

자바프로그램의 구조

2018-02

고급객체지향프로그래밍

contents

- 자바 프로그램의 개발과 구동
- `main()` 메서드
- 변수와 메모리
- 멀티 스레드/멀티 프로세스

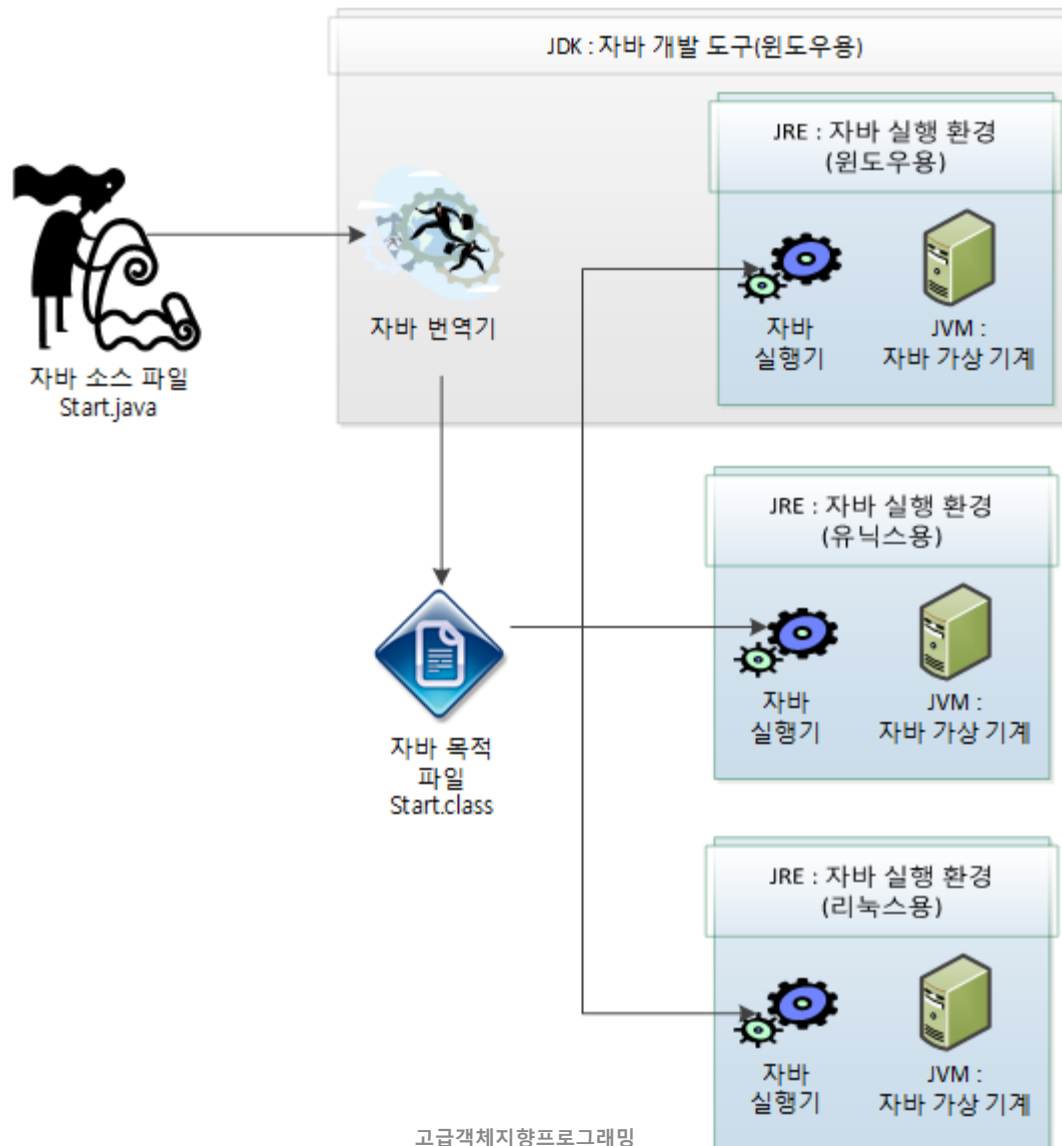
자바 프로그램의 개발과 구동

- 현실 세계와 자바 가상 세계

현실세계	자바 가상 세계
소프트웨어 개발 도구	JDK – 자바 개발 도구 (JVM용 소프트웨어 개발 도구)
운영체제	JRE – 자바 실행 환경 (JVM용 OS)
하드웨어(물리적 컴퓨터)	JVM – 자바 가상 기계 (가상의 컴퓨터)

- JDK: Java Development Kit
- JRE: Java Runtime Environment
- JVM: Java Virtual Machine

자바 프로그램의 개발과 구동



자바 프로그램의 개발과 구동

- 프로그램의 메모리 사용 방식

코드 실행 영역	데이터 저장 영역
----------	-----------

- 객체지향 프로그램의 메모리 사용 방식

코드 실행 영역	static 영역	
	stack 영역	heap 영역

main() 메서드

- main() 메서드 실행 전 JVM에서 수행하는 전처리 작업

```
1 public class Start {  
2     public static void main(String[] args) {  
3         System.out.println("Hello");  
4     }  
5 }
```

