

Lab 9

Q&A

<https://padlet.com/dcslabcp/0510-roji14mirapdhlco>

Java doc

<https://docs.oracle.com/en/java/javase/20/docs/api/index.html>

[OVERVIEW](#) [MODULE](#) [PACKAGE](#) [CLASS](#) [USE](#) [TREE](#) [PREVIEW](#) [NEW](#) [DEPRECATED](#) [INDEX](#) [HELP](#)

Java® Platform, Standard Edition & Java Development Kit Version 20 API Specification

This document is divided into two sections:

- Java SE**
The Java Platform, Standard Edition (Java SE) APIs define the core Java platform for general-purpose computing with Java.
- JDK**
The Java Development Kit (JDK) APIs are specific to the JDK and will not necessarily be available in all implementations in modules whose names start with `jdk`.

[All Modules](#) [Java SE](#) [JDK](#) [Other Modules](#)

Module	Description
<code>java.base</code>	Defines the foundational APIs of the Java SE Platform.
<code>java.compiler</code>	Defines the Language Model, Annotation Processing, and Java Compiler APIs.
<code>java.datatransfer</code>	Defines the API for transferring data between and within applications.
<code>java.desktop</code>	Defines the AWT and Swing user interface toolkits, plus APIs for accessibility.

Hello world in Java



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

- 파일명은 public 클래스명과 동일해야 함
- main 함수를 갖는 별도의 Main 클래스를 두기도 함

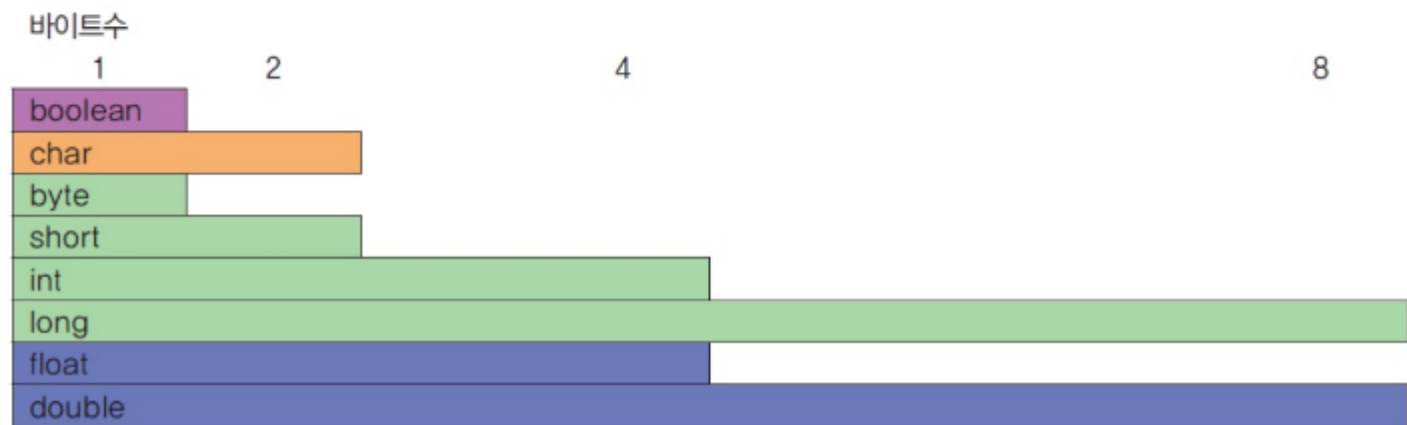
Java의 변수의 종류

- 기초 변수(primitive variable)에는 실제 데이터값이 저장된다.
- 참조 변수(reference variable)는 객체를 참조할 때 사용되는 변수로서 여기에는 객체의 참조값이 저장된다.



Java의 기본 자료 유형

자료형	설명	크기(바이트)	범위
byte	부호있는 정수	1바이트	-128에서 127
short	부호있는 정수	2바이트	-32768에서 32767
int	부호있는 정수	4바이트	-2147483648에서 2147483647(20억 정도)
long	부호있는 정수	8바이트	-9223372036854775808에서 9223372036854775807
float	실수형	4바이트	약 $\pm 3.40282347 \times 10^{+38}$ (유효숫자 6-7개 정도)
double	실수형	8바이트	약 $\pm 1.7976931 \times 10^{+308}$ (유효숫자 15개 정도)
char	문자형	2바이트	\u0000에서 \uFFFF
boolean	논리형	1바이트	NA



Java의 산술 연산자

연산자	기호	의미	예
덧셈	+	x와 y를 더한다	$x+y$
뺄셈	-	x에서 y를 뺀다.	$x-y$
곱셈	*	x와 y를 곱한다.	$x*y$
나눗셈	/	x를 y로 나눈다.	x/y
나머지	%	x를 y로 나눌 때의 나머지값	$x\%y$

Java의 증감 연산자

연산자	의미
<code>++x</code>	x값을 먼저 증가한 후에 다른 연산에 사용한다. 이 수식의 값은 증가된 x값이다.
<code>x++</code>	x값을 먼저 사용한 후에, 증가한다. 이 수식의 값은 증가되지 않은 원래의 x값이다.
<code>--x</code>	x값을 먼저 감소한 후에 다른 연산에 사용한다. 이 수식의 값은 감소된 x값이다.
<code>x--</code>	x값을 먼저 사용한 후에, 감소한다. 이 수식의 값은 감소되지 않은 원래의 x값이다.

