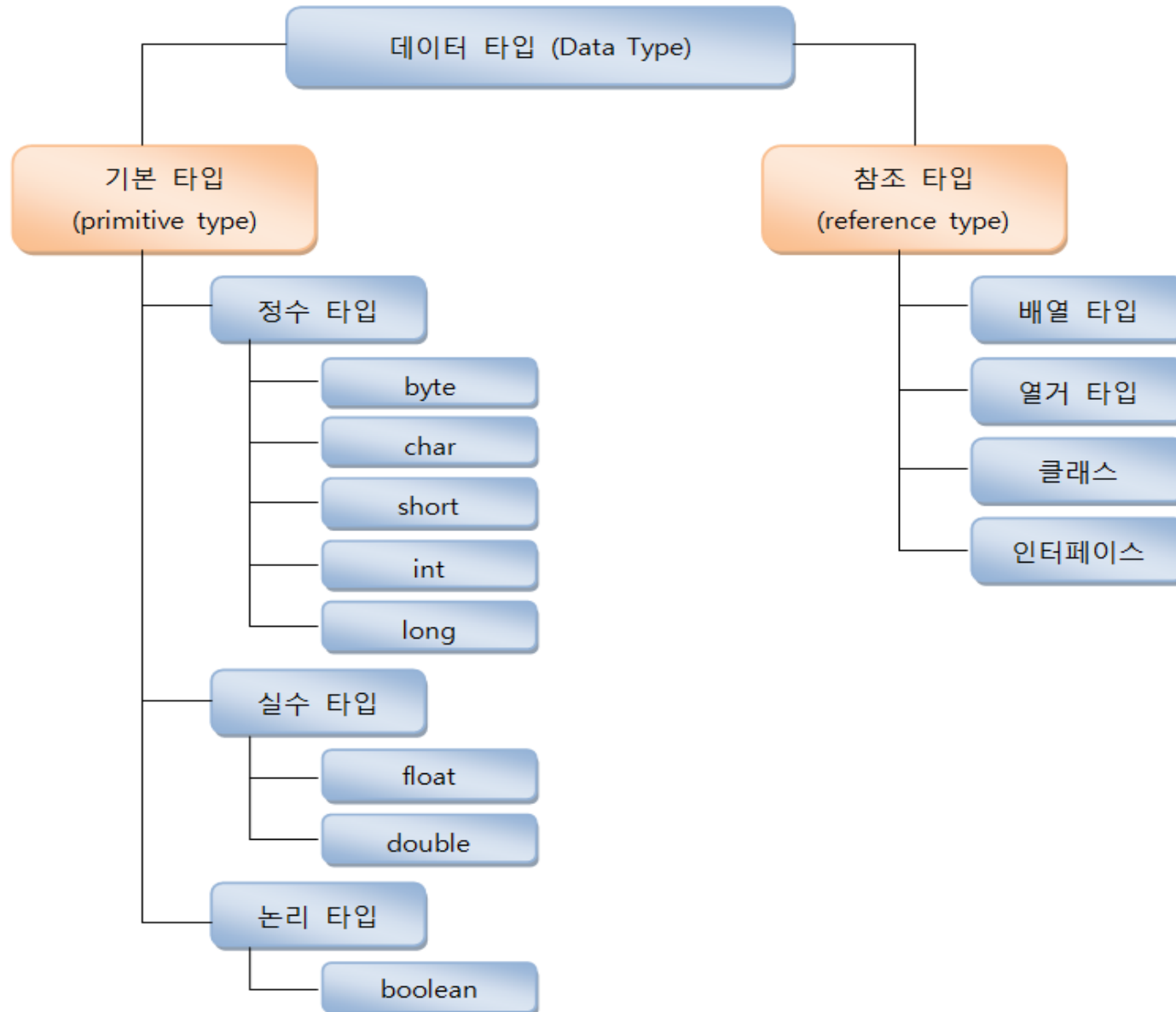


데이터 타입 분류

데이터 타입 분류



데이터 타입 분류

❖ 변수의 메모리 사용

- 기본 타입 변수 - 실제 값을 변수 안에 저장
- 참조 타입 변수 - 주소를 통해 객체 참조

[기본 타입 변수]