1



java.lang 패키지 배치

2



클래스와 import 패키지 배치

3

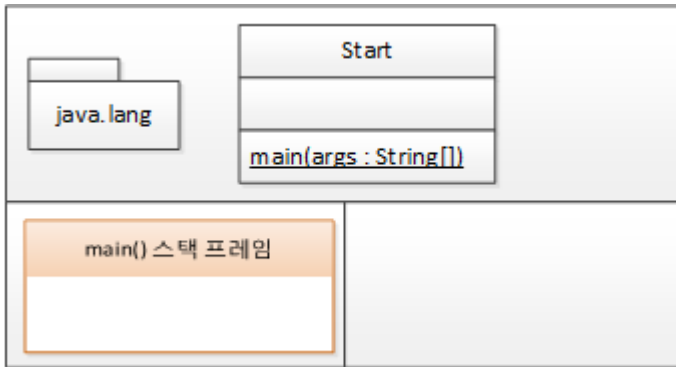
```

1 public class Start {
2     public static void main(String[] args) {
3         System.out.println("Hello");
4     }
5 }

```

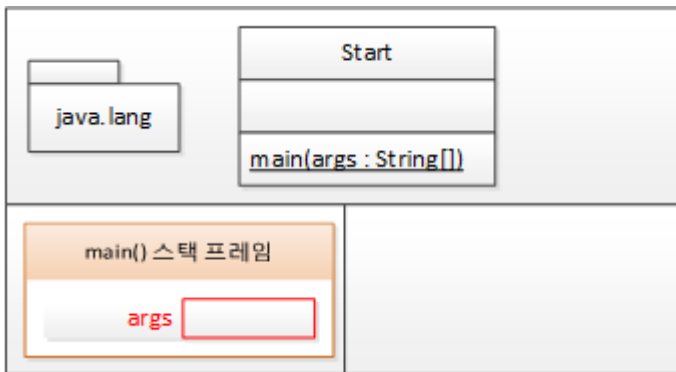
main() 메서드 스택 프레임 할당

4



메서드 인자 변수 공간 할당

5



println() 메서드 실행

6



main() 메서드 종료 후

스택 영역

main() 메서드 스택 프레임

실행위치(breakpoint)

```

1 public class Start {
2     public static void main(String[] args) {
3         System.out.println("Hello");
4     }
5 }
6
7
  
```

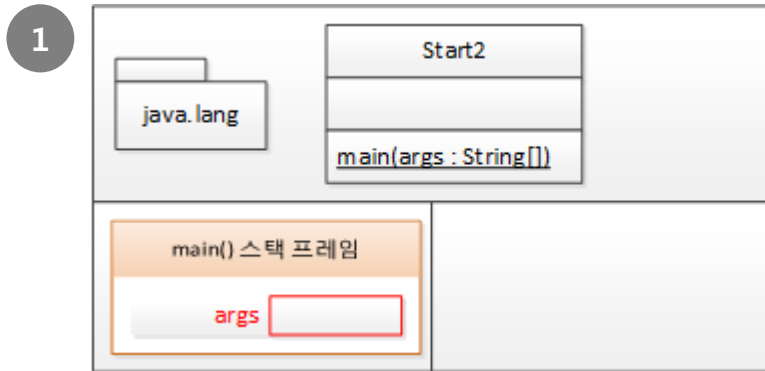
Name	Value
args	String[0] (id=16)

Start [Java Application] C:\Program Files\Java\jre1.8.0_51\bin\javaw.exe (2016. 3. 6. 오후 7:42:26)

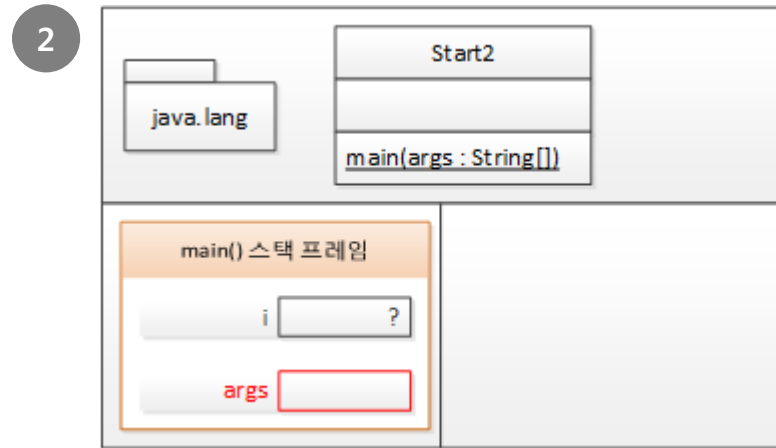
변수와 메모리

- 변수의 위치

```
1 public class Start2 {  
2     public static void main(String[] args) {  
3         int i;  
4         i = 10;  
5  
6         double d = 20.0;  
7     }  
8 }
```



