

c프로그래밍 및 실습

Automatic size

Recommendatin

System

최종보고서

제출일자: 12/24

제출자명: 정찬민

제출자학번: 232908

1. 프로젝트 목표

1) 배경 및 필요성

최근에 인터넷에서 바지를 구입한 적이 있는데, 막상 입어보니 생각했던 것보다 바지 길이가 길어서 수선을 맡겨야 했던 기억이 있다. 이처럼 고객들은 온라인 쇼핑 시 제품 사이즈 선택에 어려움을 겪는 경우가 많고, 잘 맞는 제품을 찾는 것이 어렵다. 이로 인해 반품률과 고객 불만족이 꽤 증가하고 있다. 이 문제를 해결하기 위해 자동 크기 추천 시스템이 필요하다고 생각한다.

2) 프로젝트 목표

이 프로젝트의 목표는 고객의 쇼핑 경험을 개선하고 반품률을 줄이기 위해, 고객의 개별적인 신체 측정 데이터와 이전 구매 내역을 활용하여 최적의 제품 사이즈를 추천하는 시스템을 구축하는 것이다. 이를 통해 고객은 더 정확한 제품 크기를 선택하고, 이로 인해 불만족 및 반품률을 감소시키며, 최종적으로 고객 만족도와 수익까지 향상시킬 수 있다.

3) 차별점

▶ 개인화된 사이즈 추천: 기존의 크기 차트 시스템은 일반적인 통계 데이터를 기반으로 제품 크기를 제안한다. 하지만 이 시스템은 개별 고객의 신체 측정 정보와 이전 구매 내역을 분석하여, 고객별로 정확한 크기를 추천함으로써 맞춤형 서비스를 제공한다.

▶ 유사한 사용자 추천: 비슷한 체형을 가진 다른 사용자가 구매한 제품과 사이즈 정보를 활용하여 고객에게 추천 제품을 제시한다. 이것은 다른 사용자의 실제 경험을 통해 보다 정확한 추천을 가능하게 한다.

2. 기능 계획

기능 1 : 고객 신체 측정 정보 수집

- 설명 : 고객에게 신체 측정 정보를 입력하도록 하는 기능

(1) 세부 기능 (고객 신체 정보 입력)

- 설명 : 키, 몸무게, 허리둘레 등의 신체 정보를 고객이 입력할 수 있는 폼을 제공

기능 2 : 이전 구매 내역 분석

- 설명 : 고객의 이전 주문 내역을 분석하여 선호하는 제품과 사이즈를 확인하는 기능

(1) 세부 기능 (이전 주문 내역 가져오기)

- 설명 : 고객의 이전 주문 내역 중 최근 주문 정보를 불러옴

기능 3 : 비슷한 사용자 추천

- 설명 : 비슷한 체형을 가진 다른 사용자가 구매한 제품과 사이즈 정보를 추천해주는 기능

(1) 세부 기능 (비슷한 사용자의 제품 추천)

- 설명 : 비슷한 체형을 가진 사용자가 주로 구매한 제품과 사이즈 정보를 기반으로 고객에게 추천 제품을 제시

4) 기능 4 : 고객의 구매 내역 저장

설명: 고객이 다음번에 이용했을 때 기능(2)를 사용하기 위해 고객이 입력한 구매 내역들을 저장

5) 기능 5 : 옷 구입 단계

설명: 고객이 옷과 사이즈를 선택할 수 있는 구입 단계

3. 진척사항

1) 기능 구현

(1) 고객 신체 측정 정보 수집 및 구매 내역

- 입출력

```
// 구조체 정의: 고객 정보
struct CustomerInfo {
    float height;
    float weight;
    int waistSize;
    char size[3]; // S, M, L, XL
};

// 구조체 정의: 구매 내역
struct PurchaseHistory {
    char itemName[50];
    float price;
    char size[3]; // S, M, L, XL
};
```

'CustomerInfo' : 고객의 신체 정보를 저장하는 구조체이다. 키, 몸무게, 허리둘레, 선택한 사이즈를 저장한다.

'PurchaseHistroy' : 각 구매에 대한 정보를 저장하는 구조체이다. 구매한 제품명, 가격, 선택한 사이즈를 저장한다.

