도서관리 시스템 고도화(동적메모리)

한국기술교육대학교
온라인평생교육원

학습내용

- 프로그램 설계
- 코드 분석

학습목표

- 목표로 하는 시스템에서 요구되는 자료구조를 설계할 수 있다.
- 사용 가능한 라이브러리 함수를 개발에 적용할 수 있다

프로그램 설계

1 고도화 개요

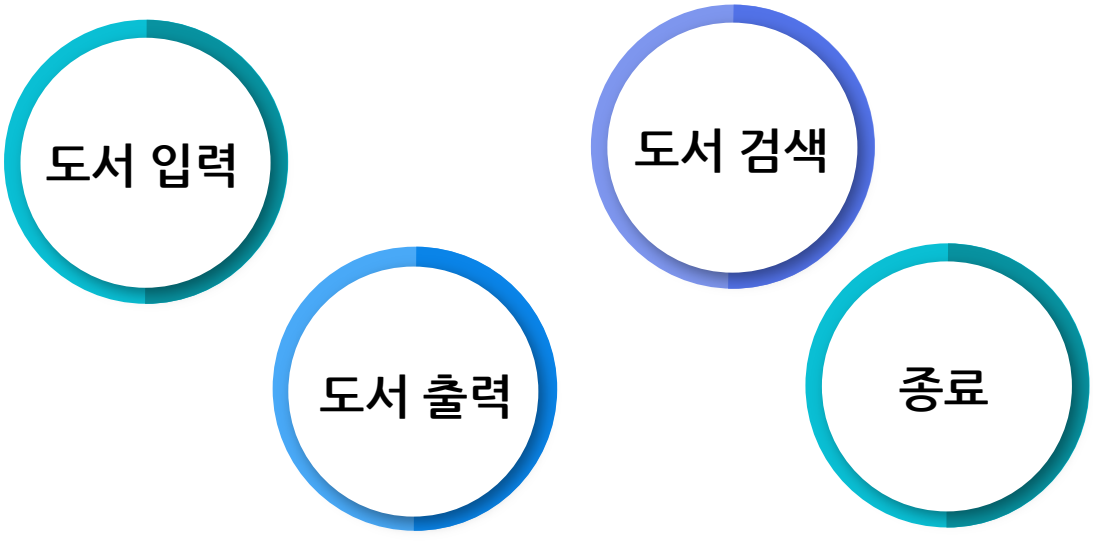
1. 베스트셀러 Top3 도서의 제목을 출력하는 기능
2. 도서 데이터를 동적할당을 이용하여 저장

프로그램 구현 시 처리 대상 자료

- 도서명
- 가격
- 코드
- 저자
- 판매수량
- 베스트셀러

2 기능 정의

1 메뉴



프로그램 설계

2 기능 정의

2 기능

- 도서 입력
- 도서 출력
- 제목 검색
- 저자 검색(검색 기능 키워드 검색)
- 출판연도별 목록 출력
- 베스트셀러 출력

3 고도화 내용

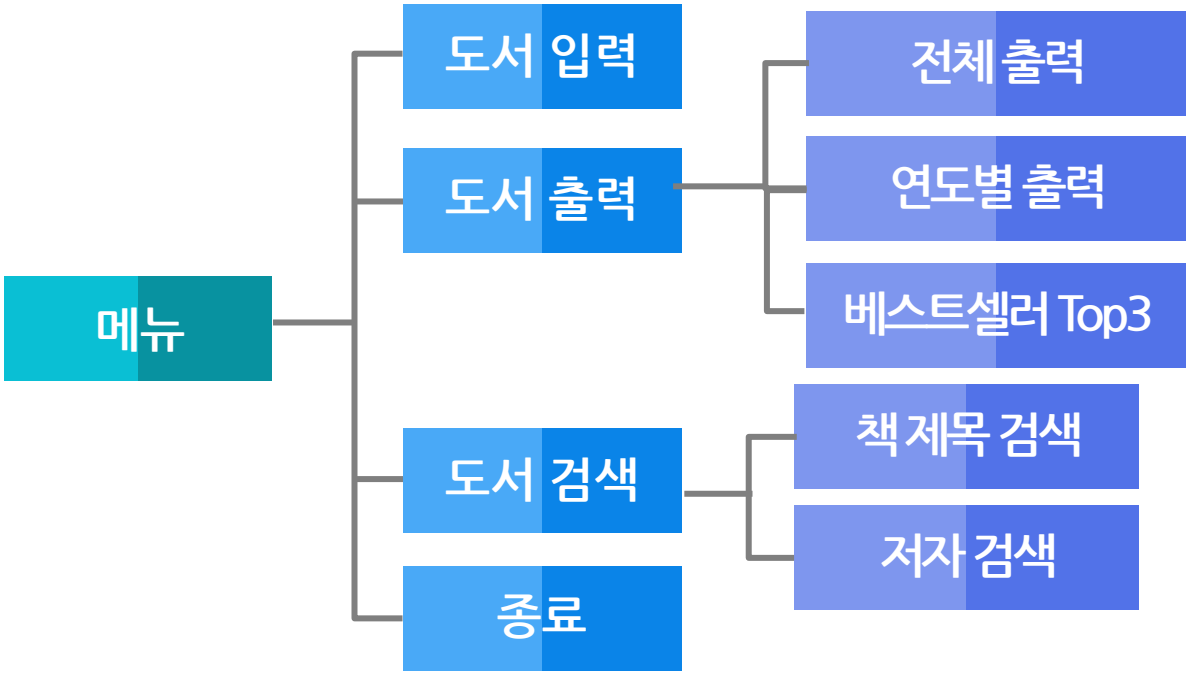
1. 베스트셀러 Top3 출력
2. 동적메모리를 할당하여 저장
3. 연속 입력 기능, 출력 메뉴 연속 사용 기능