Java의 관계 연산자

연산자 기호	의미	사용예
==	x와 y가 같은가?	x == y
!=	x와 y가 다른가?	x != y
>	x가 y보다 큰가?	x > y
<	x가 y보다 작은가?	x < y
>=	x가 y보다 크거나 같은가?	x >= y
<=	x가 y보다 작거나 같은가?	x <= y

Java의 논리 연산자

연산자 기호	사용예	의미
&&	x && y	AND 연산, x와 y가 모두 참이면 참, 그렇지 않으면 거짓
	x y	OR 연산, x나 y중에서 하나만 참이면 참, 모두 거짓이면 거짓
!	!x	NOT 연산, x가 참이면 거짓, x가 거짓이면 참

Java의 비트 연산자

연산자	설명
~	비트 NOT
&	비트 AND
^	비트 XOR
	비트 OR
<<	비트 왼쪽 이동
>>	비트 오른쪽 이동
>>>	비트 오른쪽 이동(unsigned)
<<<	비트 왼쪽 이동(unsigned)

Java의 연산자 우선순위

연산자	우선 순위	결합 규칙
배열, 멤버, 메소드 호출	[] . ()(메소드 호출)	왼쪽에서 오른쪽
단항	! ~ ++ -- + - ()(형변환) new	오른쪽에서 왼쪽
곱셈	* / %	왼쪽에서 오른쪽
덧셈	+ -	왼쪽에서 오른쪽
이동	<< >>> >>	왼쪽에서 오른쪽
관계	< > <= >= instanceof	왼쪽에서 오른쪽
동등	== !=	왼쪽에서 오른쪽
비트별 AND	&	왼쪽에서 오른쪽
비트별 XOR	^	왼쪽에서 오른쪽
비트별 OR		왼쪽에서 오른쪽
논리적 AND	&&	왼쪽에서 오른쪽
논리적 OR		왼쪽에서 오른쪽
조건	? :	오른쪽에서 왼쪽
대입	= += -= *= /= %= &= ^= = <<= >>= >>>=	오른쪽에서 왼쪽

Java의 입력 - Scanner

메소드	설명
<code>String next()</code>	다음 아이템을 문자열 타입으로 리턴한다.
<code>byte nextByte()</code>	다음 아이템을 <code>byte</code> 타입으로 리턴한다.
<code>short nextShort()</code>	다음 아이템을 <code>short</code> 타입으로 리턴한다.
<code>int nextInt()</code>	다음 아이템을 <code>int</code> 타입으로 리턴한다.
<code>long nextLong()</code>	다음 아이템을 <code>long</code> 타입으로 리턴한다.
<code>float nextFloat()</code>	다음 아이템을 <code>float</code> 타입으로 리턴한다.
<code>double nextDouble()</code>	다음 아이템을 <code>double</code> 타입으로 리턴한다.
<code>String nextLine()</code>	한 라인 전체('\n'까지)를 읽어 문자열 타입('\n' 미포함)으로 리턴한다.

연습 문제 1

```
import java.util.Scanner;

0개의 사용위치
public class Practice1 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("이름을 입력하세요 : ");
        String name = sc.next();

        System.out.print("나이를 입력하세요 : ");
        int age = sc.nextInt();

        System.out.println(name + "님은 " + age + "살 이시군요!");
    }
}
```

- 프로그램을 작성하고
본인의 이름과 나이를 입력해 봅시다.

```
이름을 입력하세요 : 홍길동
나이를 입력하세요 : 21
홍길동님은 21살 이시군요!
```

Java의 문자열

메소드	설명
<code>char charAt(int index)</code>	지정된 인덱스에 있는 문자를 반환한다.
<code>int compareTo(String anotherString)</code>	사전적 순서로 문자열을 비교한다. 앞에 있으면 -1, 같으면 0, 뒤에 있으면 1이 반환된다.
<code>String concat(String str)</code>	주어진 문자열을 현재의 문자열 뒤에 붙인다.
<code>boolean equals(Object anObject)</code>	주어진 객체와 현재의 문자열을 비교한다.
<code>boolean isEmpty()</code>	<code>length()</code> 가 0이면 <code>true</code> 를 반환한다.
<code>int length()</code>	현재 문자열의 길이를 반환한다.
<code>String toLowerCase()</code>	문자열의 문자들을 모두 소문자로 변경한다.
<code>String toUpperCase()</code>	문자열의 문자들을 모두 대문자로 변경한다.

