

9장 핵심



차례

- 함수를 정의하고 선언하기
- 변수의 존재기간과 접근범위 1 : 지역변수
- 변수의 존재기간과 접근범위 2 : 전역변수, static 변수
- 재귀함수에 대한 이해



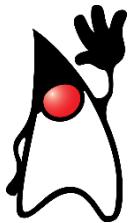
함수 선언(원형)

3

- 함수 선언(원형) : 컴파일러에게 함수에 관한 정보(이름, 반환형, 매개변수)를 미리 알리는 것

반환형 함수이름(매개변수1, 매개변수2, ...); // 함수헤더;

- 컴파일러가 함수 호출 문장을 컴파일할 때 함수에 대한 정보가 있어야 문법이 맞는지 체크할 수 있다.
- 함수 선언이 함수 호출보다 먼저 나와야 한다. 그렇지 않으면 컴파일 에러(경고) 발생
- 함수 정의가 호출보다 앞에 있다면 함수 선언은 필요 없다.



함수 선언(언형)

4

- 함수 정의가 호출보다 먼저 나오는 경우 -> 선언 필요없음(정의에 모든 정보 포함됨)

```
#include <stdio.h>
int square(int n)          // 함수 정의
{
    return (n * n);
}
int main(void)
{
    ...
    result = square(i);    // 함수 호출
    ...
}
```



함수 선언(언형)

5

- ▣ 함수 정의가 호출보다 나중에 나오는 경우-> 호출 전에 선언 해야함

```
#include <stdio.h>
int square(int n);           // 함수 선언
int main(void)
{
    ...
    result = square(i);      // 함수 호출
    ...
}
int square(int n)            // 함수 정의
{
    return (n * n);
}
```



함수 선언(언형)

6

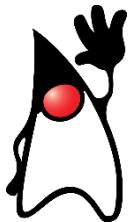
- 아래 예제와 같은 경우 에러발생

```
#include <stdio.h>

void f1()          // 함수 정의
{
    f2();          // error : 함수 f2에 대한 선언이 없으므로 컴파일 에러
}

void f2()          // 함수 정의
{
    f1();
}

int main(void)
{
    f1();          // 함수 호출
    f2();          // 함수 호출
}
```

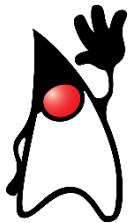


함수 선언(언형)

7

- 아래 예제와 같은 경우 에러발생

```
#include <stdio.h>
void f2()          // 함수 정의
{
    f1();          // error : 함수 f1에 대한 선언이 없으므로 컴파일 에러
}
void f1()          // 함수 정의
{
    f2();
}
int main(void)
{
    f1();          // 함수 호출
    f2();          // 함수 호출
}
```



함수 선언(언형)

8

- 일반적으로 함수 선언이 먼저 나오고 정의는 메인함수 다음에 온다

```
#include <stdio.h>
void f1();           // 함수 선언
void f2();           // 함수 선언
int main(void)
{
    f1();           // 함수 호출
    f2();           // 함수 호출
}
void f1()           // 함수 정의
{
    f2();
}
void f2()           // 함수 정의
{
    f1();
}
```



함수 선언(언형)

9

- 일반적으로 함수 선언->호출->정의 순으로 작성한다.

```
#include <stdio.h>
// 함수 선언
int main(void)
{
    ...
    // 함수 호출
    ...
}
// 함수 정의
```