프로그램 설계

2 기능 정의

4 고도화 방법

베스트셀러 출력	메모리 공간 활용	연속 입·출력
• qsort함수를 이용한 판매수량 정렬	• 동적 메모리 할당을 이용한 요구 메모리 축소	• 무한 반복 루프로 구현



프로그램 설계

3 자료구조 정의

1 자료

항목	구분	내용
도서명	char	bookTitle
저자	char	bookAuthor
가격	int	bookPrice
판매수량	int	bookSale
코드	char	bookCode(xxxx-xxx)

2 자료구조

```
struct book {
   char bookTitle[50];
   char bookAuthor[20];
   int bookPrice;
   int bookSale;
   char bookCode[8];
};
```

코드 분석

1 동적할당

```
int main()
{
        int sel, totalBCnt=0;
        BOOK myBook[100];
```

```
int main()
{
        int sel, totalBCnt=0;
        BOOK *myBook;
        myBook = (BOOK *)calloc(2,sizeof(BOOK));
```

```
typedef struct book{
        char bookTitle[50];
        char bookAuthor[20];
        int bookPrice;
        int bookSale;
        char bookCode[9];
}BOOK;
```

코드 분석

1 동적할당

```
BOOK* inBook(BOOK* mb, int* cn)
{
		char yn;
		int number;
		int cnt;
		cnt = *cn;
		do{
				while(1){
						mb = (BOOK
*)realloc(mb,sizeof(BOOK)*(cnt+2));
						중략

		}while(yn == 'y');
		*cn = cnt;
		return mb;
}
```

2 베스트셀러 출력

```
case 3 :
		qsort(mb, cnt+1, sizeof(BOOK), compare);

		printf("₩n----------------------------------₩n");
		printf("----------BestSeller Top 3----------");
		printf("₩n----------------------------------₩n");
		for (int i = 0; i < 3 ; i++)
		{
		    printf("Top %d : %30s  Sale: %d₩n", i+1, mb[i].bookTitle,
mb[i].bookSale );
		}

		printf("₩n----------------------------------₩n");
		break;
```

코드 분석

2 베스트셀러 출력

```
int compare(const void *a, const void  *b)
{
    BOOK* ptr_a = (BOOK *)a;
    BOOK* ptr_b = (BOOK *)b;

    if (ptr_a->bookSale < ptr_b->bookSale) return 1;
    else if (ptr_a->bookSale == ptr_b->bookSale) return 0;
    else return -1;
}
```

3 연속 입·출력

```
#include <conio.h>
BOOK inBook(BOOK* mb, int* cn)
{
                char yn;
                do{
                                while(1){
                                                printf("₩n책코드 : ");
                                중략
                                printf("₩n 계속 입력하시겠습니까?(Y/N) ");
                                yn = getche();
                                cnt++;
                }while(yn == 'y');
                *cn = cnt;
                return mb;
}
```

학습정리

1. 프로그램 설계

- 프로그램 개발 시 작업 목표에 맞게 사용할 라이브러리 함수와 자료구조를 결정해야 함
- 동적할당은 프로그램 실행 시 할당하는 메모리 공간이다.
- 함수포인터를 이용하여 정렬함수를 사용할 수 있다.

2. 코드 분석

- strtok는 토큰을 이용하여 문자열을 분리할 수 있는 함수임
- strcspn은 특정 문자로 이루어진 문자열인지 검사하는 것이 가능한 함수임
- strstr은 키워드 검색에 유용한 함수임