- 설명

Main 함수에서 먼저 CustomerInfo 와 PurchaseHistory 구조체를 사용하여 고객 정보와 이전 구매 내역을 저장할 변수들을 선언한다. 그리고 사용자에게 고객 정보를 입력받고, 옷 구매 창을 표시하여 제품을 선택하고 구매한다, 구매한 내역을 이전 구매 내역에 추가하고, 이전 구매 내역을 화면에 출력한다.

- 적용된 배운 내용

6~7주차에 걸쳐 배웠던 배열을 사용해서 'char itemName[]' 이라는 구매한 제품의 이름을 나타내주는 문자열 배열을 만들었다.

10주차, 12주차에 배웠던 포인터와 구조체를 사용하여 고객의 신체 정보를 담고 있는 'struct Customerinfo* customer' 를 코드를 짜서 고객의 신체 정보를 담고 있는 'CustomerInfo' 구조체에 대한 포인터를 만들었다. 이를 이용해서 고객의 정보를 확인할 수 있게끔 했다. 또 'struct PurchaseHistroy* purchaseHistory' 를 짜서 이전 구매 내역을 담고 있는 배열에 대한 포인터도 만들었다.

버퍼 오버플로우를 방지하기 위해, sizeof(purchase.size) 사용

- 코드 스크린샷

```
// 함수 선언
float purchaseProduct(char itemName[], float price, struct CustomerInfo* customer, struct PurchaseHistory* purchaseHistory, int* numPurchase);

int main() {
    // 구조체 변수 선언
    struct CustomerInfo customer;

    // 이전 구매 내역 배열 선언
    struct PurchaseHistory purchaseHistory[10]; // 최대 10개의 구매 내역 저장 가능
    int numPurchase = 0; // 현재까지 저장된 구매 내역 개수

    // 고객에게 정보 입력 요청
    printf("고객님의 키를 입력하세요 (cm): ");
    scanf_s("%f", &customer.height, sizeof(customer.height));

    printf("고객님의 몸무게를 입력하세요 (kg): ");
    scanf_s("%f", &customer.weight, sizeof(customer.weight));

    printf("고객님의 허리둘레를 입력하세요 (cm): ");
    scanf_s("%d", &customer.waistSize, sizeof(customer.waistSize));

    // 입력받은 정보 출력
    printf("\n고객님의 신체 정보:\n");
    printf("키: %.2f cm\n", customer.height);
    printf("몸무게: %.2f kg\n", customer.weight);
    printf("허리둘레: %d cm\n", customer.waistSize);
}
```

```
// 이전 구매 내역 출력
printf("\n=== 이전 구매 내역 ===\n");
for (int i = 0; i <= numPurchase; i++) {
    printf("%d. %s (Size: %s) - $%.2f\n", i + 1, purchaseHistory[i].itemName, purchaseHistory[i].size, purchaseHistory[i].price);
}

return 0;
```

```
// 이전 구매 내역에 추가
strcpy_s(purchaseHistory[numPurchase].size, sizeof(purchase.size), purchase.size);
(*numPurchase)++;

return purchase.price;
```

2) 테스트 결과

(1) 고객 신체 측정 정보 수집 및 이전 구매 내역

- 설명

고객이 이 시스템을 사용하기 전에 가장 처음에 입력 받아야 할 기능이다. 고객이 자신의 신체 정보를 입력함으로써 다음번에 의류를 구매할 때는 저장된 신체정보를 기반으로 옷의 사이즈를 추천받을 수 있게 된다.

사용자가 구입한 옷을 이전 구매 내역에 저장하고 사용자가 다음번에 사용할 때 이전 구매 내역의 저장된 내용을 보여줘야 하는데, 1차 진척보고서에서는 '흑청바지'라는 옷을 사용자가 구매했다는 전제하에 purchaseHistory에 흑청바지를 저장해놨다. 그런데 수정하면서 오류가 생겼는데, 'size' 배열의 크기와 'purchaseHistroy' 배열에 대한 인덱스 관리에서 문제가 생긴건지, 잘못된 메모리 영역을 참조한 문제가 있는지 정확하게 파악되지 않아 수정이 필요한 부분이다.