연습 문제 2

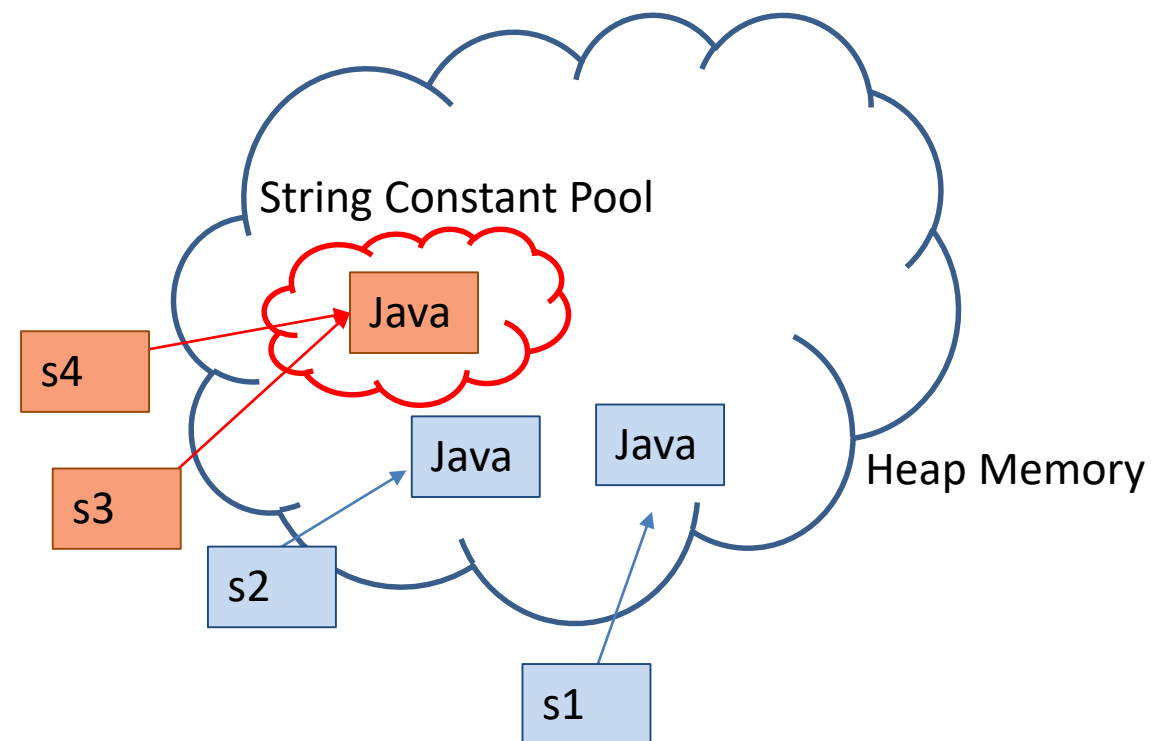
```
import java.util.Scanner;  
0개의 사용위치  
public class Practice2 {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("문자열을 입력해 주세요: ");  
        String input = sc.next();  
        System.out.print("찾을 위치를 입력해 주세요(1이상 문자열 길이 이하) : ");  
        int where = sc.nextInt();  
        System.out.println(input.charAt(where - 1));  
    }  
}
```

- 프로그램을 작성해 봅시다.

```
문자열을 입력해 주세요: abcdefg  
찾을 위치를 입력해 주세요(1이상 문자열 길이 이하) : 3  
c
```


Java의 문자열

```
String s1 = new String( original: "Java");  
String s2 = new String( original: "Java");  
String s3 = "Java";  
String s4 = "Java";
```



연습 문제 3

```
public class Practice3 {  
    public static void main(String[] args) {  
        String s1 = new String( original: "Java");  
        String s2 = new String( original: "Java");  
        String s3 = "Java";  
        String s4 = "Java";  
  
        System.out.println(s1 == s2);  
        System.out.println(s1 == s3);  
        System.out.println(s3 == s4);  
        System.out.println(s1.equals(s2));  
        System.out.println(s1.equals(s3));  
        System.out.println(s3.equals(s4));  
    }  
}
```

- 출력 결과는 어떻게 될까요?

```
false  
false  
true  
true  
true  
true
```

Java의 제어 구조 – switch문

```
public class SwitchTest {  
    0개의 사용위치  
    public static void main(String[] args) {  
        String greeting = "hello";  
  
        switch(greeting){  
            case "hello" :  
                System.out.print("만났을 때 건네는 인사말");  
                break;  
            case "goodbye" :  
                System.out.println("헤어질 때 건네는 인사말");  
                break;  
            default :  
                System.out.println("처리할 수 없습니다.");  
                break;  
        }  
    }  
}
```