line 2 실행 후



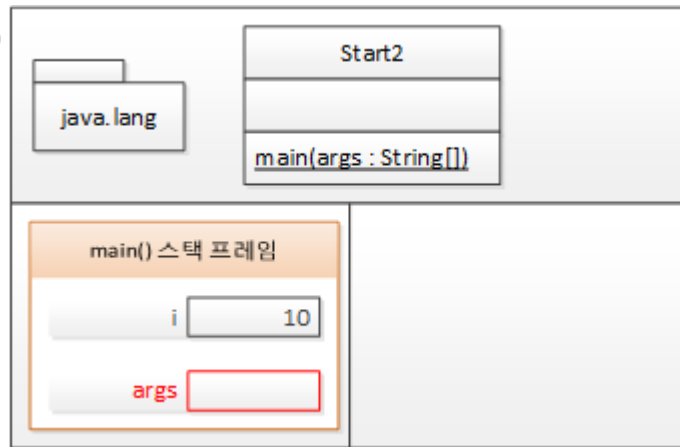
line 3 실행 후

```

1 public class Start2 {
2     public static void main(String[] args) {
3         int i;
4         i = 10;
5
6         double d = 20.0;
7     }
8 }

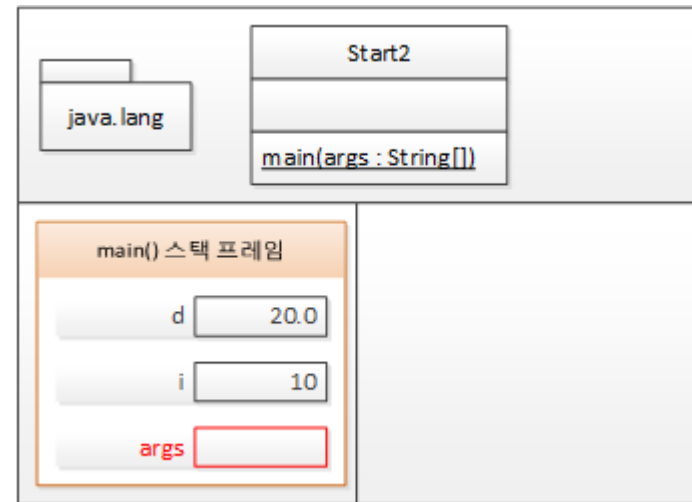
```

3



line 4 실행 후

4

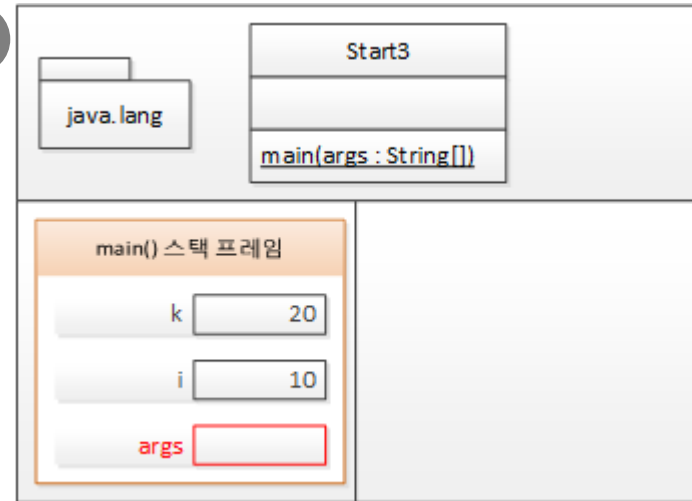


line 6 실행 후

블록 스택 프레임

```
1 public class Start3 {  
2     public static void main(String[] args) {  
3         int i = 10;  
4         int k = 20;  
5  
6         if(i == 10) {  
7             int m = k + 5;  
8             k = m;  
9         }else{  
10            int p = k + 10;  
11            k = p;  
12        }  
13  
14        //k = m + p;  
15    }  
16 }
```

