제 5 강 : 기본 자료형

※ 학습목표

✓ 기본 자료형의 종류에 대해 설명할 수 있다.

✓ 자료형을 기반으로 한 Wrapper Class에 대해 설명할 수 있다.

✓ 형변환에 대해 설명할 수 있다.

1. 기본 자료형

① 논리형

기본표현	자료크기	데이터 표현 범위
boolean	1 byte	참 : true 거짓 : false

[실습] day02라는 프로젝트를 새로 만들고 작업을 수행한다.

```
1  package tommy.java.exam01;
2
3  public class BooleanEx {
4          public static void main(String[] ar) {
5                  boolean b = true;
6                  System.out.println("변수 b의 값 : " + b);
7          }
8  }
```

② 문자형

기본표현	자료크기	데이터 표현 범위
char	2 byte	0 ~ 65,535

✓ ex) ch='\u0041' ☞ unicode방식 ch=65 , ch='A', ch='\n'

✓ 아스키 코드 - 1바이트 문자를 표현, 0 ~ 255 문자범위

✓ 유니코드(세계 문자 표준) - 2바이트 문자를 표현, 0 ~ 65535 문자범위

✓ 유니코드란? 세계 여러 국가의 문자들 (한자나 한글 같은 비 영어권 문자 2바이트) 까지 표현하기에 부족해서 1바이트를 추가로 할당해서 표현한 방식

✓ http://www.unicode.org/charts/PDF/UFF00.pdf에서 참조

[실습]

```
1  package tommy.java.exam02;
2
3  public class CharEx {
4          public static void main(String[] ar) {
5                  char ch1 = 'A';
6                  char ch2 = '\u0041';
7                  System.out.println("ch1 + ch2 = " + ch1 + ch2);
8                  System.out.println("ch1 + ch2 = " + (ch1 + ch2));
9                  System.out.println("ch1 + ch2 = " + (char) (ch1 + ch2));
10         }
11 }
```

③ 정수형 : 기본형은 int

기본표현	자료크기	데이터 표현 범위
byte	1 byte	$-2^7 \sim 2^7 - 1$ (-128 ~ 127)
short	2 byte	$-2^{15} \sim 2^{15} - 1$ (-32,768 ~ 32,767)
int	4 byte	$-2^{31} \sim 2^{31} - 1$
long	8 byte	$-2^{63} \sim 2^{63} - 1$

✓ byte

- ✓ ex) byte bb = -129 ☞ 127이다 [C 에서는]
- ✓ Java에서는 컴파일 시 error를 발생시킨다. 단, 초기 값일 경우
- ✓ Why) 안정성 때문에
- ✓ But) 초기화 후 증감연산에 의한 증감에 대해선 C와 같다.

[실습]

```
1  package tommy.java.exam03;
2
3  public class ByteEx {
4          public static void main(String[] ar) {
5                  byte bb = 127;
6                  bb++;
7                  System.out.println("byte bb= " + bb);
8          }
9  }
```

✓ short

✓ short s; C에선 쓰레기 값, Java에선 error

✓ but 자동 초기 값을 할당하는 경우도 있다.

[실습]

```
1  package tommy.java.exam04;
2
3  public class ShortEx {
4          public static void main(String[] ar) {
5                  short s;
6                  System.out.println("short s = " + s); // 에러발생
7          }
8  }
```

④ 실수형 : 기본형은 double

기본표현	자료크기	데이터 표현 범위
float	4 byte	$\pm 1.4*10^{-45} \sim \pm 3.4*10^{+38}$
double	8 byte	$\pm -4.9*10^{-324} \sim \pm 1.7*10^{+308}$

[실습]

```
1  package tommy.java.exam05;
2
3  public class FloatEx {
4          public static void main(String[] ar) {
5                  float var1, var2;
6                  var1 = 3.4f;
7                  var2 = 55.55;
8                  System.out.println("var1의 값 : " + var1);
9                  System.out.println("var2의 값 : " + var2);
10         }
11 }
```

2. Wrapper Class : 기본 데이터의 클래스화

✓ byte > Byte

✓ short > Short

✓ int > Integer

✓ long > Long

✓ float > Float

✓ double > Double

✓ boolean > Boolean

✓ char > Character

[실습]

```
1  package tommy.java.exam06;
2
3  public class WrapperEx {
4          public static void main(String[] ar) {
5                  byte a_min = Byte.MIN_VALUE;
6                  byte a_max = Byte.MAX_VALUE;
7                  char b_min = Character.MIN_VALUE;
8                  char b_max = Character.MAX_VALUE;
9                  int c_min = Integer.MIN_VALUE;
10                 int c_max = Integer.MAX_VALUE;
11                 float d_min = Float.MIN_VALUE;
12                 float d_max = Float.MAX_VALUE;
13                 System.out.println("byte = " + a_min + " ~ " + a_max);
14                 System.out.println("char = " + (int) b_min + " ~ " + (int) b_max);
15                 System.out.println("int = " + c_min + " ~ " + c_max);
16                 System.out.println("float = " + d_min + " ~ " + d_max);
17         }
18 }
```

3. 형변환

종 류	설 명	코딩 예

프로모션	더 큰 자료형으로 변환(자동)
	정보의 손실 없음
		short a, b;
		a = b = 10;
		int c = a + b;

디모션	더 작은 자료형으로 변환(명시)
	정보의 손실 가능성이 있음
		int c = 0;
		short s = 10;
		c = (int)(10+3.5f);

✓ boolean 형은 형 변환 불가
✓ byte ☞ char은 casting 이다. 입출력범위가 기준이니까.
✓ long ☞ float는 promotion [why] 실수형은 정수형보다 크니까!

① 묵시적 변환
✓ 적은 데이터 형을 큰 데이터 형으로 변환할 때
✓ 컴파일러가 자동으로 변환시켜 수행
✓ double da = int_val ;

② 명시적 변환

✓ 큰 데이터 형을 적은 데이터 형으로 변환 할 때

✓ 캐스터연산자(cast operator)를 사용

✓ 값의 정밀도를 잃을 수 있다.

✓ long la = (long) float_value ;

③ 숫자를 문자열로

✓ String ss = String.valueOf(number);

④ 문자열을 숫자로

✓ byte b = Byte.parseByte(str);

✓ short s = Short.parseShort(str);

✓ int i = Integer.parseInt(str);

✓ long lo = Long.parseLong(str);

✓ float f = Float.parseFloat(str);

✓ double d = Double.parseDouble(str);