- C++과는 달리 switch문에 문자열이 가능함

Java의 배열

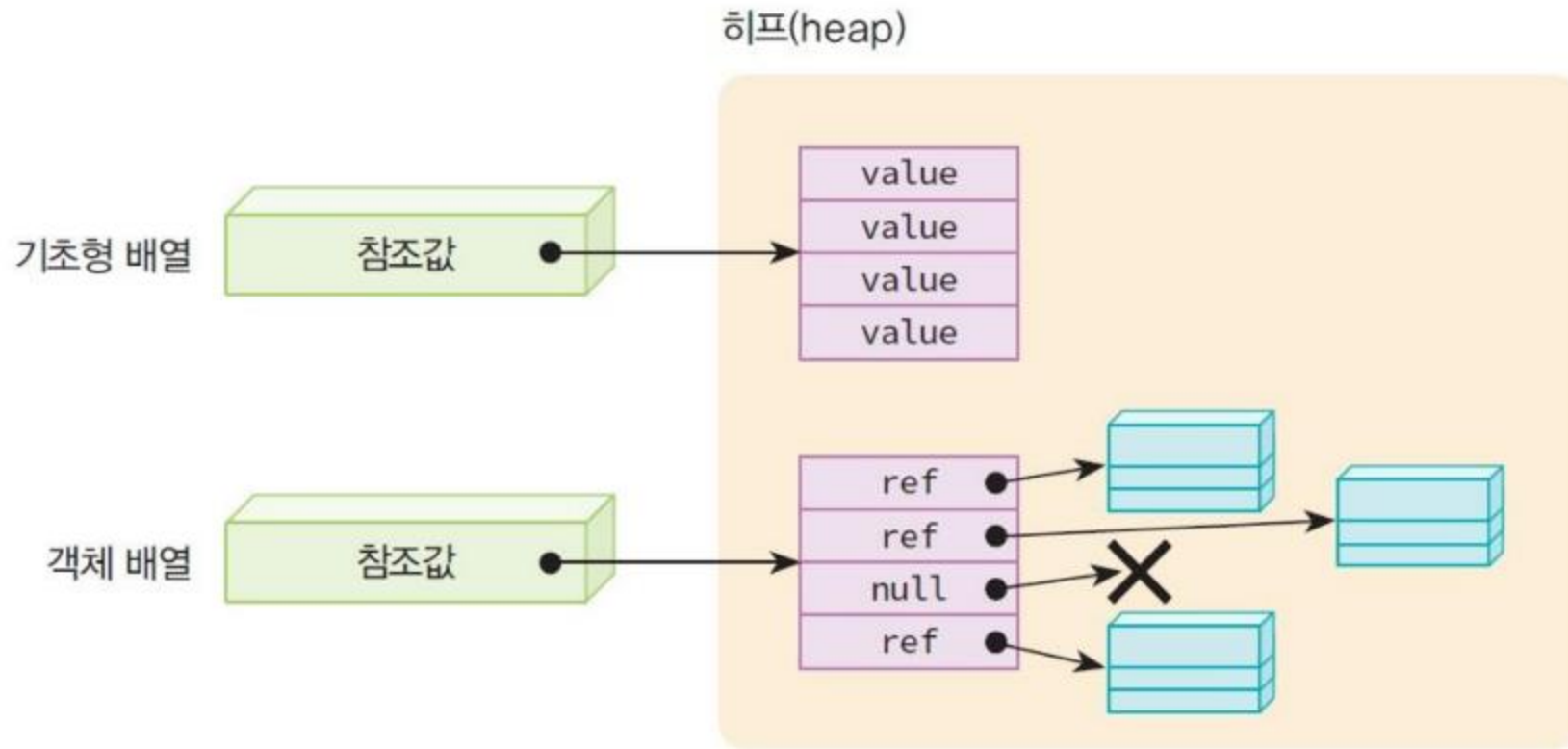
- 다양한 배열 선언 방법

```
int[] arr1;  
arr1 = new int[10];
```

```
double[] arr2 = new double[10];
```

```
String[] colors = {"red", "blue", "pink", "black", "green"};
```