1



line 5 실행 후

```

1 public class Start3 {
2     public static void main(String[] args) {
3         int i = 10;
4         int k = 20;
5
6         if(i == 10) {
7             int m = k + 5;
8             k = m;
9         }

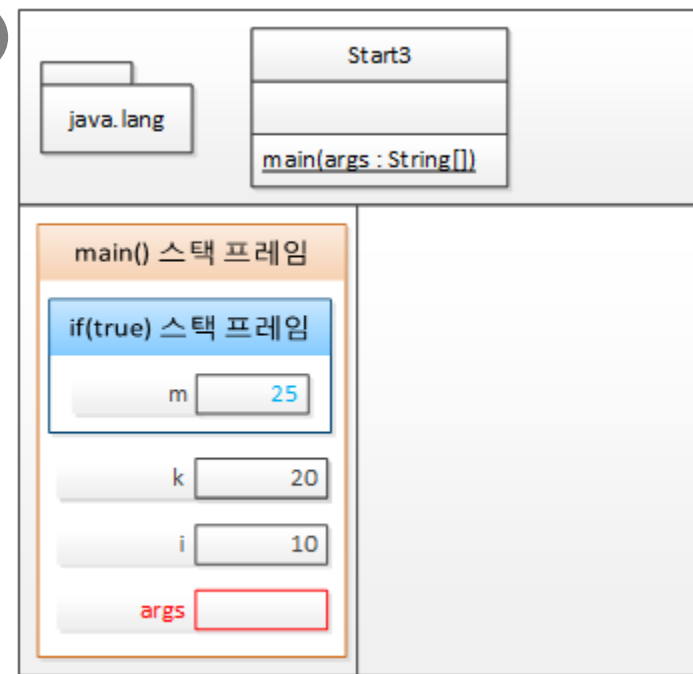
```

2



line 6 실행 후

3



line 7 실행 후

```

int m;
m = k + 5;

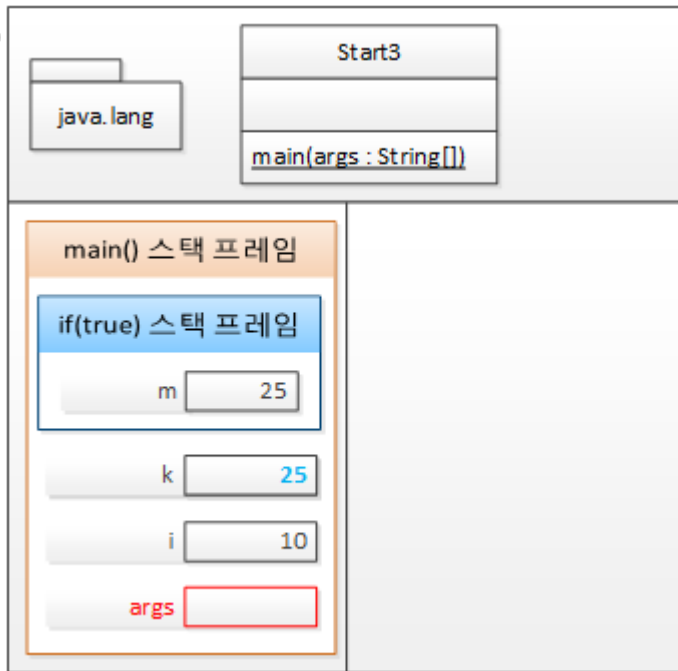
```

```

4      int k = 20;
5
6      if(i == 10) {
7          int m = k + 5;
8          k = m;
9      }else{
10         int p = k + 10;
11         k = p;
12     }
13
14     //k = m + p;
15

```

4



line 8 실행 후

5

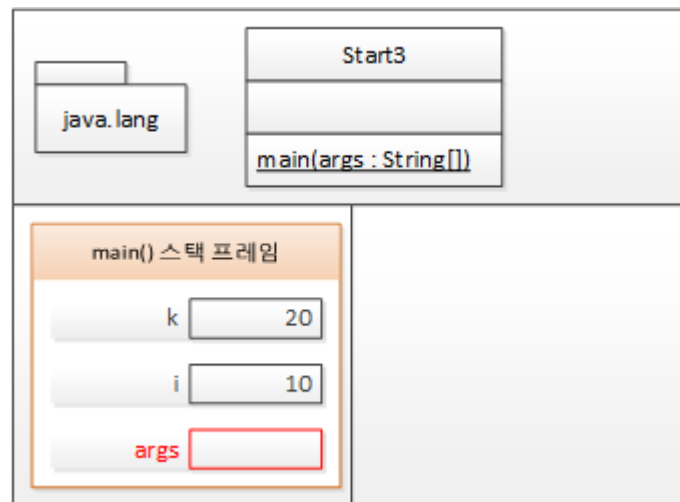


line 9 실행 후

지역 변수와 메모리

```
1 public class Start3 {  
2     public static void main(String[] args) {  
3         int i = 10;  
4         int k = 20;  
5  
6         if(i == 10) {  
7             int m = k + 5;  
8             k = m;  
9         }else{  
10            int p = k + 10;  
11            k = p;  
12        }  
13  
14        //k = m + p;  
15    }  
16 }
```

line 5: `System.out.println(m);`



```

1 public class Start3 {
2     public static void main(String[] args) {
3         int i = 10;
4         int k = 20;
5
6         if(i == 10) {
7             int m = k + 5;
8             k = m;
9         }else{
10             int p = k + 10;
11             k = p;
12         }
13         //k = m + p;
14     }
15 }
16 }

