

C프로그래밍및실습

대중교통 예매시스템

진척 보고서 #1

제출일자: 2023/12/09

제출자명: 김영재

제출자학번: 230114

1. 프로젝트 목표

1) 배경 및 필요성

현대 사회에서는 빠르게 이동하고 편리하게 교통수단을 이용하는 요구가 점차 증가함. 전통적인 예매 및 승차 시스템은 불편한 예매 프로세스와 긴 대기시간 등으로 고객의 교통 서비스의 만족도를 저하시키고 수익 감소를 야기할 수 있음. 때문에 문제를 해결하기 위해선 현대적이고 효율적인 대중 교통 예매 시스템이 필요함.

2) 프로젝트 목표

고객들이 원하는 목적지, 교통수단, 날짜, 좌석을 예매하고, 결제 내역 및 스케줄 확인을 보여 주는 프로그램을 만드는 것을 목표로 함.

3) 차별점

기존 프로그램들은 각 교통수단마다 예매 프로그램을 이용해 고객의 불편함을 야기할 수 있음. 이 프로그램은 다양한 교통 수단을 통합하여 예매하고 통합된 승차 티켓을 제공하는 것에 기존 프로그램과 차별점이 있음.

2. 기능 계획

1) 기능 1 (메인 기능 구현)

- 설명 : 고객의 요구에 따라 원하는 프로그램을 작동하는 기능

(1) 세부 기능 1 (고객 거주지 입력 및 변경)

- 설명 : 예매를 하기 위해 고객의 거주지를 입력 받거나 요구에 따라 변경한다.

(2) 세부 기능 2 (예매하기)

- 설명 : 고객이 희망하는 목적지와 교통수단을 입력 받고 가격을 계산해준다.

(3) 세부 기능 3 (결제 내역 및 예매 정보 확인)

- 설명 : 고객의 과거 예매 정보와 결제 내역을 보여준다.

(4) 세부 기능 4 (프로그램 종료)

- 설명 : 프로그램을 종료 시킨다

2) 기능 2 (대중 교통 예매)

- 설명 : 고객의 거주지에 맞춰 대중 교통을 예매하는 기능

(1) 세부 기능 1 (목적지 입력 받기)

- 설명 : 고객이 원하는 목적지(광역시만)를 입력한다.

(2) 세부 기능 2 (날짜 입력 받기)

- 설명 : 고객이 원하는 날짜를 입력 받는다.

(3) 세부 기능 3 (교통 수단 입력 받기)

- 설명 : 거주지와 목적지가 같을 경우, 택시와 지하철 중 고객이 원하는 교통수단을 입력 받고 다를 경우, 버스와 기차 중 원하는 교통수단을 입력 받는다

3. 진척사항

1) 기능 구현

(1) 예매하기

- 입출력 : 입력 - 목적지, 교통수단, 거리 ,날짜

출력 - "%s로 %dm 이동하여 %d원 결제되었습니다.",

"%d원 결제되었습니다."

- 설명 : 사용자로부터 목적지를 입력 받은 후 거주지와 목적지가 같을 경우 교통수단 및 거리를 입력 받고, 거주지와 목적지가 다를 경우 날짜 및 교통수단을 입력 받는다.

