

Ch2 변수

객체지향프로그래밍(기본) 2019
경상대학교 항공우주및소프트웨어공학전공

2.1 변수의 선언

자바의 원시 데이터 타입과 변수 선언

데이터 타입	크기	표현	값의 범위
byte	8bit	정수(integer)	-128~127
short	16bit	정수(integer)	-32,768~32,767
int	32bit	정수(integer)	-2,147,483,648 ~2,147,483,647
long	64bit	정수(integer)	-9,223,372,036,854,775,808 ~9,223,372,036,854,775,807
float	32bit	부동소수점 (floating point)	$\pm 1.4e-45$ ~ $\pm 3.4028235e+38$
double	64bit	부동소수점 (floating point)	$\pm 4.9e-324$ ~ $\pm 1.7976931348623157e+308$
boolean		부울 논리(Boolean)	true, false
char	16bit	문자(character)	'\u0000'~'\uffff'

int name;
데이터 타입 변수의 이름

변수 선언시 유의점

JAVA에서 변수 이름을 선언할 때 시작은 문자나 '_', '\$'로 선언 되어야 한다. 변수 이름의 시작은 숫자로 선언될 수 없으며 JAVA에서 정의된 예약어 또한 변수의 이름으로 사용할 수 없다.

사용가능한 예	사용불가능 예
<code>name</code>	<code>123name</code> : 숫자를 첫 번째로 사용
<code>\$age</code>	<code>#key</code> : #을 사용
<code>_value</code>	<code>this</code> : 예약어 <code>this</code> 를 사용
변수	<code>my name</code> : <code>my</code> 와 <code>name</code> 사이에 공백

변수 선언시 유의점

<code>abstract</code>	<code>assert</code>	<code>boolean</code>	<code>break</code>	<code>byte</code>
<code>case</code>	<code>catch</code>	<code>char</code>	<code>class</code>	<code>const</code>
<code>continue</code>	<code>default</code>	<code>do</code>	<code>double</code>	<code>else</code>
<code>enum</code>	<code>extends</code>	<code>final</code>	<code>finally</code>	<code>float</code>
<code>for</code>	<code>goto</code>	<code>if</code>	<code>implements</code>	<code>import</code>
<code>instanceof</code>	<code>int</code>	<code>interface</code>	<code>long</code>	<code>native</code>
<code>new</code>	<code>package</code>	<code>private</code>	<code>protected</code>	<code>public</code>
<code>return</code>	<code>short</code>	<code>static</code>	<code>strictfp</code>	<code>super</code>
<code>switch</code>	<code>synchronized</code>	<code>this</code>	<code>throw</code>	<code>throws</code>
<code>transient</code>	<code>try</code>	<code>void</code>	<code>volatile</code>	<code>while</code>

자바의 예약어

2.1.1 정수 선언

정수는 기본적으로 int 타입을 사용하며 값의 범위에 따라 byte, short, long 타입을 사용한다.

```
1 public class Example {  
2     public static void main(String[] args) {  
3         int ten = 10;  
4         byte ten2 = 0b1010; ← 2진수 표현  
5         short ten8 = 012; ← 8진수 표현  
6         int ten16 = 0xA; ← 16진수 표현  
7         long longNumber = 3000000000L;  
8  
9         System.out.println(ten);  
10        System.out.println(ten2);  
11        System.out.println(ten8);  
12        System.out.println(ten16);  
13        System.out.println(longNumber);  
14    }  
15 }
```

결과

```
10  
10  
10  
10  
3000000000
```

2.1.2 실수 선언

실수는 부동소수점으로 표현되며 기본적으로 double 타입을 사용하고 값의 범위에 따라 float 타입을 사용한다.

```
1 public class Example {  
2     public static void main(String[] args) {  
3         double pi = 3.14;  
4         double pie = 314e-2;  
5         float pif = 3.14f;  
6         float pief = 314e-2f;  
7  
8         System.out.println(pi);  
9         System.out.println(pie);  
10        System.out.println(pif);  
11        System.out.println(pief);  
12    }  
13 }
```

결과

3.14
3.14
3.14
3.14

2.1.3 문자 선언

문자는 유니코드를 이용하며 char 타입을 사용한다. 0~65,535 범위의 정수로 표현 할 수 있어 일종의 정수 타입으로 생각 할 수 있다.

```
1 public class Example {  
2     public static void main(String[] args) {  
3         char cNumber = 97;  
4         char cAlphabet = 'a';  
5         char cUnicode = '\u0061';  
6         char cHan = '한';  
7         char cYou = 'π';  
8  
9         System.out.println(cNumber);  
10        System.out.println(cAlphabet);  
11        System.out.println(cUnicode);  
12        System.out.println(cHan);  
13        System.out.println(cYou);  
14    }  
15 }
```