함수 선언과 헤더 파일

10

- printf, scanf와 같은 라이브러리 함수를 호출할 때도 선언을 먼저 해야한다.
- 라이브러리 함수들의 선언은 헤더파일에 작성되어 있으므로 라이브러리 함수를 사용(호출)할 때 헤더 파일을 포함하면 된다.

```
/* 두개의 숫자의 합을 계산하는 프로그램 */
#include <stdio.h>

int main(void)
{
    int n1; /* 첫번째 숫자 */
    int n2; /* 두번째 숫자 */
    int sum; /* 두개의 숫자의 합을 저장 */

    printf("첫번째 숫자를 입력하시오:");
    scanf("%d", &n1);

    printf("두번째 숫자를 입력하시오:");
    scanf("%d", &n2);

    sum = n1 + n2;
    printf("두수의 합: %d", sum);

    return 0;
}
```

```
/*
 * stdio.h - definitions/declarations for
 * standard I/O routines
 *
 ****/
...
_CRTIMP int __cdecl printf(const char
*, ...);
...
_CRTIMP int __cdecl scanf(const char
*, ...);
...
```

stdio.h

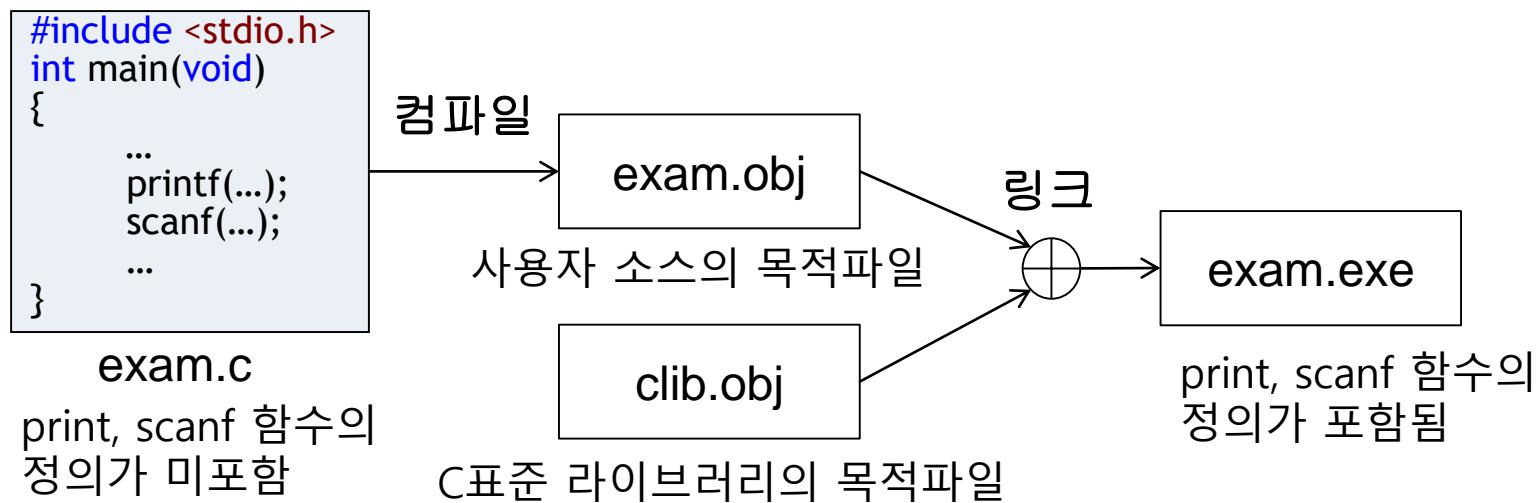
add.c



링크 과정의 이해

11

- 라이브러리함수를 메인함수에서 호출시 선언을 포함하는 헤더파일을 포함해야함
 - ▣ printf, scanf -> stdio.h
 - ▣ sqrt, sin, cos -> math.h
- 라이브러리함수의 정의도 프로그램에 포함되어야함 -> 소스파일에는 미포함되고 실행파일에 포함됨
- 링크과정에서 라이브러리함수의 목적파일과 소스코드의 목적파일이 결합하여 실행파일을 만듦->라이브러리는 목적파일 형태로 제공됨



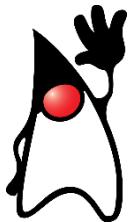


예제 1

12

```
// 정수의 제곱을 계산하는 함수 예제
#include <stdio.h>
int square(int n);           // 함수 선언
int main(void)
{
    int value, result;
    printf("정수를 입력하시오 : ");
    scanf("%d", &value);
    result = square(i);      // 함수 호출
    printf("%d의 제곱은 %d입니다. \n", value, result);
    return 0;
}
int square(int n)            // 함수 정의
{
    return (n * n);
}
```

정수를 입력하시오: 3<엔터>
3제곱은 9입니다.



