

Java를 알고 C배우기

컴퓨터프로그래밍3

week 3-4 지역변수

2022.1학기
충남대 조은선

지역변수

- ▶ 함수 내에 선언되는 지역변수
 - ▶ 함수 내에서만 접근 가능
 - ▶ 호출될 때마다 새롭게 할당-리턴되면 지역변수는 소멸
 - ▶ 하나의 지역(함수) 내에 동일한 이름의 변수 선언 불가능
 - ▶ 다른 지역에 동일한 이름의 변수 선언 가능
- ▶ 중괄호 내에 선언되는 지역변수
 - ▶ 중괄호 진입할 때 할당되고 빠져나가면 소멸
- ▶ 형식 인자도 지역변수와 비슷
 - ▶ 선언된 함수 내에서만 접근 가능
 - ▶ 함수가 리턴되면 저장되었던 값이 사라짐

```
int func1 ( int a ) {  
    int i;  
    ...  
    for (i = 0; i <10; i++) {  
        int d = i*2;  
        printf("%d ", d);  
    }  
    // printf("%d ", d); 오류
```

전역변수

▶ 특징

- ▶ 함수 외부에 선언됨
- ▶ 프로그램의 시작과 동시에 메모리 공간에 할당
- ▶ 별도의 값을 초기화 하지 않으면 0으로 초기화
- ▶ 프로그램 전체 영역 어디든 접근 가능함
- ▶ 지역변수 이름이 전역변수 이름과 동일할 때
 - ▶ 가능하긴 하다
 - ▶ 지역변수 이름을 우선으로 한다

▶ 단점

- ▶ 너무 많이 쓰지 말 것, 가독성 신뢰성 저하

```
int i=100; // (ㄱ)
int func1() {
    int i=1;    // (ㄴ)
    ...
    printf("%d \n", i); // 1 출력
    i++; //(ㄴ)의 i가 2로 증가됨
    ...
}

void func2() {
    i++; //(ㄱ)의 i가 감소됨
}

void func3() {
    func2();
    printf("%d \n", i); // 101 출력
}
```

static 변수

▶ 특징

- ▶ 선언된 함수 안에서만 접근 가능
 - ➔ 지역변수와 동일
- ▶ 1회만 초기화되고 프로그램 종료시까지 메모리 공간에 있음
 - ➔ 전역변수와 동일

```
int newID () {  
    static int i=0;  
    return i++;  
    ...  
}  
  
int func1() {  
    printf("%d \n", newID()); // 0  
    printf("%d \n", newID()); // 1  
    printf("%d \n", newID()); // 2  
    ...  
}
```

지역 참조 환경(local referencing environment)

- ▶ 함수의 지역 참조환경(local referencing environment)
 - ▶ 지역변수 위주로 구성됨
 - ▶ 형식인자는 이 때 지역 변수와 동일하게 간주함
- ▶ 실인자와 형식인자에 대응되는 절차는 지정연산 (=)으로 보면 됨

```
int main()  
{  
    int foo(int, int);  
  
    ...  
    bigger = foo(a, b);  
    ...  
}
```

호출시 **s=a; t=b;**
실행된다고 봐야함

리턴 시 **bigger = big;**
실행된다고 봐야함

```
int foo(int s, int t)  
{  
    int big;  
  
    big = s;  
    if (big < t)  
        big = t;  
    return big;  
}
```

Quiz

▶ 다음 (1)~(5)중 중 1을 출력하는 것을 모두 고르시오.

```
#include <stdio.h>
int i=0;
void func1() {
    int i = 100;
    i++;
    printf("%d", i);        // (1)
}
void func2() {
    printf("%d", i);        // (2)
    i++;
    printf("%d", i);        // (3)
}
int main() {
    func1();
    printf("%d", i);        // (4)
    func2();
    printf("%d", i);        // (5)
    return 0;
}
```