- 적용된 배운 내용 : 구조체, scanf_s 함수, 조건문, do-while문, strcmp 함수
- 코드 스크린샷

```

case 2: // 2. 예약하기
printf("원하시는 목적지를 입력해주세요.(단, 부산, 대구, 인천, 광주, 대전, 울산에서만)\n");
scanf_s("%s", reservations[reservationCount].destination, (int)sizeof(reservations[reservationCount].destination)); // 사용자로부터 목적지 입력받기
// 거주지와 목적지가 같을 경우
if (strcmp(reservations[0].location, reservations[reservationCount].destination) == 0) {
    do {
        printf("원하시는 교통수단을 선택해주세요.(택시, 지하철)\n");
        scanf_s("%s", reservations[reservationCount].transportation, (int)sizeof(reservations[reservationCount].transportation)); // 사용자로부터 원하는 교통수단 입력받기

        // 택시, 지하철을 입력하지 않았을 경우
        if (strcmp(reservations[reservationCount].transportation, "택시") != 0 && strcmp(reservations[reservationCount].transportation, "지하철") != 0) {
            printf("유효하지 않은 교통수단입니다. 다시 입력해주세요.\n");
        }
    } while (strcmp(reservations[reservationCount].transportation, "택시") != 0 && strcmp(reservations[reservationCount].transportation, "지하철") != 0);

    printf("원하는 목적지까지 걸리는 거리를 입력해주세요.(m 단위)\n");
    scanf_s("%d", &reservations[reservationCount].distance); // 사용자로부터 목적지까지 걸리는 거리 입력받기

    // 교통수단이 택시일 경우
    if (strcmp(reservations[reservationCount].transportation, "택시") == 0) {
        reservations[reservationCount].price = 4300 + ((reservations[reservationCount].distance / 150) * 100); // 기본요금 4300원에 150m당 100원 추가
        printf("%s도 %dm 이동하여 %d원 결제되었습니다.\n", reservations[reservationCount].transportation, reservations[reservationCount].distance, reservations[reservationCount].price);
    }

    // 교통수단이 지하철일 경우
    else if (strcmp(reservations[reservationCount].transportation, "지하철") == 0) {
        reservations[reservationCount].price = 1250; // 기본요금 1250원
        if (reservations[reservationCount].distance > 10000 && reservations[reservationCount].distance < 50000) {
            reservations[reservationCount].price += ((reservations[reservationCount].distance - 10000) / 5000) * 100; // 거리가 10km 초과 50km 미만 일 경우 5km당 100원 추가
            if (reservations[reservationCount].distance >= 50000) {
                reservations[reservationCount].price += ((reservations[reservationCount].distance - 50000) / 5000) * 100;
            } // 거리가 50km 이상 일 경우 5km당 100원 추가
        }
        printf("%s도 %dm 이동하여 %d원 결제되었습니다.\n", reservations[reservationCount].transportation, reservations[reservationCount].distance, reservations[reservationCount].price);
    }
    reservationCount++; // 예매 횟수 증가
}

// 거주지와 목적지가 다를 경우
else {
    // 사용자로부터 이동할 날짜 입력받기
    do {
        printf("이동하실 날짜를 입력해주세요.(년, 월, 일, 띄어쓰기 구분)\n");
        scanf_s("%d %d %d", &reservations[reservationCount].date[0], &reservations[reservationCount].date[1], &reservations[reservationCount].date[2]);

        if (reservations[reservationCount].date[0] != 2023 && reservations[reservationCount].date[0] != 2024) {
            printf("2023년~2024년 사이 날짜를 입력해주세요.\n");
        }
        if (reservations[reservationCount].date[1] > 12 && reservations[reservationCount].date[1] < 0) {
            printf("1월~12월 사이 날짜를 입력해주세요.\n");
        }
    } while (reservations[reservationCount].date[0] != 2023 && reservations[reservationCount].date[0] != 2024 || reservations[reservationCount].date[1] > 12 && reservations[reservationCount].date[1] < 0);

    do {
        printf("원하시는 교통수단을 선택해주세요.(고속버스, 기차)\n");
        scanf_s("%s", reservations[reservationCount].transportation, (int)sizeof(reservations[reservationCount].transportation)); // 사용자가 원하는 교통수단 입력받기

        if (strcmp(reservations[reservationCount].transportation, "고속버스") != 0 && strcmp(reservations[reservationCount].transportation, "기차") != 0) {
            // 고속버스, 기차를 입력하지 않았을 경우
            printf("유효하지 않은 교통수단입니다. 다시 입력해주세요.\n");
        }
    } while (strcmp(reservations[reservationCount].transportation, "고속버스") != 0 && strcmp(reservations[reservationCount].transportation, "기차") != 0);

    // 교통수단이 고속버스일 경우
    if (strcmp(reservations[reservationCount].transportation, "고속버스") == 0) {
        reservations[reservationCount].price = 20000;
        printf("%d원 결제되었습니다.\n", reservations[reservationCount].price);
    }

    // 교통수단이 기차일 경우
    else if (strcmp(reservations[reservationCount].transportation, "기차") == 0) {
        reservations[reservationCount].price = 50000;
        printf("%d원 결제되었습니다.\n", reservations[reservationCount].price);
    }
    reservationCount++; // 예매 횟수 증가
}
break;

