

Java를 알고 C배우기

컴퓨터프로그래밍3

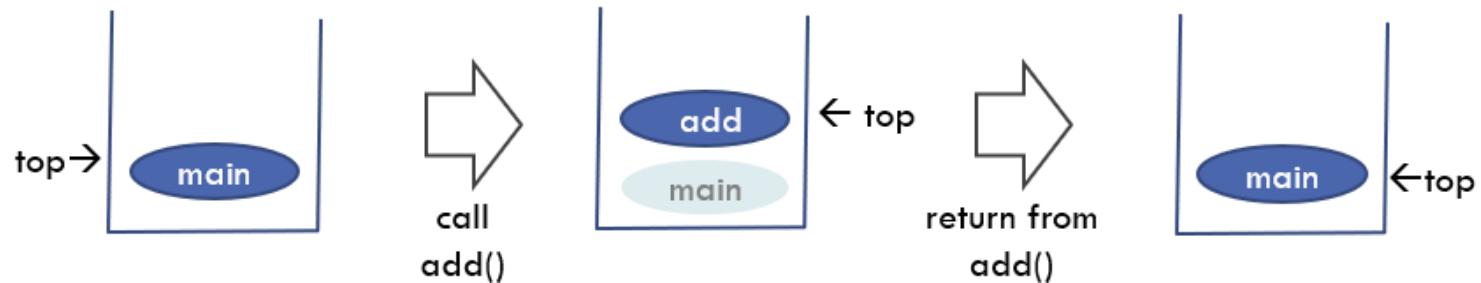
week 1 1-1 동적 메모리-Stack과 Heap

2022.1학기
충남대 조은선

복습 전에 공부했던 스택 (4장)

▶ 스택과 지역변수

- ▶ 함수 호출 후 리턴되면, 정의-사용했던 모든 지역 변수들이 사라짐



▶ 함수 간에 3가지 정보교환 방법

만일 함수 main 에서 자신의 지역변수 변수 x의 값을 n 증가한 값으로 바꾸는 함수 add를 만들려면

▶ ~~add(x,n);~~

// 안 된다. call by value

▶ 리턴값: x = add(x, n);

// 가능, x의 값을 n 증가한 값을 리턴, 리턴값을 x에 지정

▶ 인자(매개변수): add(&x, n);

// 가능, 함수 add에서 변수 x 값을 x+n 으로 바로 바꿈 (x+=n)

▶ 전역변수: 이런 int x는 전역변수로 둘 수도... // 가능

고민해 볼 문제

- ▶ 아래와 같은 함수에서 문제점은?

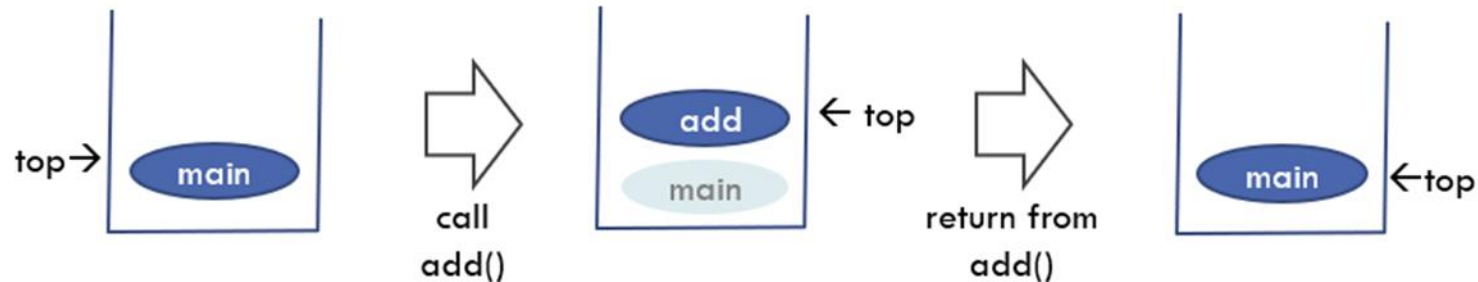
// scanf 로 숫자를 입력 받아서, 인자로 받은 n만큼 더하여 리턴

```
int add1(int n) {  
    int y;  
    scanf("%d", &y);  
    y = y + n;  
    return y;  
}
```

// 사용할 때는 이렇게 int x = add1(10);

```
int * add2(int n) {  
    int y;  
    scanf("%d", &y);  
    y = y + n;  
    return &y;  
}
```

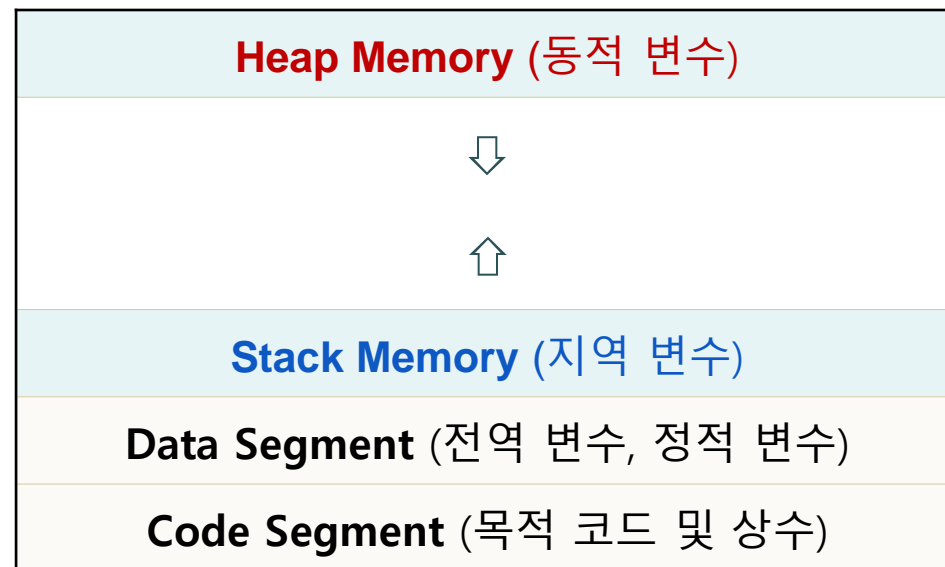
// 사용할 때는 이렇게 int x = *(add2(10));