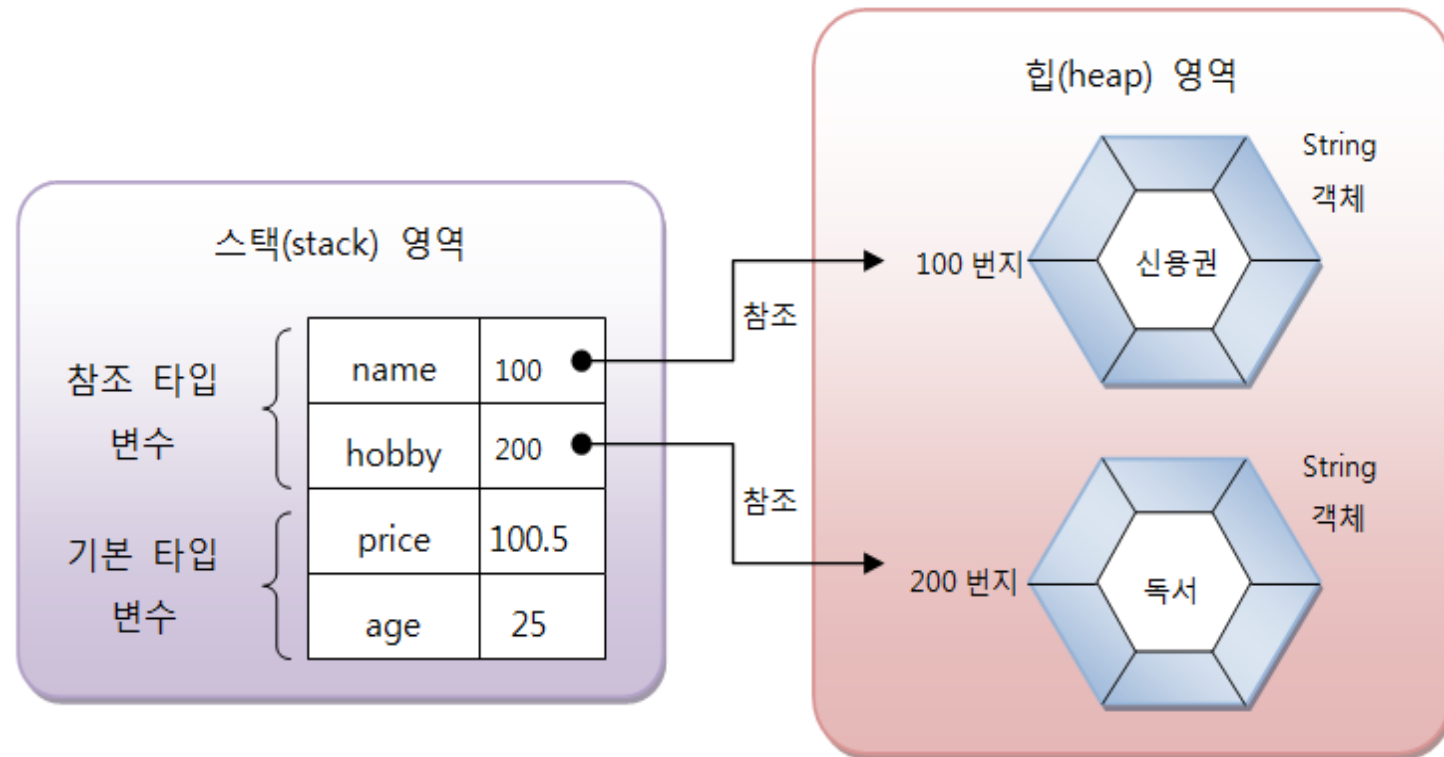
```
int age = 25;
```

```
double price = 100.5;
```