- 테스트 결과 스크린샷

```
고객님의 키를 입력하세요 (cm): 170
고객님의 몸무게를 입력하세요 (kg): 65
고객님의 허리둘레를 입력하세요 (cm): 80

고객님의 신체 정보:
키: 170.00 cm
몸무게: 65.00 kg
허리둘레: 80 cm
```

[illegible]

2) 기능 구현

(2) 상품 및 사이즈 선택

- 입출력

옷구매창 표시 - 사용자 입력받기 - 선택한 제품에 따른 처리 - 이전 구매 내역 출력

purchaseProduct 함수 - 사이즈 입력 받기

- 설명

사용자에게 옷 구매 창을 표시한다. 구매창에는 각 제품에 대한 정보와 가격이 출력된다.

그리고 사용자에게 어떤 제품을 구매할지 선택하도록 입력을 받는다. 각 제품에는 정수
구값이 달려있고 'scanf_s' 함수를 사용하여 정수 값을 입력 받는다. 사용자가 선택한 제
품에 따라 'purchaseProduct' 함수를 호출하고, 해당 제품에 대한 정보를 구매 내역에 추
가한다. 이전 구매 내역을 출력하고 'numpurchase' 변수를 사용하여 현재까지의 구매 내
역을 반복하며 출력한다. 사용자에게 사이즈를 입력받는다.

- 적용된 배우 내용

4주차에 배웠던 switch(조건문을 사용해서 사용자가 선택한 제품 번호('choice')에 따라 다른 동작을 수행하도록 했다. 각 'case'는 사용자가 선택한 제품 번호에 대응되며, 해당하는 제품에 대한 처리를 수행한다. 또 함수 'purchaseProduct'를 호출하여 해당 제품을 구매하고, 구매 내역을 업데이트 한다. 이때, 'purchaseProduct' 함수에는 선택된 제품에 대한 정보뿐만 아니라 사용자의 정보와 이전 구매 내역을 전달한다.

만약 사용자가 유효한 범위의 제품 번호를 선택하지 않았을 경우('default'), 잘못된 선택이라는 메시지를 출력하고 프로그램을 종료한다.

- 코드 스크린샷

```
// 옷 구매 창 표시 및 옷 구매
printf("\n=== 옷 구매 창 ===\n");
printf("1. 와이드 밴딩 데님 바지 - 59000원\n");
printf("2. 원터 줄 와이드 슬렉스 - 54000원\n");
printf("3. 골덴 워싱 와이드 팬츠 - 56000원\n");
printf("4. 워싱 블랙 와이드 데님 - 46000원\n");
printf("5. 얼리 와이드 카고 데님 - 42000원\n");
printf("6. 와이드 하프 데님 쇼츠 - 34000원\n");

int choice;
printf("\n구매할 제품 번호를 선택하세요 (1-6): ");
scanf_s("%d", &choice, sizeof(choice));

// 선택한 제품에 따라 가격 계산 및 구매 내역에 추가
switch (choice) {
case 1:
    purchaseHistory[numPurchase].price = purchaseProduct("와이드 밴딩 데님 바지", 59000, &customer, purchaseHistory, &numPurchase);
    break;
case 2:
    purchaseHistory[numPurchase].price = purchaseProduct("원터 줄 와이드 슬렉스", 54000, &customer, purchaseHistory, &numPurchase);
    break;
case 3:
    purchaseHistory[numPurchase].price = purchaseProduct("골덴 워싱 와이드 팬츠", 56000, &customer, purchaseHistory, &numPurchase);
    break;
case 4:
    purchaseHistory[numPurchase].price = purchaseProduct("워싱 블랙 와이드 데님", 46000, &customer, purchaseHistory, &numPurchase);
    break;
case 5:
    purchaseHistory[numPurchase].price = purchaseProduct("얼리 와이드 카고 데님", 42000, &customer, purchaseHistory, &numPurchase);
    break;
case 6:
    purchaseHistory[numPurchase].price = purchaseProduct("와이드 하프 데님 쇼츠", 34000, &customer, purchaseHistory, &numPurchase);
    break;
default:
    printf("잘못된 선택입니다.\n");
    return 1;
}
```

```
// 옷 구매 함수
float purchaseProduct(char itemName[], float price, struct CustomerInfo* customer, struct PurchaseHistory* purchaseHistory, int* numPurchase) {
    struct PurchaseHistory purchase;
    strcpy_s(purchase.itemName, sizeof(purchase.itemName), itemName);
    purchase.price = price;

    printf("제품을 구매한 다른 고객분들이 선택한 사이즈의 평균입니다.");
    // 사용자에게 사이즈 추천
    printf("\n=== 사이즈 추천 ===\n");
    printf("-----|-----|-----|-----|");
    printf("사이즈 | 키(cm) | 몸무게(kg) | 허리둘레(cm) |");
    printf("-----|-----|-----|-----|\n");
    printf("S | 160 ~ 170 | 50 ~ 65 | 70 ~ 80 |");
    printf("-----|-----|-----|-----|\n");
    printf("M | 170 ~ 180 | 65 ~ 75 | 80 ~ 90 |");
    printf("-----|-----|-----|-----|\n");
    printf("L | 180 ~ 190 | 75 ~ 90 | 90 ~ 100 |");
    printf("-----|-----|-----|-----|\n");
    printf("XL | 190 이상 | 90 이상 | 100 이상 |");
    printf("-----|-----|-----|-----|\n");

    // 사용자에게 사이즈 선택
    printf("\n사이즈를 선택하세요 (S, M, L, XL): ");
    scanf_s("%2s", purchase.size, sizeof(purchase.size));

    // 옷 구매 정보 출력
    printf("%s를 구매하였습니다. 가격: $%.2f, 사이즈: %s\n", purchase.itemName, purchase.price, purchase.size);
}
```