Java의 객체 배열



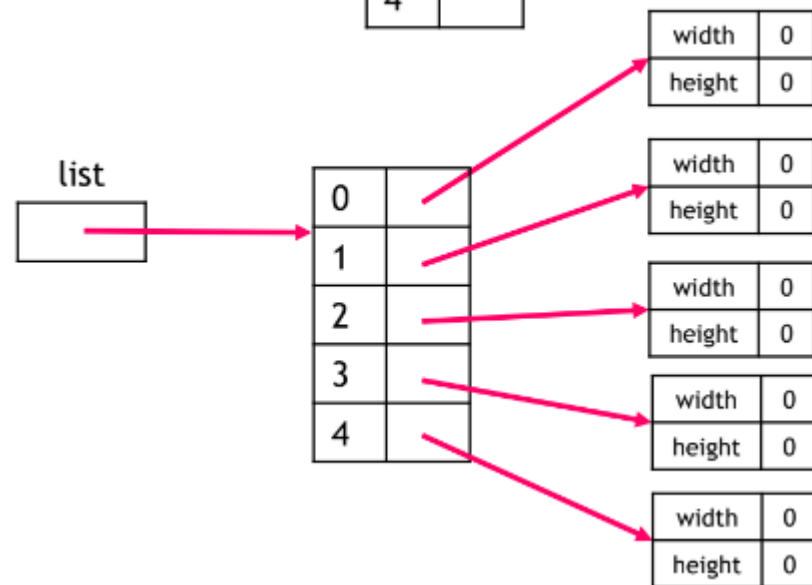
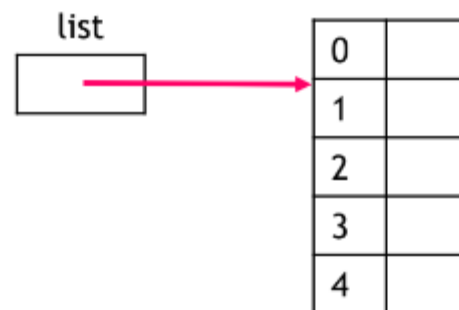
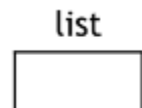
Java의 객체 배열

```
public class ObjectListTest {  
    0개의 사용위치  
    public static void main(String[] args) {  
        Rect[] list;  
        list = new Rect[5];  
  
        for(int i = 0; i < list.length; i++) {  
            list[i] = new Rect(i, i);  
        }  
  
        for(int i = 0; i < list.length; i++) {  
            list[i] = new Rect();  
        }  
  
        for(int i = 0; i < list.length; i++) {  
            list[i].setWidth(i);  
            list[i].setHeight(i);  
        }  
  
        Rect r1 = new Rect( width: 10, height: 20);  
        Rect r2 = new Rect( width: 15, height: 21);  
        list[0] = r1;  
        list[1] = r2;  
  
        for(int i=0;i < list.length; i++)  
            System.out.println(i+"번째 사각형의 면적=" + list[i].calcArea());  
    }  
}
```

```
private static class Rect {  
    3개의 사용 위치  
    int width;  
    3개의 사용 위치  
    int height;  
  
    1개의 사용 위치  
    public Rect() {  
        //this(0, 0); // 컴파일러가 디폴트 값으로 0으로 값을 넣으므로 생략해도 넣어도 똑같음  
    }  
  
    3개의 사용 위치  
    public Rect(int width, int height) {  
        this.width = width;  
        this.height = height;  
    }  
  
    1개의 사용 위치  
    public void setWidth(int width) { this.width = width; }  
    1개의 사용 위치  
    public void setHeight(int height) { this.height = height; }  
    1개의 사용 위치  
    double calcArea() { return (double)(width*height); }  
}
```

Java의 객체 배열

```
public class ObjectListTest {  
    0개의 사용위치  
    public static void main(String[] args) {  
        Rect[] list;  
        list = new Rect[5];  
  
        for(int i = 0; i < list.length; i++) {  
            list[i] = new Rect(i, i);  
        }  
  
        for(int i = 0; i < list.length; i++) {  
            list[i] = new Rect();  
        }  
  
        for(int i = 0; i < list.length; i++) {  
            list[i].setWidth(i);  
            list[i].setHeight(i);  
        }  
  
        Rect r1 = new Rect( width: 10, height: 20);  
        Rect r2 = new Rect( width: 15, height: 21);  
        list[0] = r1;  
        list[1] = r2;  
  
        for(int i=0; i < list.length; i++)  
            System.out.println(i+"번째 사각형의 면적=" + list[i].calcArea());  
    }  
}
```