NOTE: add() 리턴 후에는 y가 사라짐!

(거의 모든 PL에서의) 메모리 모델: Stack vs. Heap

- ▶ **Stack**: 지역변수, 인자, 리턴 주소
- ▶ **Heap**:
 - ▶ 변수와 비슷하지만 변수 이름이 없는 메모리 조각 (“동적 변수”, “무명 변수”라고도 함)
 - ▶ 전역변수처럼 함수와 독립적으로 존재
 - ▶ 전역변수와 달리 동적으로 생성/제거
 - ▶ “쓸 메모리 조각을 만들어라!” - `malloc()`, `new()`...
 - ▶ “이 메모리 조각을 없애라!” - `free()`



참고:

- ▶ Stack 지역변수, 전역변수: 변수 선언이 어디에 위치했는지에 따라서 생성/제거 시기가 암묵적 자동 결정
- ▶ Heap 동적변수: 개발자 마음대로, 프로그램 실행하는 도중에 명시적으로 만들었다가, 쓰고 나면 없앴

Java의 예

```
A a = new A();
```

// 지역변수 a는 stack에 존재, OID를 가짐

new A()를 실행하면 heap에 동적 변수가 생성됨

- ▶ 이 동적변수의 타입은 클래스는 A
- ▶ 이 동적변수는 OID 를 가짐
- ▶ new는 A의 생성자를 실행한 후 OID를 리턴

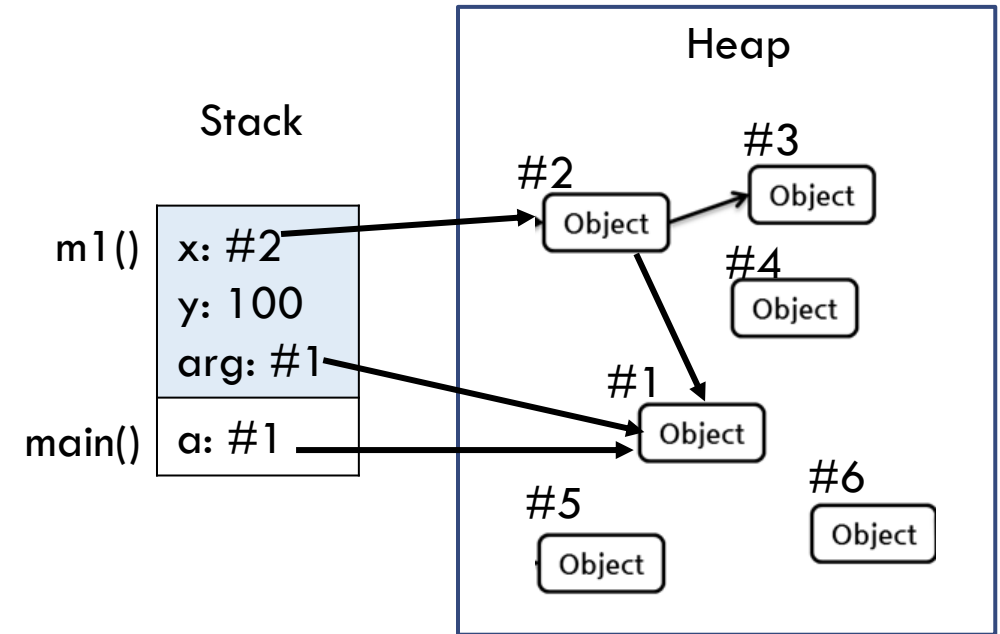
동적변수의 삭제는?

- ▶ garbage collection!
(참고 mark-and-sweep 알고리즘)

```
this.m1(a);
```

함수 인자 전달은 call-by-value

- ▶ main의 a의 값(동적변수의 OID #1)이 m1의 arg에 copy됨



```
void m1(A arg) {
    B x = new B(arg);
    int y = 100;
    ...
}

public static void main(String[] args){
    A a = new A();
    this.m1(a);
    ...
}
```

Quiz

- ▶ Java에서, 클래스 A는 필드의 개수가 10개이고 객체의 크기는 40바이트라고 알려져있고, 클래스 B는 필드의 개수가 1개이고 객체의 크기는 4바이트라고 알려져 있다. 아래 지역 변수 x와 y 의 정의에 대한 설명 중 거리가 가장 먼 것을 고르시오.

```
A x = new A();
```

```
B y = new B();
```

- (1) 변수 x는 함수가 리턴되면 사라진다
- (2) new A()로 생성된 동적변수는 함수가 리턴되어도 사라지지 않는다
- (3) new B()로 생성된 동적 변수보다 new A()로 생성된 동적변수의 크기가 크다
- (4) 변수 x와 y는 크기가 같다
- (5) new A()는 stack에, x는 heap에 위치한다