2) 테스트 결과

(2) 상품 및 사이즈 선택

- 설명

사용자에게 옷 종류 및 가격을 보여줄 수 있는 상태창을 띄어준다, 각 제품에 정수값을

설정해두고 사용자에게 제품 번호를 입력받는다. 그리고 사용자는 자신의 키, 몸무게, 허리둘레를 적은 신체정보를 바탕으로 추천된 사이즈추천 표를 보고 사이즈를 선택하게 된다.

- 테스트 결과 스크린샷

```

=== 옷 구매 창 ===
1. 와이드 밴딩 데님 바지 - 59000원
2. 윈터 울 와이드 슬랙스 - 54000원
3. 골덴 워싱 와이드 팬츠 - 56000원
4. 워싱 블랙 와이드 데님 - 46000원
5. 델리 와이드 카고 데님 - 42000원
6. 와이드 하프 데님 쇼츠 - 34000원

구매할 제품 번호를 선택하세요 (1-6): 5
제품을 구매한 다른 고객분들이 선택한 사이즈의 평균입니다.
=== 사이즈 추천 ===

```

사 이 즈	키 (cm)	몸 무 게 (kg)	허 리 둘 레 (cm)
S	160 - 170	50 - 65	70 - 80
M	170 - 180	65 - 75	80 - 90
L	180 - 190	75 - 90	90 - 100
XL	190 이상	90 이상	100 이상

```

사이즈를 선택하세요 (S, M, L, XL): S
델리 와이드 카고 데님을 구매하였습니다. 가격: $42000.00, 사이즈: S

```

3) 기능 구현

(3)쇼핑 유지

- 입출력

사용자에게 계속해서 쇼핑할 것인지 물어보고, 사용자가 'Y' 또는 'y'를 입력하면 쇼핑을 계속하고, 다른 문자를 입력하면 쇼핑을 종료하는 부분이다. 이 부분의 입출력은 사용자로부터 문자를 입력받는 것으로 이루어져 있다.

사용자에게 계속 쇼핑할지 여부 확인
char continueShopping;

-설명

이 코드는 사용자로부터 'Y' 또는 'N' 중 하나를 입력받아, 'Y'를 입력하면 계속해서 쇼핑하도록 하고, 'N'을 입력하면 쇼핑을 종료하고 이전에 구매한 내역을 출력하는 기능을 수행한다. 먼저 사용자에게 계속해서 쇼핑할 것인지 묻는 안내 메시지를 출력하고 'scanf_s' 함수를 사용하여 사용자로부터 문자 하나를 입력받는다. 입력 시에 여분의 공백 문자를

무시하기 위해서 "" %c"" 형태로 입력 포맷을 저장하고 있다.