Java의 객체 배열

```
public class ObjectListTest {
    0개의 사용위치
    public static void main(String[] args) {
        Rect[] list;
        list = new Rect[5];

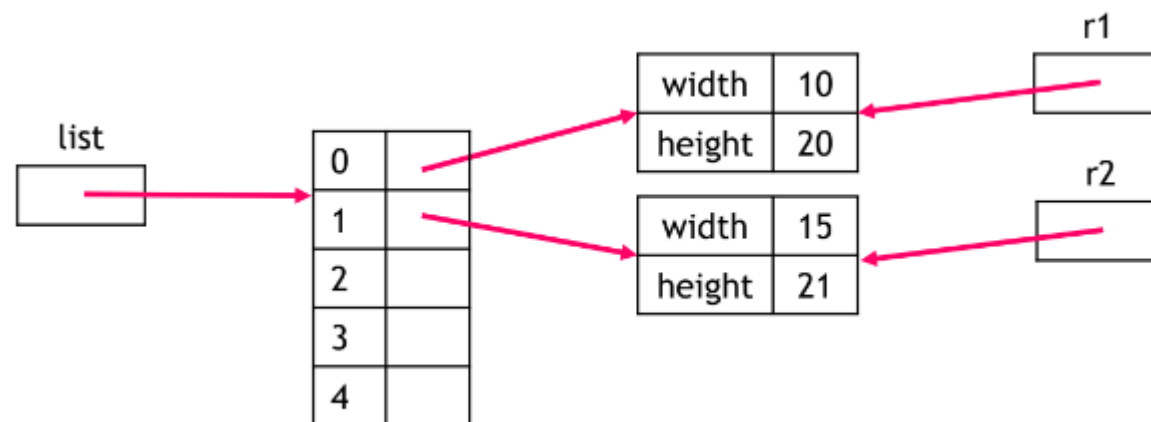
        for(int i = 0; i < list.length; i++) {
            list[i] = new Rect(i, i);
        }

        for(int i = 0; i < list.length; i++) {
            list[i] = new Rect();
        }

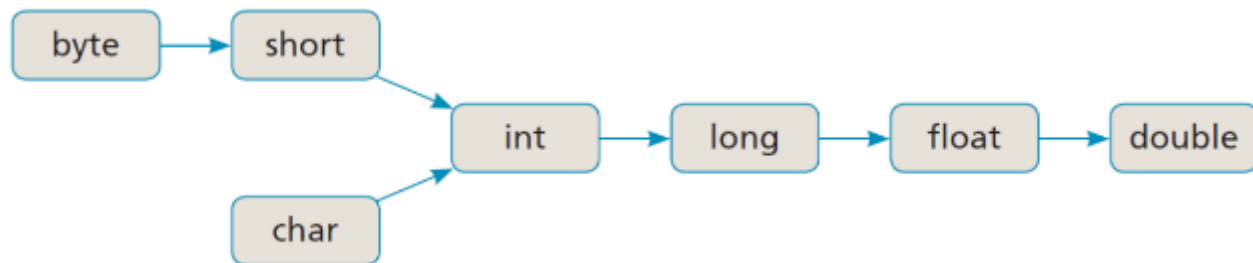
        for(int i = 0; i < list.length; i++) {
            list[i].setWidth(i);
            list[i].setHeight(i);
        }

        Rect r1 = new Rect( width: 10, height: 20);
        Rect r2 = new Rect( width: 15, height: 21);
        list[0] = r1;
        list[1] = r2;

        for(int i=0; i < list.length; i++)
            System.out.println(i+"번째 사각형의 면적=" + list[i].calcArea());
    }
}
```



Java의 형변환



- 크기가 작은 자료형을 큰 자료형으로 변경하는 경우에 자바 컴파일러가 자동으로 수행

```
double sum = 2.8 + 3; // 뒤에 있는 3은 3.0으로 형변환

int x = 3;
double y = (double) x; // 명시적으로 double을 사용한 형변환
double z = x;          // 컴파일러가 알아서 3.0으로 변환해줌

int i = (int) 1.5;     // i에는 1만 저장
```

- 형변환 연산자를 사용하는 것도 가능

연습 문제 4

```
public class Practice4 {  
    public static void main(String[] args) {  
        int i;  
        double f;  
        f = 5 / 4;  
        System.out.println(f);  
  
        f = (double) 5 / 4;  
        System.out.println(f);  
  
        i = (int) 1.3 + (int) 1.8;  
        System.out.println(i);  
    }  
}
```

- 출력 결과는 어떻게 될까요?

```
1.0  
1.25  
2
```

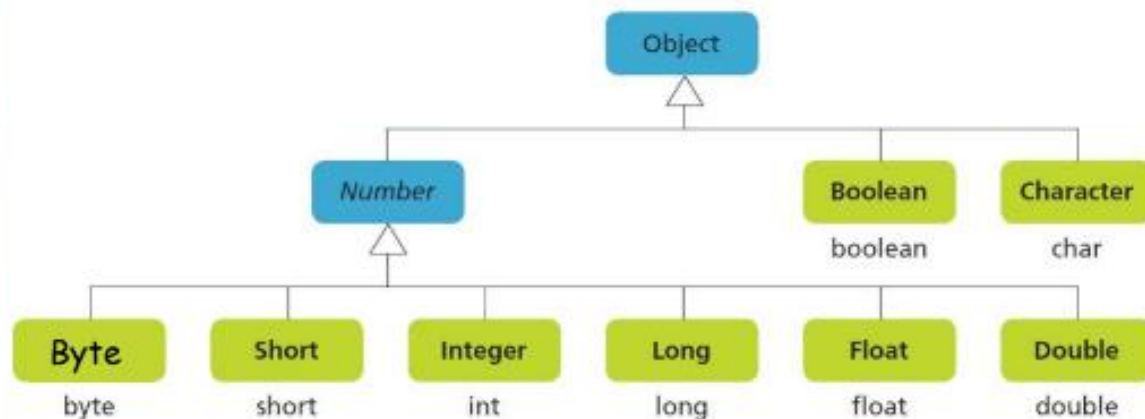
Java의 Object 클래스

- 모든 자바 클래스의 부모 클래스
- 어떤 클래스를 정의하든지 부모 클래스를 명시하지 않으면 Object 클래스의 하위 클래스
- 아래는 주로 사용하는 메소드, 오버라이딩을 해서 재정의해서 사용 가능

메소드	설명
public boolean equals(Object obj)	두 객체가 같은지 판단한다.
public String toString()	객체의 문자열 표현을 반환한다.
protected Object clone()	객체를 복사한다.
public int hashCode()	객체를 hashCode로 반환한다.

