

## 2-3.演算子

提出する

評価を受ける

### はじめに

演算子には、「算術演算子」、「代入演算子」、「比較演算子」、「論理演算子」などがあります。  
この演算子を使い「式」を作ることが可能です。  
「算術演算子」「代入演算子」を使えば「計算式」になり、  
「比較演算子」を使えば「条件式」になり、  
「論理演算子」を使えば「論理式」になります。

### 算術演算子

記号	機能
+	加算
-	減算
*	乗算
/	除算
%	剰余

「算術演算子」を使い計算式を作ります。

```
int i = 0;
// 10 と 5 を加算した結果 (15) を、変数「i」に代入
i = 10 + 5;
```

#### 文字列の連結

文字列を使い、「+」の演算子で計算式を作ることが可能です。  
この場合、加算ではなく文字列の連結となりますので注意してください。

```
String a = "ABC";
String b = a + "DEF"; // 変数bには、連結結果の文字列 ("ABCDEF") が保持される。
```

### 代入演算子

代入演算子は、変数に代入する値に何かしらの計算を行い、その結果を再び同じ変数に代入する際に便利です。

記号	機能	意味
A += B	AとBを足してAに代入	A = A+B
A -= B	AからBを引いてAに代入	A = A-B
A*=B	AとBを掛け合わせてAに代入	A = A*B
A /= B	AをBで割りAに代入	A = A/B

?

記号	機能	意味
A %= B	AをBで割りその余りをAに代入	A = A%B
A++ (インクリメント)	Aから値を1増やす	A+1
A-- (デクリメント)	Aから値を1減らす	A-1

代入演算子を使い、条件式を作ります。

```
int a = 10;
a += 5; // 結果は15
a -= 3; // 結果は7
a *= 4; // 結果は40
a /= 2; // 結果は5
a %= 6; // 結果は4
```

## 比較演算子

比較演算子は、左辺と右辺を比較し、数字の大小などを評価します。そして結果として真(true)か偽(false)の値を取ります。

記号	機能	別名
A == B	AとBが等しい時にtrue	eq
A != B	AとBが等しくない時にtrue	ne, neq
A > B	AがBより大きい時にtrue	gt
A < B	AがBより小さい時にtrue	lt
A >= B	AがBより大きいか等しい時にtrue	gte, ge
A <= B	AがBより小さいか等しい時にtrue	lte, le

### 文字列の比較

文字列の比較は特別です。文字列は、比較演算子で比較することはできません。  
文字列を比較する際は、Stringクラスにある「equals」という名前のメソッド（関数）を利用します。

### 【サンプル: 比較演算子】

```
// 変数「i」の値が 1 の場合は 真(true)、1 でない場合は 偽(false) となる。
i == 1;

int i = 1;
// 変数「i」の値が 1 の場合は 真(true)、1 でない場合は 偽(false) となる。
System.out.println(i == 1);

i = 2;
System.out.println(i != 1);
```

### 【出力結果】

```
true
true
```

## 論理演算子

論理演算子は複数の比較演算子の結果を組み合わせる場合に使用されます。  
そのため、左辺や右辺には比較演算式が入ります。  
そしてその左辺と右辺を比べた結果として、真(true)か偽(false)の値を取ります。

記号	機能
A && B	AとBが共にtrueの場合にtrue (かつ)
A    B	AかBのどちらかがtrueの場合にtrue(または)
!A	Aがtrueでないときにtrue(ではない)

記号の読み方

- **&** = アンパサンド
- **|** = パイプライン or パイプ
- **!** = エクスクラメーションマーク

比較演算子、論理演算子を使い論理式を作ります。

【サンプル: 論理演算子】

```
// 「i >= 5」と「i <= 20」のどちらも正しければ 真(true)、それ以外の場合は 偽(false) となる。
int i = 10;
System.out.println(i >= 5 && i <= 20);

// 「a > 20」と「b < 30」のどちらかが正しければ 真(true)、どちらも正しくない場合は 偽(false) となる。
int a = 10;
int b = 20;
System.out.println(a > 20 || b < 30);

// 「i < 20」の結果が「!」によって否定される。
int i = 10;
System.out.println(!(i < 20));
```

【出力結果】

```
true
true
false
```

ポイント

式の評価は、演算子の優先順位に従いながら **左から** 行われます。  
複数の演算子が含まれる式を記述する場合、どの順番で計算が行われるのか注意しましょう。  
上記の「**a > 20 || b < 30**」の式の場合は以下の順序になります。

- 優先順位1 「>」と「<」
- 優先順位2 「||」

提出課題

提出ファイル

・Task2\_3.java  
**2-3/Task2\_3.java** に課題がありますので、指示に従って記述して下さい。

評定概要

学生から秘匿	No
--------	----

参加者	80
提出	55
要評定	0