- 적용된 배운 내용

'do' 키워드로 시작하여 중괄호 안의 코드 블록을 실행한다.

'while (1)'은 항상 참인 조건으로 무한 루프를 의미한다.

사용자가 'return'을 통해 함수를 빠져나가지 않는 한 계속해서 반복하도록 한다.

- 코드 스크린샷

```
// 사용자에게 계속 쇼핑할지 여부 확인
char continueShopping;
printf("\n계속해서 쇼핑하시겠습니까? (Y/N): ");
scanf_s(" %c", &continueShopping, sizeof(continueShopping));

if (continueShopping != 'Y' && continueShopping != 'y') {
    printf("\n쇼핑을 종료합니다. 이전 구매 내역:\n");

    // 이전 구매 내역 출력
    for (int i = 0; i < numPurchase; ++i) {
        printf("제품명: %s, 가격: $%.2f, 사이즈: %s\n", purchaseHistory[i].itemName, purchaseHistory[i].price, purchaseHistory[i].size);
    }

    return;
}

while (1);
```

3) 테스트 결과

(3) 쇼핑 유지 및 사이즈 비교

- 설명

계속해서 쇼핑할건지에 대한 질문에

-> Y : 옷 구매창을 다시 띄어준다

-> N : 쇼핑을 종료한다는 메시지와 함께 이전 구매 내역 출력

- 테스트 결과 스크린샷

```
계속해서 쇼핑하시겠습니까? (Y/N): N

쇼핑을 종료합니다. 이전 구매 내역:
제품명: 윈터 울 와이드 슬랙스, 가격: $54000.00, 사이즈: S
제품명: 윈터 울 와이드 슬랙스, 가격: $54000.00, 사이즈: L
```

```
switch (choice) {
case 1:
    purchaseHistory[numPurchase].price = 59000; // 직접 가격 할당
    purchaseProduct("와이드 밴딩 데님 바지",
purchaseHistory[numPurchase].price, &customer, purchaseHistory,
&numPurchase):
```

```

        break;
    case 2:
        purchaseHistory[numPurchase].price = 54000; // 직접 가격 할당
        purchaseProduct("윈터 올 와이드 슬랙스", 
        purchaseHistory[numPurchase].price, &customer, purchaseHistory,
        &numPurchase);
        break;
    // 나머지 case들도 동일하게 수정

```

4. 계획 대비 변경 사항(x)

1) 상품 및 사이즈 선택 기능

- 이전

기능 중에서 이전 구매 내역을 볼 수 있는 기능만 존재, 제품을 구매하는 기능이 없었음

- 이후

사용자로부터 신체정보를 입력받고 나서 사용자에게 구입할 수 있는 옷 구매 창을 띄어 줄 수 있게 코드를 짰

- 사유

제품을 구매하는 기능없이 이전 구매 내역을 업데이트 시키는건 인과관계가 어긋남.

5. 프로젝트 일정

업무	11/3	11/10	11/17	11/24	12/1	12/8
제안서	----->					
기능1		----->				
기능2			----->			
기능3				-----	----->	
기능4~5						----->
업무	12/15			12/22		
보충 및 수정	-----			----->		
제출				O		

6. 느낀점

처음 공부해보는 언어로 스스로 기획해서 프로그램을 만드는 일이 쉽지는 않았다.

수업과 병행하면서 프로그램을 만드는 2달간의 프로젝트였는데, 기억 남는 것은 구조체의 활용 부분이었다. 구조체를 사용하기전에는 내가 쓴 코드들이 정리가 되지 않는 느낌이 컸었다. 그래서 구조체를 처음 접했을때 여러 가지의 데이터들을 묶어놓을 수 있어서 프로그램의 가독성과 유지보수성을 향상시킬 수 있는 강력한 도구라고 생각했다.

이외에도 수업시간에 배웠던 내용을 내가 만들고자 하는 프로그램에 적용시키는 것은 어렵지만 꽤 보람찬 작업이었던 것 같다. 아직은 완성이라고 하기에 애매한 프로젝트, 코드들이지만 나만의 코딩 스타일을 점점 구축해나가자 라는 다짐을 할 수 있게된 이번 개인 프로젝트였다.