```

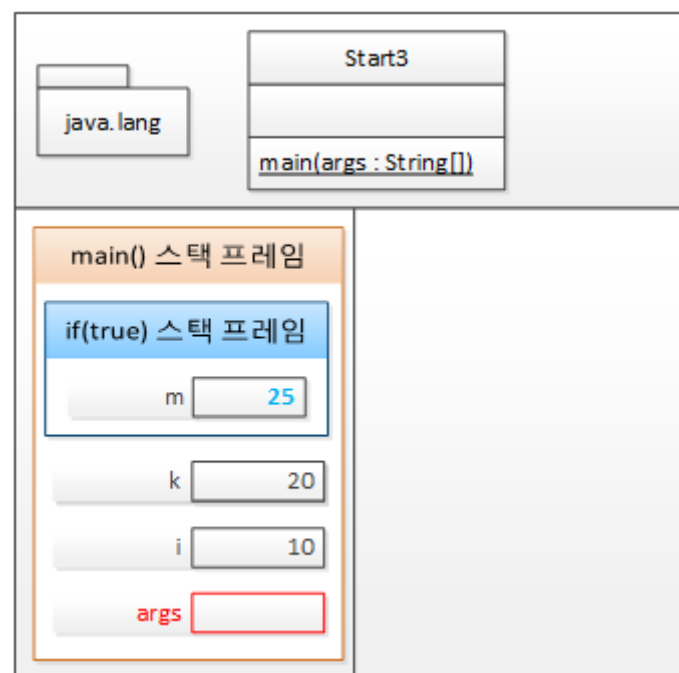
line 13: `System.out.println(m);`



```

1 public class Start3 {
2     public static void main(String[] args) {
3         int i = 10;
4         int k = 20;
5
6         if(i == 10) {
7             int m = k + 5;
8             k = m;
9         }else{
10             int p = k + 10;
11             k = p;
12         }
13
14         //k = m + p;
15     }
16 }

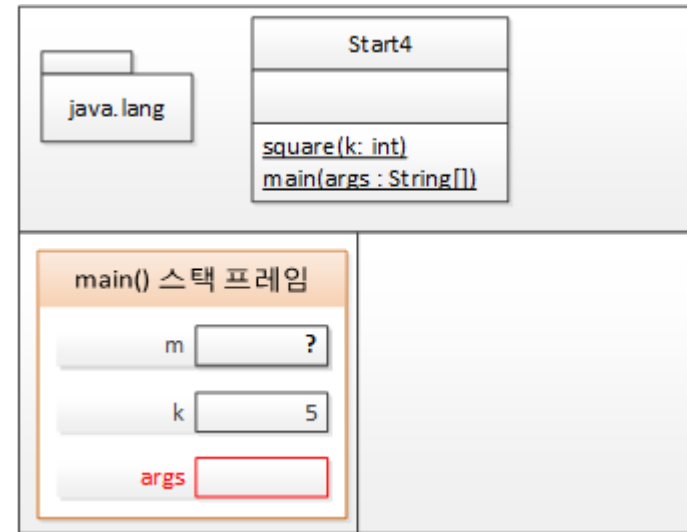
```



line 7 수행 후

메서드 호출과 메모리

```
1 public class Start4 {  
2     public static void main(String[] args) {  
3         int k = 5;  
4         int m;  
5  
6         m = square(k);  
7     }  
8  
9     private static int square(int k){  
10        int result;  
11  
12        k = 25;  
13  
14        result = k;  
15  
16        return result;  
17    }  
18 }
```



line 5 수행 후

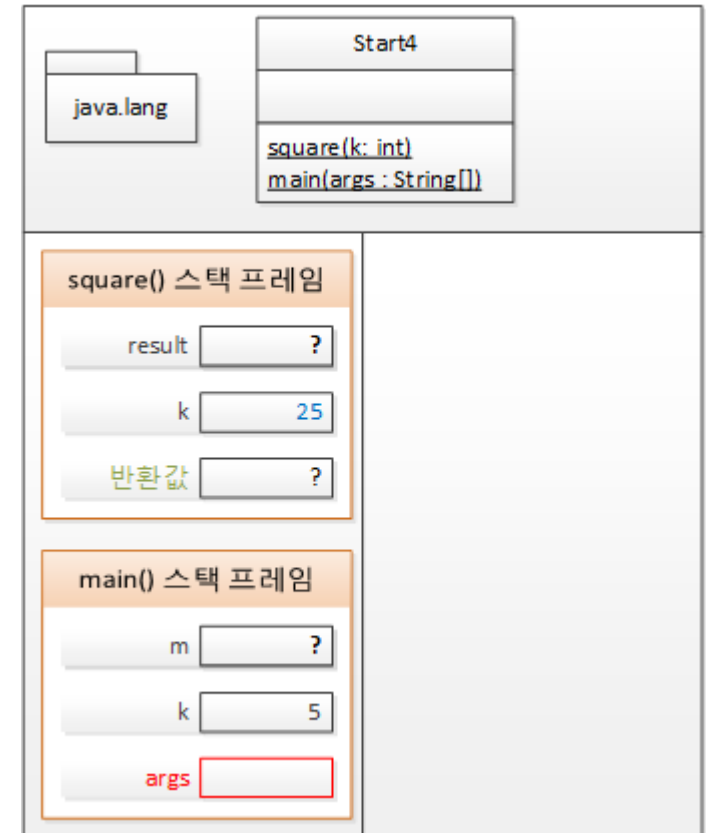
```

1 public class Start4 {
2     public static void main(String[] args) {
3         int k = 5;
4         int m;
5
6         m = square(k);
7     }
8
9     private static int square(int k){
10        int result;
11
12        k = 25;
13    }