예제 2

13

```
#include <stdio.h>
int get_integer();           // 함수선언
int main(void)
{
    int x, y, result;
    x = get_integer();      // 함수 호출
    y = get_integer();      // 함수 호출
    result = x + y;
    printf("두수의 합 = %d \n", result);
    return 0;
}
int get_integer() // 함수정의
{
    int value;
    printf("정수를 입력하시오 : ");
    scanf("%d", &value);
    return value;
}
```

정수를 입력하시오: 10<엔터>
정수를 입력하시오: 20<엔터>
두수의 합 = 30



예제 3

14

```
// 두수 중에서 큰 수를 찾는 함수 예제
#include <stdio.h>
int get_max(int x, int y); // 함수 선언
int main(void)
{
    int a, b,result;
    printf("두개의 정수를 입력하시오: ");
    scanf("%d %d", &a, &b);
    result = get_max(a, b);
    printf("두수 중에서 큰 수는 %d입니다.\n", result); // 함수 호출
    return 0;
}
int get_max(int x, int y) // 함수 정의
{
    if( x > y ) return x;
    else return y;
}
```

두개의 정수를 입력하시오: 2 3<엔터>
두 수 중에서 큰 수는 3입니다.



- 키보드로부터 2개의 정수를 받아 곱을 계산하는 코드를 작성하라.
단, 2 개의 정수의 곱을 계산하는 부분을 함수로 작성하여 아래 결과처럼 실행되도록 코드를 작성하라. 함수의 선언, 호출, 정의를 모두 사용할 것

```
정수를 입력하세요:7<엔터>
정수를 입력하세요:9 <엔터>
곱은 63입니다.
```



실습과제 2

16

- 키보드로부터 국어, 영어, 수학 점수를 입력 받아 평균값을 계산하는 프로그램을 작성하라. 단, 3과목 점수의 평균값을 구하는 부분을 함수로 만들어서 아래처럼 동작하도록 함수 선언, 호출, 정의를 모두 사용하라.

```
국어점수를 입력하세요:70 <엔터>
영어점수를 입력하세요:90 <엔터>
수학점수를 입력하세요:80 <엔터>
평균값은 70점입니다.
```



실습과제 3

17

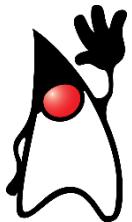
- 정수 2개 중에 최대값을 구하는 함수 `get_max`, 최소값을 구하는 함수 `get_min`함수를 작성하여 아래 결과처럼 동작하는 코드를 작성하라. 함수의 선언, 호출, 정의를 모두 사용할 것.

첫번째 정수를 입력하시오:20<엔터>

두번째 정수를 입력하시오:30<엔터>

최대값은 30입니다.

최소값은 20입니다.



과제제출방법

18

- 소스코드, 라인단위의 주석, 실행결과를 포함하는 pdf파일을 작성한 후 eclass 과제 게시판에 업로드, **반드시 하나의 pdf파일로 업로드할 것**
- 기한 : 과제 게시판에 마감시간 참조
- 실행결과를 캡쳐할 때 글자를 알아보기 쉽게 확대해서 캡쳐할 것.
- 소스코드의 첫 부분은 아래처럼 제목, 날짜, 작성자(학번, 이름)를 작성할 것

```
// *****
// 제 목 : 정수 4개의 평균을 구하는 프로그램
// 날 짜 : 2023년 9월10일
// 작성자 : 15010101 홍길동
// *****
```

```
// 소스코드 작성
```