

할 일 관리 프로그램 보고서

232908 정찬민

1. 프로젝트 목적 및 배경

-프로젝트 목적

이 프로젝트의 목적은 사용자가 할 일을 관리하고 추적할 수 있는 간단한 TODO 리스트 애플리케이션을 개발하는 것이다. 이 애플리케이션은 C프로그래밍 수업에서 7주동안 학습했던 내용을 가지고, 사용자가 할 일을 추가, 삭제, 수정하고 목록으로 볼 수 있는 간단한 기능을 제공하여 사용자의 업무 및 일상 생활을 더 효율적으로 관리할 수 있도록 돕는 것이 주된 목표이다.

-프로젝트 배경

일상 생활 및 업무에서 할 일을 효율적으로 관리하는 것은 중요한 과제 중 하나이다. 현대 사회에서는 다양한 업무와 의무가 늘어나면서 할 일을 효율적으로 추적하고 관리하는 필요성이 더 커졌다. 이에 따라 개인 및 조직에서 할 일 관리를 돕는 애플리케이션의 수요가 증가하고 있다.

본 프로젝트는 이러한 배경에서 사용자들이 간편하게 할 일을 관리할 수 있는 도구를 제공하고자 합니다. 간단한 목록 관리 애플리케이션을 통해 사용자가 할 일을 추가, 삭제, 수정하며 효과적으로 추적할 수 있게 될 것입니다.

이 프로젝트를 통해 사용자들은 더 효율적으로 일상 생활과 업무를 관리할 수 있으며, 간단하고 직관적인 사용자 경험을 제공하는 애플리케이션의 중요성을 이해할 수 있을 것이다.

2. 요구사항

-사용자 요구사항

프로젝트의 요구사항은 사용자의 요구사항을 충족시키기 위해 구현해야 하는 구체적인 작업과 기능을 기반으로 정의된다. 이 프로젝트의 기능 요구사항은 다음과 같다.

1. 할 일 추가 기능:

- 사용자는 할 일을 목록에 추가할 수 있어야 합니다.
- 목록에 할 일을 추가하려면 사용자는 할 일 내용을 입력해야 합니다.

2. 할 일 삭제 기능:

- 사용자는 목록에서 특정 할 일을 선택하여 삭제할 수 있어야 합니다.
- 삭제한 후 해당 할 일은 목록에서 완전히 제거되어야 합니다.

3. 목록 보기 기능:

- 사용자는 현재 목록의 모든 할 일을 볼 수 있어야 합니다.
- 목록을 효과적으로 출력하여 사용자에게 편리한 방식으로 보여줘야 합니다.

4. 할 일 수정 기능:

- 사용자는 목록에서 특정 할 일을 선택하여 수정할 수 있어야 합니다.
- 수정할 때는 기존 내용을 표시하고 새로운 내용을 입력할 수 있어야 합니다.

프로젝트는 사용자가 할 일을 관리하는 간단한 TODO 리스트 애플리케이션을 개발하는 주요 동기와 목표를 나타내며, 이러한 요구사항을 충족시키는 데 프로젝트가 초점을 맞추고 있다.

3. 설계 및 구현

```
// 입력에 따른 기능 수행
switch (choice) {
case 1: // 할 일 추가
    printf("할 일을 입력하세요 (공백 없이 입력하세요): ");
    scanf_s("%s", tasks[taskCount], (int)sizeof(tasks[taskCount]));
    printf("할 일 \"%s\"가 저장되었습니다.\n", tasks[taskCount]);
    taskCount++;
    break;
```

1. 할 일 추가 기능

입력값: taskCount = 현재 작업 수를 나타내는 변수. 이 변수는 새로운 할 일이 목록에 추가 될 때 증가된다.

tasks = 할 일 목록을 저장하는 2차원 배열. 이 배열은 사용자로부터 입력받은 새로운 할 일 내용을 저장한다.

반환값: 없음

결과: 사용자로부터 입력받은 할 일이 tasks 배열에 저장됩니다.

설명: 이 함수는 사용자로부터 입력을 받아 새로운 할 일을 목록에 추가하는 역할을 한다. 함수 내에서는 사용자로부터 입력받은 할 일을 tasks 배열에 저장하고 taskCount 값을 증가시켜 새로운 할 일을 목록에 추가한다.

```
case 2: // 할 일 삭제
    printf("삭제할 할 일의 번호를 입력해주세요. (부터 시작):\n");
    scanf_s("%d", &delIndex);
    if (delIndex > taskCount || delIndex <= 0) { //사용자 입력값에 범위에서 벗어났을때
        printf("삭제 범위가 벗어났습니다.\n");
    }
    else {
        printf("Id %d : 할 일을 삭제합니다.\n", delIndex, tasks[delIndex - 1]);
        strcpy_s(tasks[delIndex - 1], sizeof(tasks[delIndex - 1]), "");
        // 비어 있는 칸을 메우기 위해 뒤에 있는 칸들 내리기
        for (int i = delIndex; i < taskCount + 1; i++) {
            strcpy_s(tasks[i - 1], sizeof(tasks[i]), tasks[i]);
        }
        taskCount -= 1;
    }
    break;
```

2. 할 일 삭제 기능

입력값: delIndex = 삭제할 할 일의 인덱스. 사용자가 입력한 값으로, 삭제할 할 일을 식별하는 데 사용된다.

반환값: 없음

결과: 선택한 할 일은 목록에서 삭제된다.

delIndex로 지정한 인덱스의 할 일이 삭제되고, 목록은 필요에 따라 정리된다.