```



line 11 수행 후

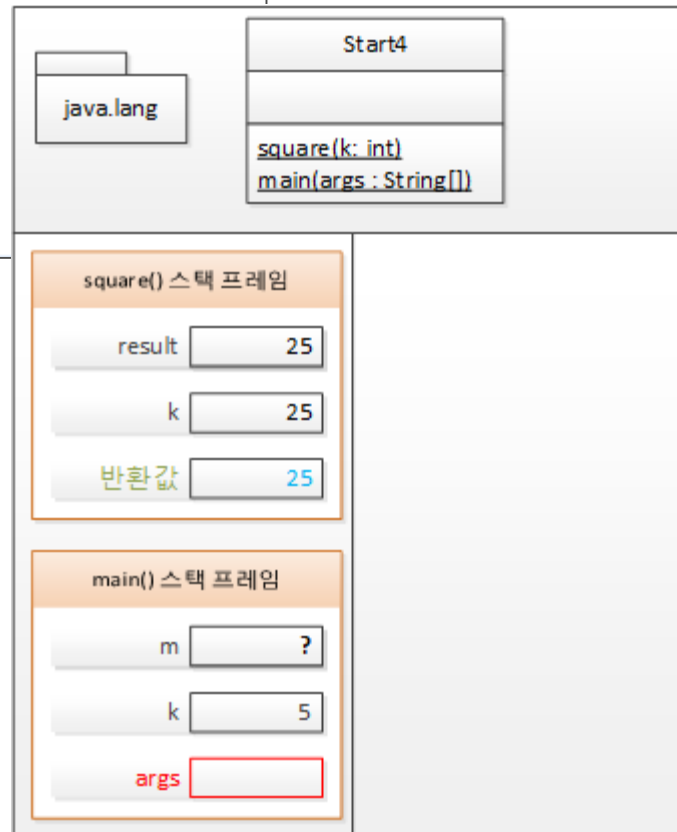


line 12 수행 후

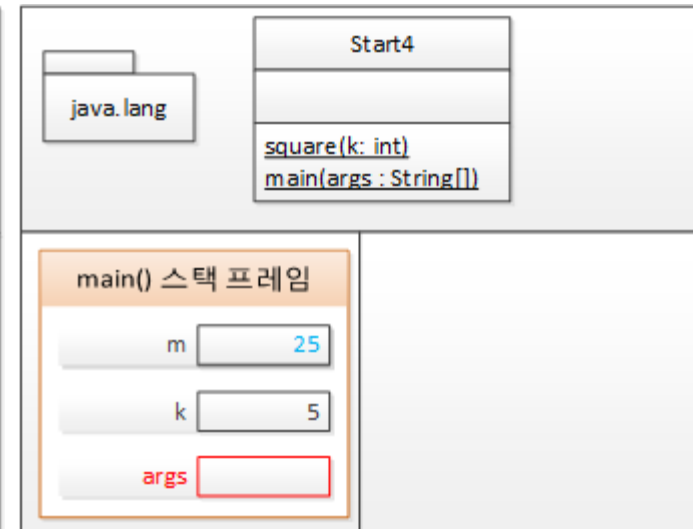
```

5
6     m = square(k);
7 }
8
9 private static int square(int k){
10     int result;
11
12     k = 25;
13
14     result = k;
15
16     return result;
17 }
18 }

```



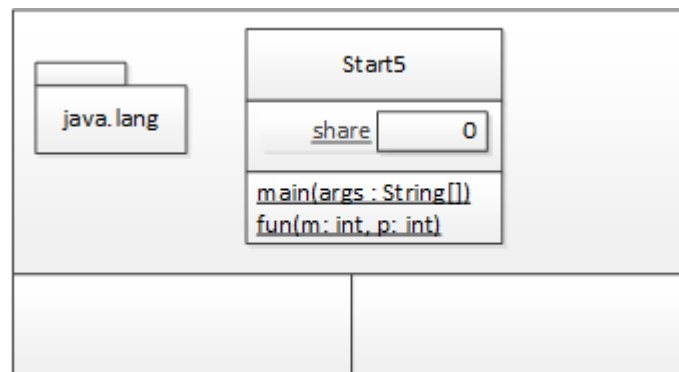
line 16 수행 후



line 6 수행 후

전역 변수와 메모리

```
1 public class Start5 {  
2     static int share;  
3  
4     public static void main(String[] args) {  
5         share = 55;  
6  
7         int k = fun(5, 7);  
8  
9         System.out.println(share);  
10    }  
11  
12    private static int fun(int m, int p){  
13        share = m + p;  
14  
15        return m - p;  
16    }  
17 }
```