```

(2) 결제 내역 및 예매 정보 확인

- 입출력 : 입력 - menu에 3 입력

출력 - "예매한 정보를 확인합니다.

----- 예매 정보 -----

예매 번호: %d

교통수단 : %s

이동 거리: %dm

결제 금액 : %d원

-----"

"----- 예매 정보 -----"

예매 번호 : %d

거주지 : %s

목적지 : %s

이동 날짜: %d년 %d월 %d일

교통수단 : %s

결제 금액: %d원

-----"

- 설명 : 메인 메뉴 선택에서 3번 선택시 예매한 정보를 출력해준다.
- 적용된 배운 내용 : switch문, 조건문, for문
- 코드 스크린샷

```
case 3:
    printf("예매한 정보를 확인합니다.\n");
    if (reservationCount > 0) {
        for (int i = 0; i < reservationCount; i++) {
            if (strcmp(reservations[0].location, reservations[i].destination) == 0) {
                printf("----- 예매 정보 ----- \n");
                printf("예매 번호: %d\n", i + 1);
                printf("교통수단: %s\n", reservations[i].transportation);
                printf("이동 거리: %d\n", reservations[i].distance);
                printf("결제 금액: %d원\n", reservations[i].price);
                printf("----- \n");
            }
            else {
                printf("----- 예매 정보 ----- \n");
                printf("예매 번호: %d\n", i + 1);
                printf("거주지: %s\n", reservations[0].location);
                printf("목적지: %s\n", reservations[i].destination);
                printf("이동 날짜: %d년 %d월 %d일\n", reservations[i].date[0], reservations[i].date[1], reservations[i].date[2]);
                printf("교통수단: %s\n", reservations[i].transportation);
                printf("결제 금액: %d원\n", reservations[i].price);
                printf("----- \n");
            }
        }
    }
    else {
        printf("예매된 정보가 없습니다.\n");
    }
    break;
```

2) 테스트 결과

(1) 예매 하기 (거주지 = 목적지)

- 설명 : 사용자로부터 목적지를 입력 받고 거주지와 일치할 경우 교통수단과 거리를 입력받는다.

- 테스트 결과 스크린샷

```
-----
메뉴를 입력해주세요
1. 거주지 변경
2. 예매하기
3. 결제 내역 및 예매 정보 확인
4. 종료
-----
2
원하시는 목적지를 입력해주세요.
(단, 부산, 대구, 인천, 광주, 대전, 울산에서만)
대전
원하시는 교통수단을 선택해주세요.(택시, 지하철)
택시
원하는 목적지까지 걸리는 거리를 입력해주세요.(m 단위로)
500
택시로 500m 이동하여 4600원 결제되었습니다.
```

(2) 예매 하기 (거주지 != 목적지)

- 설명 : 사용자로부터 목적지를 입력 받고 거주지와 일치하지 않을 경우 날짜와 교통수단을 입력받는다.

- 테스트 결과 스크린샷

메뉴를 입력해주세요

1. 거주지 변경
 2. 예매하기
 3. 결제 내역 및 예매 정보 확인
 4. 종료
-

2

원하시는 목적지를 입력해주세요.