설명: 이 기능은 사용자가 선택한 할 일을 목록에서 삭제한다. 사용자가 입력한 삭제할 할 일의 번호를 받아 delIndex로 저장한 후, 범위를 확인하여 삭제할 수 있는지 확인한다. 삭제된 할 일의 내용은 strcpy_s 함수를 사용하여 빈 문자열로 대체하고, 필요에 따라 다른 할 일들을 한 칸씩 앞으로 옮겨 정리한다. 마지막으로 taskCount를 1 감소시켜 현재의 할 일 수를 업데이트한다.

```
case 3:
    printf("할 일 목록\n"); // 할 일의 목록 보여주기
    for (int i = 0; i < taskCount; i++) {
        printf("%d. %s\n", i + 1, tasks[i]);
    }
    printf("\n");
    break;
```

3. 할 일 목록 보기 기능

입력값: taskCount = 현재 작업 수를 나타내는 변수. 이 변수는 목록을 보여주는 데 사용된다.

tasks = 할 일 목록을 저장하는 2차원 배열. 이 배열은 현재 목록을 표시하는 데 사용된다.

반환값: 없음

결과: 현재 목록에 있는 모든 할 일이 출력된다.

사용자에게 현재 목록의 할 일을 보여줌으로써 목록의 내용을 확인할 수 있게 한다.

설명: 이 기능은 현재 목록에 있는 모든 할 일을 사용자에게 보여주는 역할을 한다. for 반복문을 사용하여 목록을 순회하며 각 할 일의 번호와 내용을 출력한다. 목록이 보여지면 사용자는 현재의 할 일 목록을 확인할 수 있다.

```
case 5: // 할 일 수정
    printf("수정할 할 일의 번호를 입력해주세요 (1부터 시작): ");
    scanf_s("%d", &changeIndex);
    if (changeIndex > taskCount || changeIndex <= 0) {
        printf("수정 범위가 벗어났습니다.\n");
    }
    else {
        printf("현재 내용: %s\n", tasks[changeIndex - 1]);
        printf("새로운 할 일 내용을 입력하세요 (공백 없이 입력하세요): ");
        scanf_s("%s", tasks[changeIndex - 1], (int)sizeof(tasks[changeIndex - 1]));
        printf("%d. 할 일이 수정되었습니다: %s\n", changeIndex, tasks[changeIndex - 1]);
    }
    break;
```

4. 할 일 수정 기능

입력값: tasks = 할 일 목록을 저장하는 2차원 배열. 이 배열은 수정할 할 일 내용을 저장하는 데 사용된다.

changeIndex = 수정할 할 일의 인덱스. 사용자가 입력한 값으로, 수정할 할 일을 식별하는 데 사용된다.

반환값: 없음

결과: 사용자로부터 입력받은 새로운 내용으로 선택한 할 일이 수정된다.

결과 메시지가 사용자에게 표시되어, 사용자는 선택한 할 일이 수정되었음을 확인한다.

설명: 이 함수는 사용자가 선택한 할 일을 수정하는 역할을 한다. 함수 내에서는 changeIndex로 지정된 인덱스의 할 일을 사용자로부터 새로운 내용을 입력받아 수정한다.

4. 테스트

1. 할 일 추가

```
TODO 리스트 시작!
-----
메뉴를 입력해주세요.
1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료
5. 할 일 수정
현재 할 일 수 = 0
-----
1
할 일을 입력하세요 (공백 없이 입력하세요): study
할 일 study가 저장되었습니다
```

2. 할 일 목록

```
-----
3
할 일 목록
1. study
```

3. 할 일 삭제

```
-----
2
삭제할 할 일의 번호를 입력해주세요. (1부터 시작): 1
1. eating : 할 일을 삭제합니다.
```

4. 할 일 수정

```
-----
5
수정할 할 일의 번호를 입력해주세요 (1부터 시작): 1
현재 내용: study
새로운 할 일 내용을 입력하세요 (공백 없이 입력하세요): eating
1. 할 일이 수정되었습니다: eating
```

5. 최종 테스트

```
메뉴를 입력해주세요.
1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료
5. 할 일 수정
현재 할 일 수 = 1
-----
5
수정할 할 일의 번호를 입력해주세요 (1부터 시작): 1
현재 내용: study
새로운 할 일 내용을 입력하세요 (공백 없이 입력하세요): eating
1. 할 일이 수정되었습니다: eating
-----
메뉴를 입력해주세요.
1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료
5. 할 일 수정
현재 할 일 수 = 1
-----
2
삭제할 할 일의 번호를 입력해주세요. (1부터 시작): 1
1. eating : 할 일을 삭제합니다.
-----
메뉴를 입력해주세요.
1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료
5. 할 일 수정
현재 할 일 수 = 0
-----
1
할 일을 입력하세요 (공백 없이 입력하세요): study
할 일 study가 저장되었습니다
```

5. 결론 및 느낀점

본 프로젝트는 사용자가 할 일을 관리하고 효율적으로 처리할 수 있는 간단한 목록 관리 프로그램을 개발하는 데 중점을 두었다. 프로젝트의 목적은 사용자가 할 일을 추가, 삭제, 수정하고 현재의 목록을 확인할 수 있는 기능을 제공하여 편리함을 느낄 수 있게 하는 것 이었다. C언어는 1학기때 배웠던 파이썬과 비슷한 것 같으면서도 다른점도 많았던 것 같다. 그래서 처음에는 많이 해매고 어려워했으나 프로젝트를 끝낸 지금은 조금 성장한 기분이 든다. 아직도 많이 부족하지만, 프로젝트를 통해 프로그래밍 언어와 배열, 함수, 조건문, 루프 등의 프로그래밍 기본 개념을 익히고, 사용자 요구사항을 충족시키기 위한 기능을 구현하는 연습이 되었다.