

프로그래밍 공부의 기초!



Java

자바(Java)

반복문

자바(Java)

반복문

무한 반복

```
while (true) {  
    실행 코드  
}
```

자바(Java)

반복문

break : 반복의 종료

```
while (조건식) {  
    실행 코드  
    break;  
}
```

자바(Java)

반복문

continue : 반복의 처음으로 이동

```
while (조건식) {  
    실행 코드  
    continue;  
}
```

자바(Java)

예제9)

**1 - 10까지 정수 입력 시 반복 종료
그 외 정수 입력 시 다시 입력 받게 하시오.**

자바(Java)

반복문

do - while 반복문 기본 구조

```
do {  
    실행 코드  
} while (조건식);
```

자바(Java)

예제10)

do - while문 사용해보기

자바(Java)

참조 타입

자바(Java)

참조 타입

JVM Memory 구조

Method

클래스에서 사용되는 모든 코드 값을 저장

Stack

메서드의 정보를 프레임 형태로 저장

새로운 메서드가 실행 될 때마다, 새로운 프레임이 Stack Memory에 저장
프레임 내부에는 지역변수 스택이 존재하며, 지역변수 관리 작업을 수행

Heap

인스턴스 변수가 저장되는 영역 (문자열, 배열, 클래스, 인터페이스 등)
new 키워드를 이용하여 생성되는 값은 Heap 영역에 저장

JVM Memory

Method

stack

Heap

자바(Java)

참조 타입

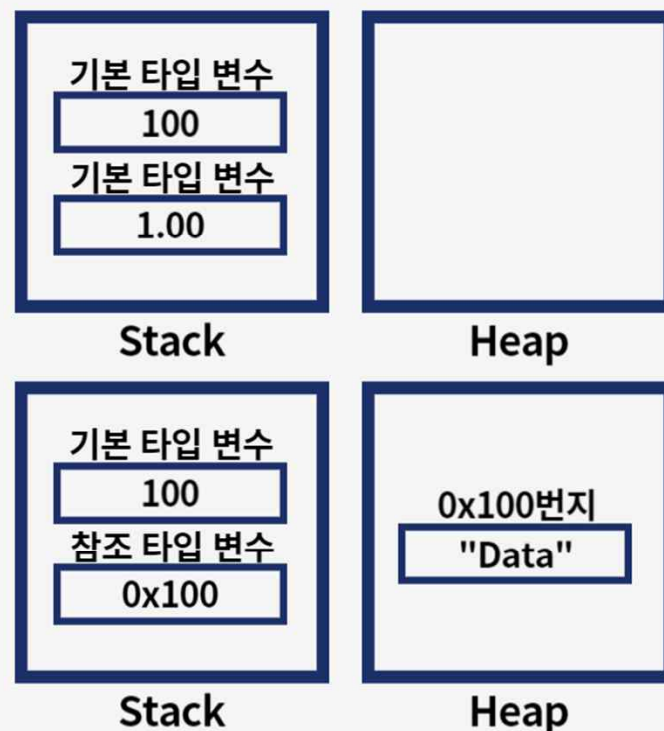
기본 타입 & 참조 타입

기본 타입

- 일반적인 정수, 실수, 단일문자, 논리 리터럴을 저장하는 타입
- Stack Memory 영역에 데이터를 저장하는 공간을 생성
- Stack Memory 영역에 실제 데이터를 저장

참조 타입

- 객체의 주소값을 저장하는 타입
(문자열, 배열, 클래스, 인터페이스)
- Heap Memory 영역에 데이터 저장공간을 생성 및 데이터를 저장
- Stack Memory 영역에는 Heap Memory의 주소 값을 저장



자바(Java)

참조 타입

String Class

String Class : 자바에서 문자열 데이터를 정의할 때 사용되는 클래스

java.lang.String 클래스에는 문자열과 관련 된 작업을 할 때 유용한 다양한 메소드를 포함

- 문자열 참조 변수의 저장 된 주소값 비교

참조 변수에 저장 된 주소값을 비교할 때에는 (“ == “ , “ != “)을 사용

동일 한 리터럴 문자열값을 저장하는 참조 변수를 선언 할 경우 자바에서는 동일한 객체를 공유하도록 되어있음
새로운 객체를 생성 할 때에는 반드시 “ new “ 연산자를 사용해야함 (new 연산자 : 객체 생성 연산자)

- 문자열 참조 변수가 가르키는 곳에 저장 된 데이터 값 비교

객체의 주소값이 아닌 실제 객체에 저장 된 데이터 값을 비교하기 위해서는 “ .equals() “ 메소드를 사용
원본 문자열과 .equals() 메소드에 주어진 매개값을 직접 비교하고, 같으면 true , 다르면 false를 반환

EX) : boolean flag = str1.equals(str2);

자바(Java)

참조 타입

String Class Method & Operator

메서드	설명
.length()	지정 된 문자열의 길이값을 반환
.charAt(index)	지정 된 Index에 단일 문자값을 반환
+ operator : 기존 문자열 데이터에 새로운 문자값을 이어 붙임	

자바(Java)

참조 타입

String Class Method

메서드	설명
.toLowerCase()	지정 된 문자열을 소문자로 반환
.toUpperCase()	지정 된 문자열을 대문자로 반환
.replace(“old” , “new)	지정 된 문자를 새로운 문자로 치환
.split(regex)	지정 된 값을 기준으로 문자열을 분리
.split()	배열을 이용하여 분리 된 문자열을 따로 저장
.trim()	지정 된 문자열의 앞뒤 공백을 제거

See you
Again!

→ 다음 이 시간에 만나요!