지출의 이유를 입력해주세요 (예: 음식명:가격): ");
101         getchar();
102         fgets(detail, sizeof(detail), stdin);
103         detail[strcspn(detail, "\n")] = 0;
104
105         if (strlen(detail) >= MAX_DETAIL_LENGTH) {
106             printf("입력이 너무 길니다.\n");
107             continue;
108         }
109     }
110 }
```

Windows 정품 인증

2) 기능 구현

(2) 구현한 기능 이름 :지출방식의 동향

설명:

1.사용자 입력 및 선택:

printf("지출 방법을 선택하세요 (1. 카드, 2. 현금, 3. 기타): ");를 통해 사용자에게 지출 방법을 선택하라는 메시지를 표시합니다.scanf("%d", &method_choice);를 사용하여 사용자로부터 지출 방법을 입력받습니다.

2.Switch 문을 통한 지출 방법 처리:

사용자가 선택한 method_choice값에 따라서 switch문을 통해 다른 작업이 수행됩니다.

예를 들어, 사용자가 1을 선택하면 카드 지출을 의미하며, 해당하는 case 1:의 블록이 실행됩니다.

선택된 지출 방법은 detail문자열에 해당 내용을 추가하고, 해당 지출 방법 카운트 (cardCount, cashCount, otherCount)를 증가시킵니다.

3.Expense 구조체 업데이트:

currentExpense구조체에 사용자가 선택한 카테고리과 지출 내역(detail)을 저장합니다.

이후 saveExpenseToFile(¤tExpense);를 호출하여 해당 지출 내역을 파일에 저장합니다.

선택한 카테고리의 인덱스를 구한 후, 해당 카테고리의 지출 횟수를 업데이트합니다.

4.출력 및 확인:

지출이 성공적으로 추가되었다는 메시지를 출력합니다.

showExpenseFromFile(&expenses[index]);를 호출하여 해당 카테고리의 지출 내역을 파일에서 읽어와 출력합니다.

- 적용된 배운 내용:입출력함수,조건문(switch문),구조체,함수

-코드스크린샷

```

printf("지출 방법을 선택하세요 (1. 카드, 2. 현금, 3. 기타): ");
int method_choice;
scanf("%d", &method_choice);

switch (method_choice) {
case 1:
    strcpy_s(detail, sizeof(detail), " (카드)");
    cardCount++;
    break;
case 2:
    strcpy_s(detail, sizeof(detail), " (현금)");
    cashCount++;
    break;
case 3:
    strcpy_s(detail, sizeof(detail), " (기타)");
    otherCount++;
    break;
default:
    printf("잘못된 선택입니다.\n");
    continue;
}

struct Expense currentExpense;
strcpy(currentExpense.category, categories[category_choice - 1]);
strcpy(currentExpense.detail, detail);

saveExpenseToFile(&currentExpense);
int index = category_choice - 1;
expenses[index].cardCount = cardCount;
expenses[index].cashCount = cashCount;
expenses[index].otherCount = otherCount;

printf("지출이 성공적으로 추가되었습니다!\n");
showExpenseFromFile(&expenses[index]);
}

return 0;

```

Windows 정품

2) 기능 구현

(2) 구현한 기능 이름 : 지출 내역에 대한 표시

설명:

1: 사용자가 "햄버거:9000"을 입력하면 이 값은 detail 변수에 저장됩니다.

2: saveExpenseToFile 함수에서 fprintf(file, "%s\n", expense->detail); 이 부분은 expense->detail에 저장된 "햄버거:9000"을 파일에 쓰는 역할을 합니다.

3: fprintf 함수는 파일에 문자열을 출력하는 함수입니다. %s는 문자열 형식 지정자로, expense->detail에 있는 내용을 파일에 씁니다. 그래서 "햄버거:9000"이 파일에 저장됩니다. 개행 문자(\n)는 각 지출 내역을 새 줄에 기록하기 위해 추가됩니다.

fprintf(file, "%s\n", expense->detail); 이 부분은 사용자가 입력한 내용인 "햄버거:9000"을 파일에 한 줄로 저장하게 됩니다.

-코드스크린샷:


```

void saveExpenseToFile(struct Expense* expense) {
    FILE* file;
    if (fopen_s(&file, expense->category, "a") != 0) {
        printf("파일을 열 수 없습니다.\n");
        return;
    }

    fprintf(file, "%s\n", expense->detail);
    fclose(file);
}

```

2) 테스트 결과

(1) 테스트한 기능 이름: 자신의 지출의 카테고리 및 가격 입력

- 설명: 카테고리 1~4중 하나를 선택하면 식비라는 txt파일이 만들어진다 이 후 햄버거:9000 처럼 제품의 이름과 가격을 입력하면 저장이 된다
- 테스트 결과 스크린샷:

```

카테고리를 선택하세요:
1. 식비
2. 학교생활
3. 의류
4. 기타
선택(1-4) (0을 입력하면 종료): 1
식비를 선택하셨습니다.
세부 내역 및 지출의 이유를 입력해주세요 (예: 음식명:가격): 햄버거:9000

```

(2) 테스트한 기능 이름: 지출방식의 동향

- 설명: 지출방법을 선택하라는 문구가 나오고 1카드 2현금 3기타중 하나를 선택하면 지출방식의 카테고리를 누적해서 몇번 사용했는지 보여준다
- 테스트 결과 스크린샷:

```

카테고리를 선택하세요:
1. 식비
2. 학교생활
3. 여류
4. 기타
선택(1-4) (0을 입력하면 종료): 1
식비를 선택하셨습니다.
세부 내역 및 시술의 이유를 입력해주세요 (예: 음식명:가격): 햄버거:9000
시술 방법을 선택하세요 (1. 카드, 2. 현금, 3. 기니): 1
시술이 성공적으로 수기되었습니다!

식비 지출 내역:
1. 햄버거:9000 (카드)

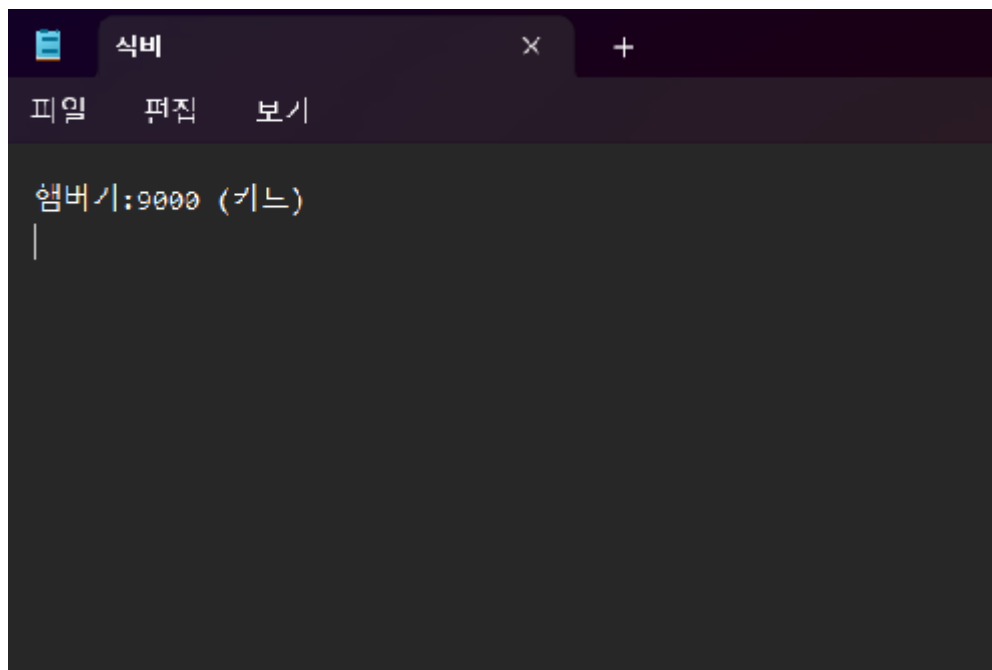
카드: 1회, 현금: 0회, 기타: 0회
선택(1-4) (0을 입력하면 종료): |

```

(3) 테스트한 기능 이름:지출 내역에 대한 표시

-설명:앞서 저장한 햄버거:9000 추가로 카드는 소비방식까지 선택했으므로 햄버거:9000(카드) 라는 문구가 식비라는 txt파일에 저장이 된다 추후에 식비 카테고리를 또 한번 다시 고르게 되면 이 txt파일에 다른 내역이 추가가 되게 된다

-테스트 결과 스크린샷:



4. 계획 대비 변경 사항:없음

5. 프로젝트 일정

업무		11/3	11/26	12/10	12/26
제안서 작성		완료			
기능1	세부기능1		완료		
	세부기능2		완료		
기능2	세부기능1		완료		
기능3	세부기능1			----->	
기능4	세부기능1			완료	
	세부기능2			----->	
기능5	세부기능1			----->	
기능6	세부기능1			완료	