Java의 Wrapper 클래스

기초 자료형	래퍼 클래스
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
char	Character
boolean	Boolean
void	Void



- 자바의 모든 기본 자료유형에 대응하는 참조 자료유형 클래스가 존재
- 기본 자료유형을 사용할 수 없는 위치에 사용 (e.g. 제너릭)

Java의 Wrapper 클래스 – Integer 클래스

반환값	메소드 이름	설명
int	intValue()	int형으로 반환한다.
double	doubleValue()	double형으로 반환한다.
float	floatValue()	float형으로 반환한다.
static int	parseInt(String s)	문자열을 int형으로 변환한다.
static String	toBinaryString(int i)	int형의 정수를 2진수 형태의 문자열로 변환한다.
static String	toHexString(int i)	int형의 정수를 16진수 형태의 문자열로 변환한다.
static String	toOctalString(int i)	int형의 정수를 8진수 형태의 문자열로 변환한다.
static String	toString(int i)	int형의 정수를 10진수 형태의 문자열로 변환한다.
static Integer	valueOf(String s)	문자열 s를 Integer 객체로 변환한다.
static Integer	valueOf(String s, in radix)	문자열 s를 radix진법의 Integer 객체로 변환한다.

연습 문제 5

```
import java.time.LocalDate;
import java.util.Scanner;

0개의 사용위치
public class Practice5 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("주민번호 앞 7자리를 입력해주세요: ");
        String input = sc.next();

        int birthYear = Integer.parseInt(input.substring(0, 2));
        int gender = input.charAt(input.length() - 1) - '0';

        if(gender < 3)
            birthYear += 1900;
        else
            birthYear += 2000;

        LocalDate now = LocalDate.now();

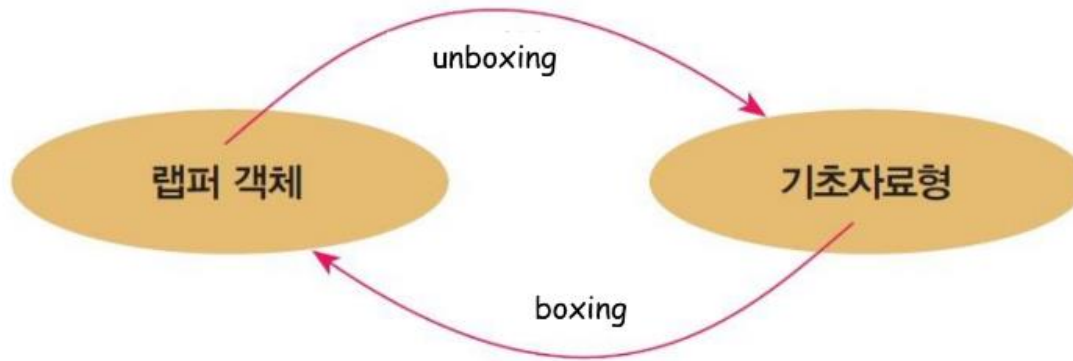
        int thisYear = now.getYear();
        int age = thisYear - birthYear + 1;

        System.out.println("당신은 " + age + "살 이군요!");
    }
}
```

- 프로그램을 작성해 봅시다.

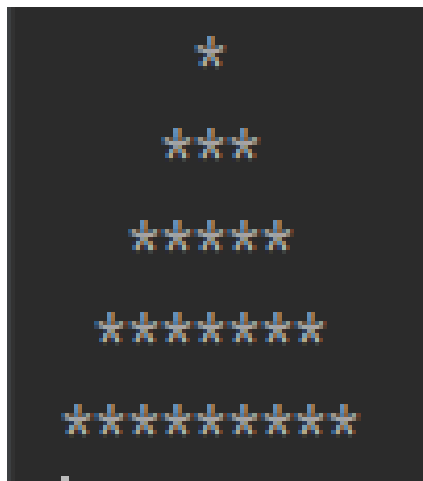
주민번호 앞 7자리를 입력해주세요: 0305104
당신은 21살 이군요!