line 4 수행 전

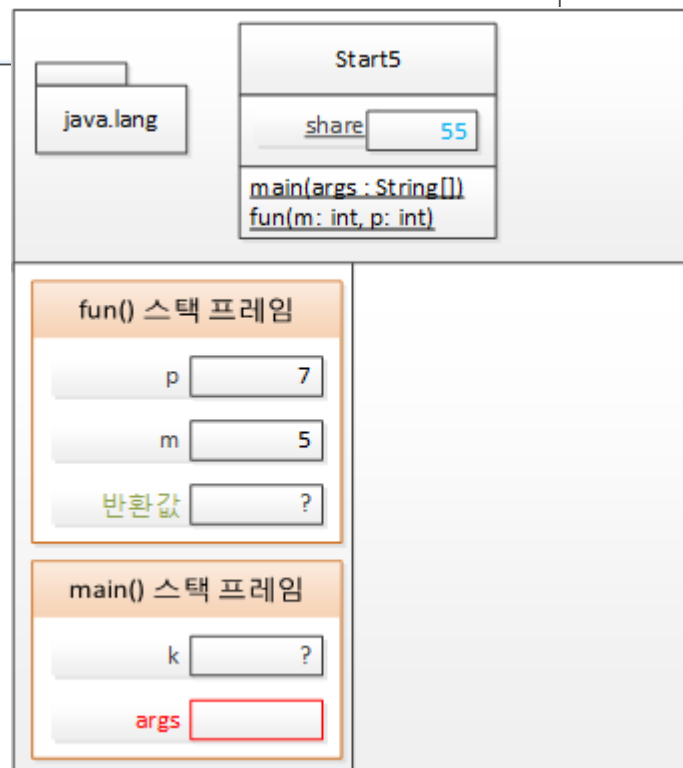


line 5 수행 후

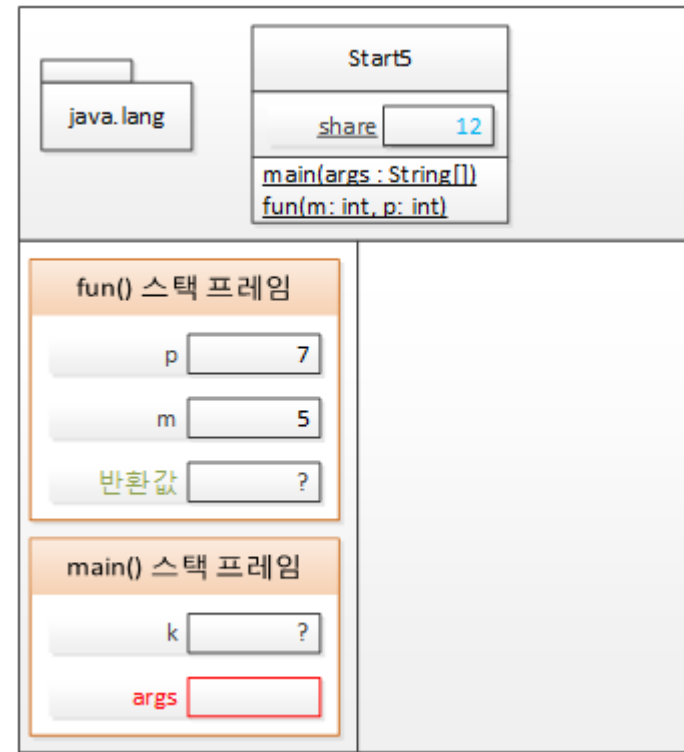
```

4 public static void main(String[] args) {
5     share = 55;
6
7     int k = fun(5, 7);
8
9     System.out.println(share);
10 }
11
12 private static int fun(int m, int p){
13     share = m + p;
14
15     return m - p;
16 }
17 }

```



line 12 수행 후



line 13 수행 후

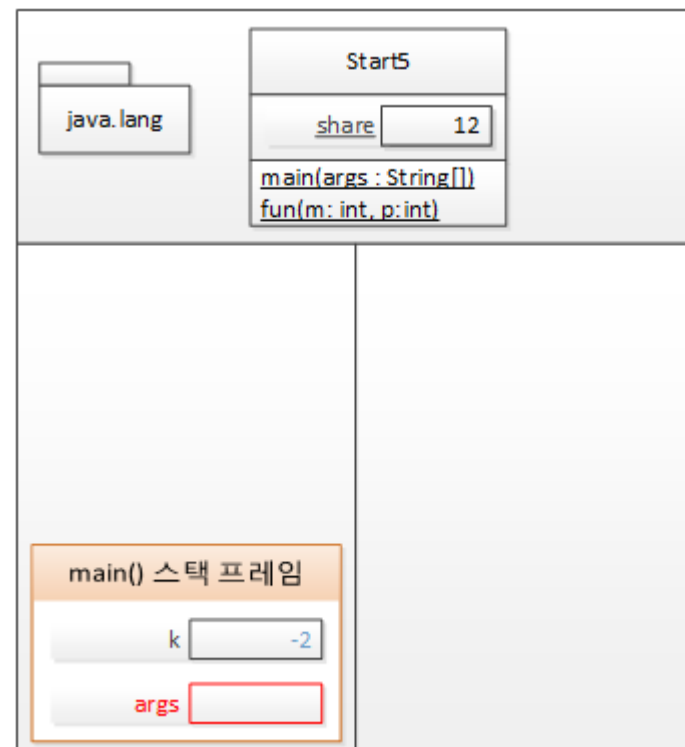
```

4 public static void main(String[] args) {
5     share = 55;
6
7     int k = fun(5, 7);
8
9     System.out.println(share);
10 }
11
12 private static int fun(int m, int p){
13     share = m + p;
14
15     return m - p;
16 }
17 }