결과

```
a  
a  
a  
한  
π
```

2.1.4 논리 선언

논리 값은 참과 거짓의 값을 저장하기 위해 이용하며 boolean 타입을 사용한다. 변수 선언 시 데이터 값으로 true나 false로 저장할 수 있다.

```
1 public class Example {  
2     public static void main(String[] args) {  
3         boolean TRUE = true;  
4         boolean FALSE = false;  
5  
6         System.out.println(TRUE);  
7         System.out.println(FALSE);  
8     }  
9 }
```

결과

true

false

2.2 숫자 데이터 타입의 범위

데이터 타입	최솟값	최댓값
byte	Byte.MIN_VALUE	Byte.MAX_VALUE
short	Short.MIN_VALUE	Short.MAX_VALUE
int	Integer.MIN_VALUE	Integer.MAX_VALUE
long	Long.MIN_VALUE	Long.MAX_VALUE
float	Float.MIN_VALUE	Float.MAX_VALUE
double	Double.MIN_VALUE	Double.MAX_VALUE

```
1 public class Example {
2     public static void main(String[] args) {
3         byte byteMin = Byte.MIN_VALUE;
4         byte byteMax = Byte.MAX_VALUE;
5
6         short shortMin = Short.MIN_VALUE;
7         short shortMax = Short.MAX_VALUE;
8
9         int intMin = Integer.MIN_VALUE;
10        int intMax = Integer.MAX_VALUE;
11
12        long longMin = Long.MIN_VALUE;
13        long longMax = Long.MAX_VALUE;
14
15        float floatMin = Float.MIN_VALUE;
16        float floatMax = Float.MAX_VALUE;
17
18        double doubleMin = Double.MIN_VALUE;
19        double doubleMax = Double.MAX_VALUE;
20
21        System.out.println(byteMin+" ~ "+byteMax);
22        System.out.println(shortMin+" ~ "+shortMax);
23        System.out.println(intMin+" ~ "+intMax);
24        System.out.println(longMin+" ~ "+longMax);
25        System.out.println(floatMin+" ~ "+floatMax);
26        System.out.println(doubleMin+" ~ "+doubleMax);
27    }
28 }
```

결과

-128 ~ 127

-32768 ~ 32767

-2147483648 ~ 2147483647

-9223372036854775808 ~ 9223372036854775807

1.4E-45 ~ 3.4028235E38

4.9E-324 ~ 1.7976931348623157E308

오버플로우

데이터 범위를 넘어가는 값이 변수에 저장되었을 경우 변수는 제대로 된 정보를 표현 하지 못한다. 이러한 상황을 변수의 데이터가 오버플로우(Overflow)되었다고 한다.

```
1 public class Example {  
2     public static void main(String[] args) {  
3         byte x= Byte.MAX_VALUE;  
4         System.out.println(x);  
5  
6         x++; ← x 변수의 값을 1 증가 시킴  
7         System.out.println(x);  
8     }  
9 }
```

결과

127

-128

2.3 데이터 타입 변환

데이터 타입을 변화시키기 위해 변환시킬 변수 앞에 변환시킬 데이터 타입을 ()에 넣어 표기한다.

```
1 public class Example {  
2     public static void main(String[] args) {  
3         int intNumber = 200;  
4         double doubleNumber = 3.141592;  
5  
6         System.out.println((short)intNumber);  
7         System.out.println((byte)intNumber);  
8         System.out.println((int)doubleNumber);  
9     }  
10 }
```

결과

200

-56

3

데이터 타입 변환

데이터 타입을 변화시키기 위해 변환시킬 변수 앞에 변환시킬 데이터 타입을 ()에 넣어 표기한다.

```
1 public class Example {  
2     public static void main(String[] args) {  
3         int intNumber = 200;  
4         double doubleNumber = 3.141592;  
5  
6         System.out.println((short)intNumber);  
7         System.out.println((byte)intNumber);  
8         System.out.println((int)doubleNumber);  
9     }  
10 }
```

결과

200

-56

3

자동 숫자 데이터타입 변환 문제

```
1 public class Example {  
2     public static void main(String[] args) {  
3         int intNumber ;  
4         double doubleNumber;  
5  
6         intNumber = 3/4;  
7         doubleNumber = 3/4;  
8  
9         System.out.println(intNumber);  
10        System.out.println(doubleNumber);  
11    }  
12 }
```

결과

0
0.0

자동 숫자 데이터타입 변환 문제 해결

```
1 public class Example {
2     public static void main(String[] args) {
3         double doubleNumber;
4
5         doubleNumber = 3/((double)4); ← 행변환 연산자를 사용하여 데이터 타입을 변경
6         System.out.println(doubleNumber);
7
8         doubleNumber = 3.0/4; ← 정수 3을 실수 3.0으로 변경
9         System.out.println(doubleNumber);
10    }
11 }
```

결과

0.75

0.75