[참조 타입 변수]

```
String name = "신용권"
```

```
String hobby = "독서";
```

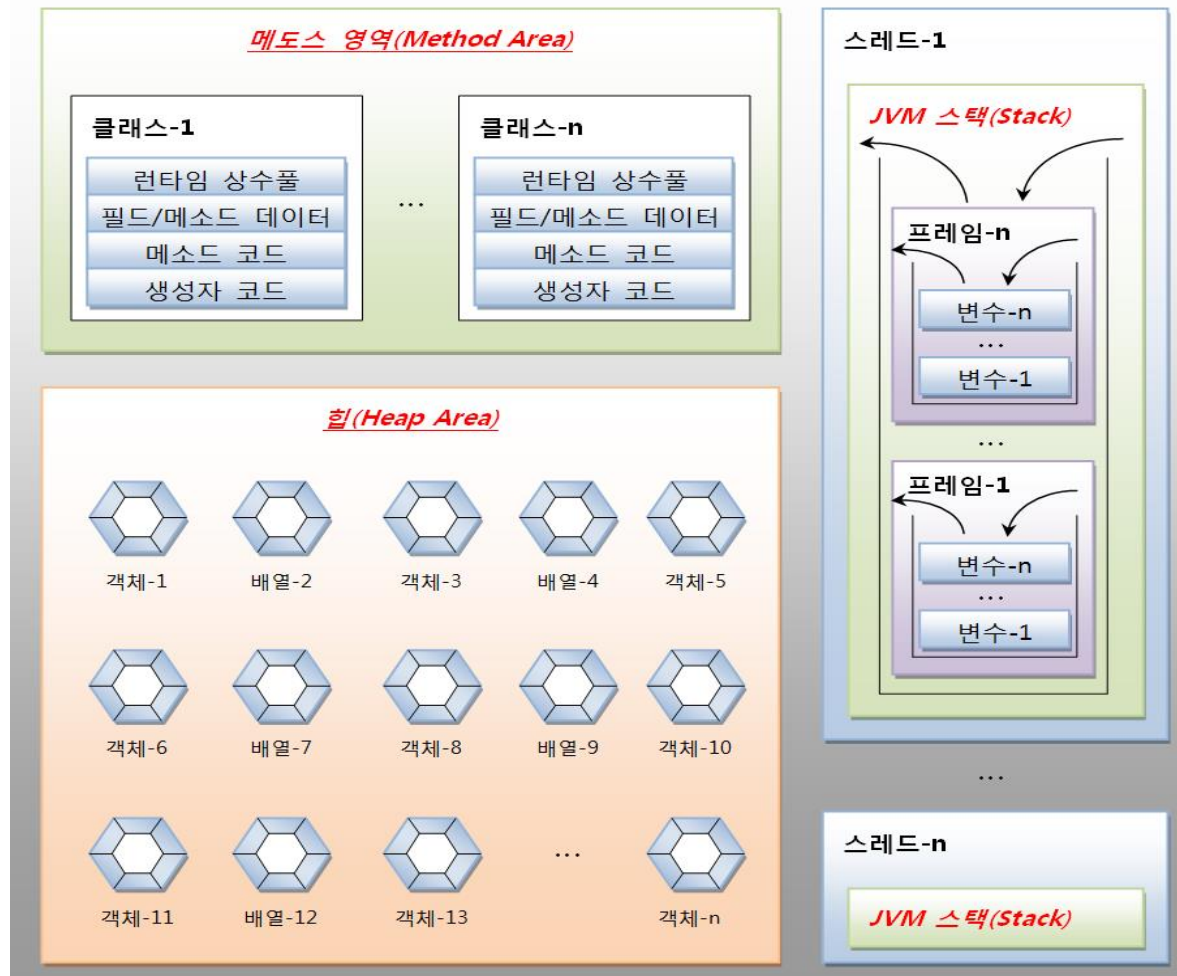


데이터 타입 분류

❖ JVM이 사용하는 메모리 영역

- OS에서 할당 받은 메모리 영역(Runtime Data Area)을 세 영역으로 구분

Runtime Data Area



데이터 타입 분류

❖ JVM이 사용하는 메모리 영역

- 메소드 영역
 - JVM 시작할 때 생성
 - 로딩된 클래스 바이트 코드 내용을 분석 후 저장
- 모든 스레드가 공유

❖ 힙 영역

- JVM 시작할 때 생성
- 객체/배열 저장
- 사용되지 않는 객체는 Garbage Collector 가 자동 제거

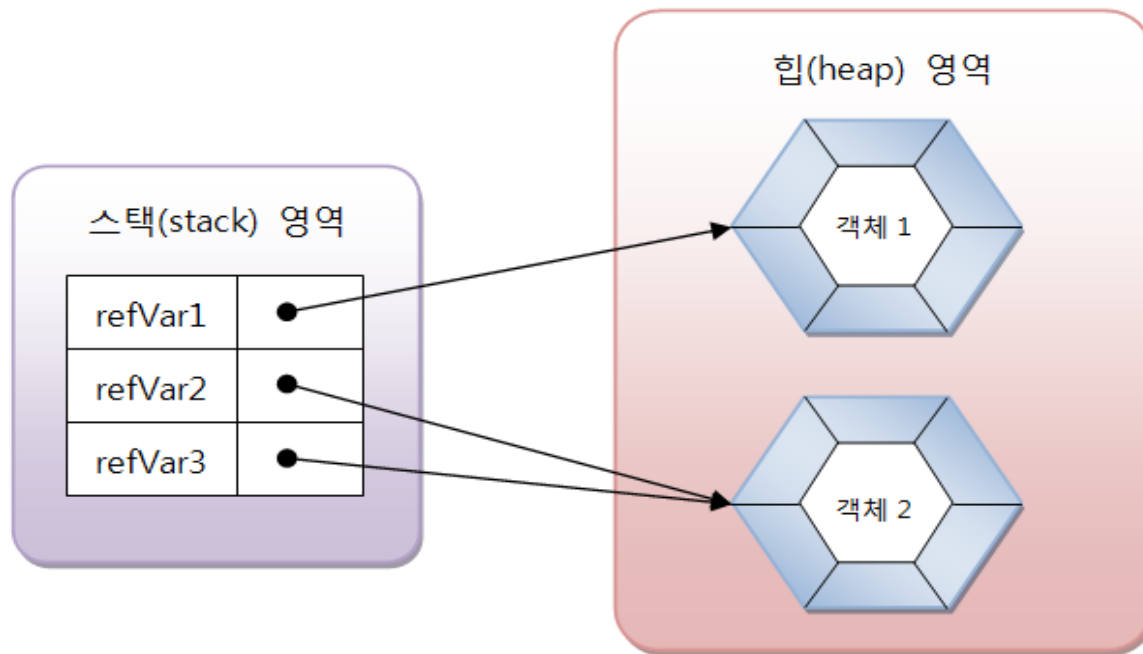
❖ JVM 스택

- 스레드 별 생성
- 메소드 호출할 때마다 Frame을 스택에 추가(push)
- 메소드 종료하면 Frame 제거(pop)

참조 변수의 ==, != 연산

❖ 변수의 값이 같은지 다른지 비교

- 기본 타입: byte, char, short, int, long, float, double, boolean
 - 의미 : 변수의 값이 같은지 다른지 조사
- 참조 타입: 배열, 열거, 클래스, 인터페이스
 - 의미 : 동일한 객체를 참조하는지 다른 객체를 참조하는지 조사



refVar1 == refVar2	결과: false
refVar1 != refVar2	결과: true

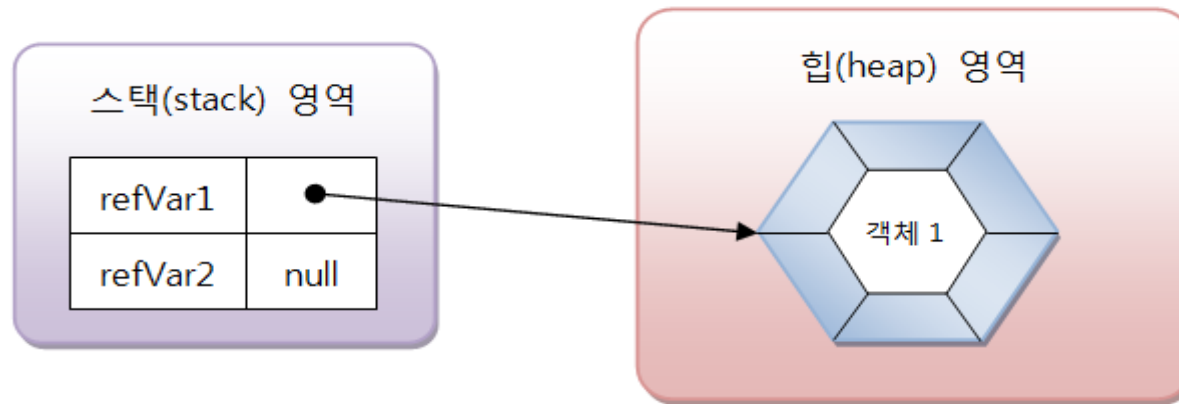
refVar2 == refVar3	결과: true
refVar2 != refVar3	결과: false

```
if( refVar2 == refVar3 ) { ... }
```

null과 NullPointerException

❖ null(널)

- 변수가 참조하는 객체가 없을 경우 초기값으로 사용 가능
- 참조 타입의 변수에만 저장가능
- null로 초기화된 참조 변수는 스택 영역 생성



그림에서 refVar1 은 힙 영역의 객체를 참조하므로 연산의 결과는 다음과 같다.

- ==, != 연산 가능

refVar1 == null	결과: false
refVar1 != null	결과: true

refVar2 는 null 값을 가지므로 연산의 결과는 다음과 같다.

refVar2 == null	결과: true
refVar2 != null	결과: false

null과 NullPointerException

❖ NullPointerException의 의미

- 예외(Exception)
 - 사용자의 잘못된 조작 이나 잘못된 코딩으로 인해 발생하는 프로그램 오류
- NullPointerException
 - 참조 변수가 null 값을 가지고 있을 때
 - 객체의 필드나 메소드를 사용하려고 했을 때 발생

```
int[] intArray = null;  
intArray[0] = 10;      //NullPointerException
```

```
String str = null;  
System.out.println("총 문자수: " + str.length()); //NullPointerException
```