(단, 부산, 대구, 인천, 광주, 대전, 울산 에서만)

인천

이동하실 날짜를 입력해주세요.(년, 월, 일, 띄어쓰기 구분)

2023 2 2

원하시는 교통수단을 선택해주세요.(고속버스, 기차)

고속버스

20000원 결제되었습니다.

(3) 최대 예매 수 까지 예매할 때

- 설명 : 예매를 10개까지 할 경우 프로그램을 종료 시킨다.
- 테스트 결과 스크린샷

메뉴를 입력해주세요

1. 거주지 변경
 2. 예매하기
 3. 결제 내역 및 예매 정보 확인
 4. 종료
-

2

원하시는 목적지를 입력해주세요.

(단, 부산, 대구, 인천, 광주, 대전, 울산 에서만)

울산

이동하실 날짜를 입력해주세요.(년, 월, 일, 띄어쓰기 구분)

2023 1 23

원하시는 교통수단을 선택해주세요.(고속버스, 기차)

기차

50000원 결제되었습니다.

최대 예매 가능 횟수 10개로 다 찼으므로 프로그램을 종료합니다.

(4) 예매 오류 (교통수단이 틀릴 경우)

- 설명 : 지정한 교통수단과 다른 값을 입력할 경우 오류 메시지와 함께 다시 입력받는다.

- 테스트 결과 스크린샷

메뉴를 입력해주세요

1. 거주지 변경
 2. 예매하기
 3. 결제 내역 및 예매 정보 확인
 4. 종료
-

2

원하시는 목적지를 입력해주세요.

(단, 부산, 대구, 인천, 광주, 대전, 울산에서만)

대전

원하시는 교통수단을 선택해주세요.(택시, 지하철)

버스

유효하지 않은 교통수단입니다. 다시 입력해주세요.

원하시는 교통수단을 선택해주세요.(택시, 지하철)

(5) 예매 오류 (날짜를 잘못 입력했을 경우)

- 설명 : 2023년~2024년 사이의 날짜를 입력하지 않을 경우 오류 메시지와 함께 다시 입력받는다.

- 테스트 결과 스크린샷

메뉴를 입력해주세요

1. 거주지 변경
 2. 예매하기
 3. 결제 내역 및 예매 정보 확인
 4. 종료
-

2

원하시는 목적지를 입력해주세요.

(단, 부산, 대구, 인천, 광주, 대전, 울산 에서만)

부산

이동하실 날짜를 입력해주세요.(년, 월, 일, 띄어쓰기 구분)

2025 1 1

2023년~2024년 사이 날짜를 입력해주세요.

(6) 결제 내역 및 예매 정보 확인

- 설명 : 저장 되어있는 예매 정보를 출력한다.

- 테스트 결과 스크린샷

3

예매한 정보를 확인합니다.

----- 예매 정보 -----

예매 번호: 1

거주지: 대전

목적지: 부산

이동 날짜: 2023년 11월 2일

교통수단: 고속버스

결제 금액: 20000원

----- 예매 정보 -----

예매 번호: 2

교통수단: 택시

이동 거리: 500m

결제 금액: 4600원

----- 예매 정보 -----

예매 번호: 3

거주지: 대전

목적지: 인천

이동 날짜: 2023년 2월 2일

교통수단: 고속버스

결제 금액: 20000원

----- 예매 정보 -----

예매 번호: 4

거주지: 대전

목적지: 대구

4. 계획 대비 변경 사항

1) 예매 제한

- 이전 : 예매 횟수에 제한을 두지 않았다.
- 이후 : 최대 예매 횟수를 10개로 두고 10번을 한 경우 오류 메시지와 함께 프로그램을 종료 한다.
- 사유 : 데이터 크기 제한과 프로그램 제한을 둘 필요가 있어서

5. 프로젝트 일정

업무		11/3	11/10	11/17	12/1	12/15
제안서 작성		완료				
기능1	세부기능1		완료			
	세부기능2					----->
	세부기능3				완료	
	세부기능4		완료			
기능2	세부기능1		완료			
	세부기능2					----->
	세부기능3				완료	