



独習 Java (第3版)

- 3.6 バックスラッシュコード
- 3.7 関係演算子と論理演算子
- 3.8 三項演算子

発表者

佐々木研 田島 勇樹

3.6 バックスラッシュコード(1)

- printメソッドなどに使われる特殊な文字コードのこと
エスケープシーケンスとも呼ばれる。
先頭に円記号(¥)をつけて表す。

例

```
class NewlineDemo{
    public static void main(String args[]){
        System.out.print("This is line one\n");
        System.out.print("This is line two\n");
        System.out.print("This is line three\n");
    }
}
```

表 バックスラッシュコード

コード	意味
¥b	バックスペース
¥n	改行
¥r	キャリッジリターン
¥f	フォームフィード
¥t	水平タブ
¥"	二重引用符
¥'	一重引用符
¥0	null値
¥¥	バックスラッシュ
¥N	8進定数(N=0~7)
¥NN	8進定数(N=0~7)
¥MNN	8進定数(M=0~3,N=0~7)
¥uxxxx	Unicode文字(yyyyは4桁の16進数定数)



3.6 バックslashコード(2)

- Unicode文字を出力するプログラム

```
class SpecialCharacters{
    public static void main(String
        args[]){
        System.out.print("\u00a0 \u00a1
            \u00a2 \u00a3");
    }
}
```

出力結果

?? ¢ £

```
class SpecialCharacters{
    public static void main(String
        args[]){
        System.out.println("ABCD\rE");
        System.out.print("ABCD\fE");
    }
}
```

出力結果

EBCD

ABCD E



3.6 バックスラッシュコード(3)

- 改行文字を出力するプログラム

```
class EmbeddedNewlines{  
    public static void main(String args[]){  
        System.out.print("one\n\two\n\tthree\n\tfour");  
    }  
}
```

出力結果

one

two

three

four



3.7 関係演算子と論理演算子(1)

- 関係演算子とは…
2つのオペランドの間関係をテストする演算子
評価結果は論理型Trueまたはfalse

表 1 関係演算子

演算子	意味	例
==	等しい	i==j
!=	等しくない	i!=j
>	より大きい	i>j
<	より小さい	i<j
>=	より大きいか等しい	i>=j
<=	より小さいか等しい	i<=j



3.7 関係演算子と論理演算子(2)

関係演算子の使い方のプログラム

```
class RelationalOperatorDemo{
    public static void main(String args[]){
        int i=Integer.parseInt(args[0]);
        int j=Integer.parseInt(args[1]);
        System.out.println("i = "+ i);
        System.out.println("j = "+ j);
        System.out.println("Relational operators:");
        System.out.println("i < j "+ (i < j));
        System.out.println("i <= j "+ (i <= j));
        System.out.println("i == j "+ (i == j));
        System.out.println("i > j "+ (i > j));
        System.out.println("i >= j "+ (i >= j));
        System.out.println("i != j "+ (i != j));
    }
}
```

出力結果

```
i=2
j=5
Relational operators:
i < j true
i <= j true
i == j false
i > j false
i >= j false
i != j true
```



3.7 関係演算子と論理演算子(3)

- 論理演算子とは…

論理演算子は、論理値をもとに、NOT、AND、OR などの論理演算を行う演算子

表2 論理演算子

演算子	動作	説明
&	論理積	AND。2つの項が真のときにだけ結果が真となる論理演算
	論理和	OR。2つの項が偽のときにだけ結果が偽となる論理演算
^	排他的論理和	Exclusive OR。2つの項が異なるときだけ結果が真となる論理演算
!	否定	NOT。項の真偽が逆になる論理演算
&&	論理積(ショートカット)	
	論理和(ショートカット)	
==	等しい	
!=	等しくない	

3.7 関係演算子と論理演算子(4)