Java의 오토박싱(auto-boxing)



```
public class AutoBoxingTest {  
    0개의 사용위치  
    public static void main(String[] args) {  
        Integer box;  
        box = 10; //정수를 자동으로 Integer 객체로 포장한다. (boxing)  
        System.out.println(box + 1); //box는 자동으로 int형으로 변환 (unboxing)  
  
        System.out.println(box.intValue() + 1); //Integer에 정의된 method를 이용한 형변환  
  
        double d1 = 10;  
        Double d2 = new Integer( value: 10);  
    }  
}
```

실습 문제 1



- 왼쪽과 같은 출력이 나오는 프로그램을 작성해 봅시다.


```
public class Practice1 {  
    public static void main(String[] args) {  
        int num = 5;  
        for(int i = 0; i < num; i++){  
            for(int j = num - 1; j > i; j--){  
                System.out.print(" ");  
            }  
  
            for(int j = 0; j < 2 * i + 1; j++){  
                System.out.print("*");  
            }  
            System.out.println();  
        }  
    }  
}
```

실습 문제 2

```
toString을 이용한 배열 전체 출력: [3.4, 5.2, 2.0, 3.0]  
배열 합 : 13.6  
배열 최대 값 : 5.2
```

- 왼쪽과 같은 출력이 나오는 프로그램을 작성해 봅시다.
- double이나 float 배열을 이용합니다.

```
import java.util.Arrays;
```

07의 사용위치

```
public class Practice2 {
```

```
    public static void main(String[] args) {
```

```
        double[] listA = { 3.4, 5.2, 2.0, 3.0};
```

```
        double sum = 0;
```

```
        double max = Double.MIN_VALUE;
```

```
        for(int i = 0; i < listA.length; i++) {
```

```
            sum += listA[i];
```

```
            if (max < listA[i])
```

```
                max = listA[i];
```

```
        }
```

```
        System.out.println("toString을 이용한 배열 전체 출력: " + Arrays.toString(listA));
```

```
        System.out.println("배열 합 : " + sum);
```

```
        System.out.println("배열 최대 값 : " + max);
```

```
    }
```

```
}
```

실습 문제 3

```
문자를 입력하세요: a
모음입니다.
문자를 입력하세요: b
자음입니다.
문자를 입력하세요: o
모음입니다.
문자를 입력하세요: i
모음입니다.
문자를 입력하세요: c
자음입니다.
문자를 입력하세요: z
자음입니다.
문자를 입력하세요: quit
프로그램 종료!
```

- 왼쪽과 같은 출력이 나오는 프로그램을 작성해 봅시다.
- “quit ” 을 제외하고는 영문 한 글자만 입력합니다.

```
import java.util.Scanner;
```

0개의 사용위치

```
public class Practice3 {
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        String input;
```

```
        while(true) {
```

```
            System.out.print("문자를 입력하세요: ");
```

```
            input = sc.next();
```

```
            if(input.equals("quit"))
```

```
                break;
```

```
            else if(input.equals("a") || input.equals("e") || input.equals("i") || input.equals("o") || input.equals("u") ||
```

```
                    input.equals("A") || input.equals("E") || input.equals("I") || input.equals("O") || input.equals("U"))
```

```
                System.out.println("모음입니다.");
```

```
            else
```

```
                System.out.println("자음입니다.");
```

```
        }
```

```
        System.out.println("프로그램 종료!");
```

```
    }
```

```
}
```

```
import java.util.Scanner;
```

0개의 사용위치

```
public class Problem3 {
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        String input;
```

```
        while(true) {
```

```
            System.out.print("문자를 입력하세요: ");
```

```
            input = sc.next();
```

```
            input = input.toUpperCase();
```

```
            if(input.equals("quit"))
```

```
                break;
```

```
            else if(input.equals("A") || input.equals("E") || input.equals("I") || input.equals("O") || input.equals("U"))
```

```
                System.out.println("모음입니다.");
```

```
            else
```

```
                System.out.println("자음입니다.");
```

```
        }
```

```
        System.out.println("프로그램 종료!");
```

```
    }
```

```
}
```

실습 문제 4

```
정수를 입력하세요: 700  
반복문을 이용한 출력: 1010111100  
재귀를 이용한 출력: 1010111100  
Integer에 있는 함수로 출력: 1010111100
```

- 왼쪽과 같은 출력이 나오는 프로그램을 작성해 봅시다.
- 입력은 0 이상의 정수만 합니다.

```
import java.util.Scanner;
```

0개의 사용위치

```
public class Practice4 {
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        System.out.print("정수를 입력하세요:");
```

```
        int num = sc.nextInt();
```

```
        System.out.println("반복문을 이용한 출력: " + toBinaryString(num));
```

```
        System.out.println("재귀를 이용한 출력: " + toBinaryString2(num));
```

```
        System.out.println("Integer에 있는 함수로 출력: " + Integer.toBinaryString(num));
```

```
    }
```

1개 사용 위치

```
    private static String toBinaryString(int num) {
```

```
        String result = "";
```

```
        while(num != 0) {
```

```
            result += Integer.toString(num % 2);
```

```
            num /= 2;
```

```
        }
```

```
        String reversed = "";
```

```
        for(int i = result.length() - 1; i >= 0; i--)
```

```
            reversed += result.charAt(i);
```

```
        return reversed;
```

```
    }
```

2개 사용 위치

```
    private static String toBinaryString2(int num) {
```

```
        if(num <= 1)
```

```
            return String.valueOf(num % 2);
```

```
        else
```

```
            return toBinaryString2(num / 2) + num % 2;
```

```
    }
```

```
}
```