```



line 15 수행 후



line 7 수행 후

Debug - chap02/src/Start5.java - Spring Tool Suite

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access Spring Java Debug

Debug Servers

- Start5 [Java Application]
 - Start5 at localhost:50532
 - Thread [main] (Suspended)
 - Start5.fun(int, int) line: 15
 - Start5.main(String[]) line: 7

C:\Program Files\Java\jre1.8.0_51\bin\javaw.exe (2016. 3. 6. 오후 10:35:00)

(x)= Variables Breakpoints Expressions

Name	Value
x+y =? "Start5.share"	12
x+y =? "share"	12
+ Add new expression	

Start.java Start2.java Start3.java Start4.java Start5.java

```
11
12 private static int fun(int m, int p){
13     share = m + p;
14
15     return m - p;
16 }
17
18
19
```

Outline

- Start5
 - share : int
 - main(String[]) : void
 - fun(int, int) : int

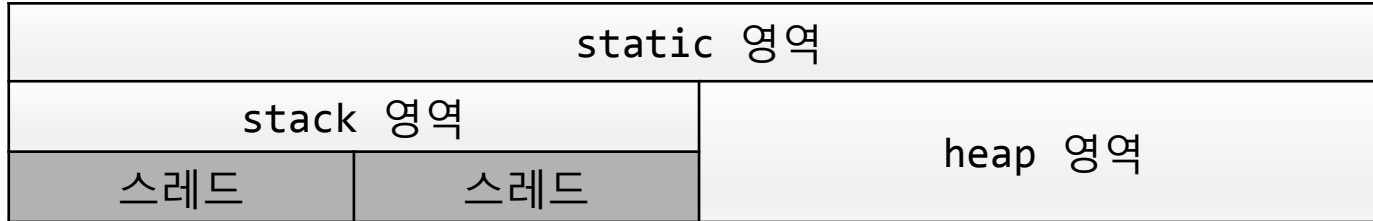
Console Tasks

Start5 [Java Application] C:\Program Files\Java\jre1.8.0_51\bin\javaw.exe (2016. 3. 6. 오후 10:35:00)

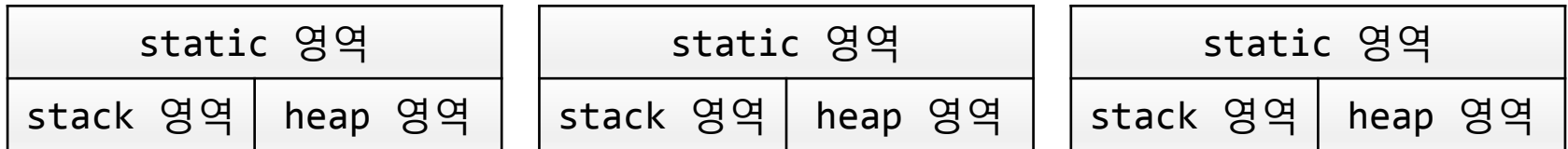
고급객체지향프로그래밍

멀티 스레드 / 멀티 프로세스

- 멀티 스레드: 스택 영역을 스레드 개수만큼 분할해서 사용



- 멀티 프로세스: 다수의 데이터 저장 영역을 갖는 구조



멀티 스레드 / 멀티 프로세스

- 멀티 스레드의 전역변수 문제

스레드 1	스레드 2	
전역 변수 A에 10 할당		전역 변수 A는 10을 저장
	전역 변수 A에 20 할당	전역 변수 A는 20을 저장
전역 변수 A의 값을 출력		20이 출력된다.

```
1 public class Start6 extends Thread {
2     static int share;
3
4     public static void main(String[] args) {
5         Start6 t1 = new Start6();
6         Start6 t2 = new Start6();
7
8         t1.start();
9         t2.start();
10    }
11
12    public void run(){
13        for(int count = 0; count < 10; count++){
14            System.out.println(share++);
15
16            try{sleep(20000);}
17            catch(InterruptedException e){ }
18        }
19    }
20 }
```