- 論理演算子は真理値表に基づいて論理積、論理和、排他的論理和、否定の各論理演算をサポート

表3 真理値表

p	q	p&q	p q	p^q	!p
偽	偽	偽	偽	偽	真
偽	真	偽	真	真	真
真	偽	偽	真	真	偽
真	真	真	真	偽	偽

- ・ショートカット演算子とは左のオペランドで結果が判定できるなら右のオペランドは評価しない演算子のこと
&&と||がこの演算子に含まれる。

例 ((a >= 0) & (c > b / a))

この場合2つのオペランドは評価される

((a >= 0) && (c > b / a))

この場合aが負なら左のオペランドだけ評価され右のオペランドは評価しない



3.7 関係演算子と論理演算子(5)

論理演算子を使ったプログラム

```
class LogicalOperatorDemo{
    public static void main(String args[]){
        boolean
            i=Boolean.valueOf(args[0]).boolean
            Value();
        boolean
            j=Boolean.valueOf(args[1]).boolean
            Value();
        System.out.println("i = "+ i);
        System.out.println("j = "+ j);
        System.out.println("Logical
            operators:");
        System.out.println("i & j "+ (i & j));
        System.out.println("i | j "+ (i | j));
        System.out.println("i ^ j "+ (i ^ j));
        System.out.println("!i "+ !i);
```

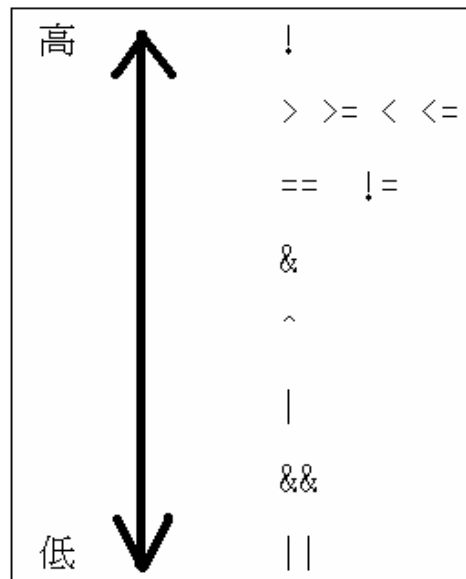
```
System.out.println("i && j "+ (i && j));
System.out.println("i || j "+ (i || j));
System.out.println("i == j "+ (i == j));
System.out.println("i != j "+ (i != j));
    }
}
```

出力結果

```
2 5
i = false
j = false
Logical operators:
i & j false
i | j false
i ^ j false
!i true
i && j false
i || j false
i == j true
i != j false
```

3.7 関係演算子と論理演算子(6)

- 関係演算子と論理演算子の優先順位



関係演算子、論理演算子 < 算術演算子
という関係を持つ

例 $10 + \text{count} > a + 12$

は次と等価である

$(10 + \text{count}) > (a + 12)$

関係演算子を論理演算子でつなぐことも
可能である。

例 $\text{var} > \text{max} \ || \ !(\text{max} == 100) \ \&\& \ 0 \leq \text{item}$



3.8 三項演算子

- 三項演算子の形式
 `expr1 ? expr2 : expr3;`
expr1: 任意の論理式
expr2, expr3: 同じ型の値
論理式が真 `expr2`が評価
論理式が偽 `expr3`が評価

```
Class TernaryOperatorDemo{  
public static void main(String args[]){  
int i=10,j=5;  
System.out.println((i > j) ? i : j);  
}  
}
```

出力結果
10



練習問題

- “あいうえおいろふにほと”と出力するプログラムがある。これを“”いろはにほへと”と表示されるようにバックスラッシュコードを加えてプログラムを作成せよ。ただしバックスペースは1回しか使用してはいけない。
- りんごの値段が120円であった。りんごを5個以上買った人でかつ、うるう年の人は2割引される。個数と生まれた西暦を入力し、合計の値段を表示するプログラムを作成せよ。
- 三項演算子を使って入力した数が偶数か奇数か判別するプログラムを作成せよ。