

データの種類と型 p.38

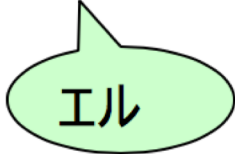


データ	型（データの種類）
オブジェクト 例： <code>new JFrame()</code>	クラス <code>JFrame</code>
値 例： <code>3.14</code>	基本型 <code>double</code>

基本型 整数

基本型 (ビット数) 内容 リテラルの例

- byte (8) 8ビット整数
- short (16) 短長整数
- int (32) 整数 3, -10
- long (64) 倍長整数 100L, -51



エル

リテラル: ソースプログラム中の定数

基本型 実数

基本型 (ビット数) 内容 リテラルの例

- float (32) 単精度実数 3.14F, -5.0e+10f
- double (64) 倍精度実数 3.14, 3.0e+12



3.0e+12の意味: 3.0×10^{12}

実数はコンピュータ内部では浮動小数点(floating point)で表現されているので、浮動小数点数(floating point number)とも呼ばれる

基本型 論理値、文字

基本型 (ビット数) 内容 リテラルの例

- boolean (8) 論理値 true, false
 - char (16) 文字 'a' 'b' '理'
-

- 文字列は基本型ではない！
Stringクラスのインスタンス

文字列 "abc", "0", ""

↓
空の文字列

包囲クラス (wrapper class)

基本型 対応する包囲クラス

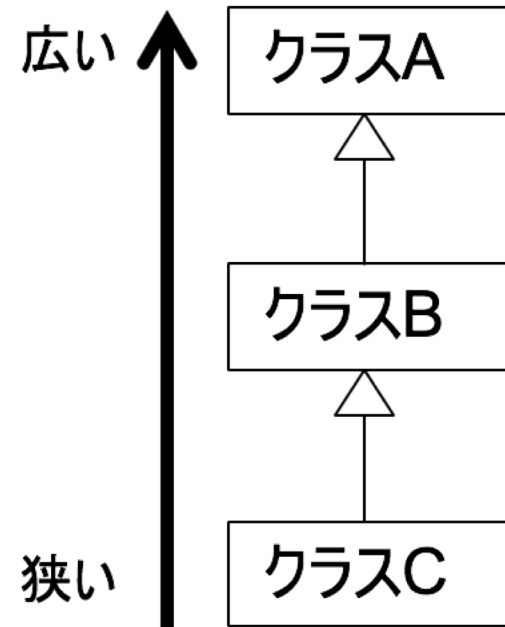
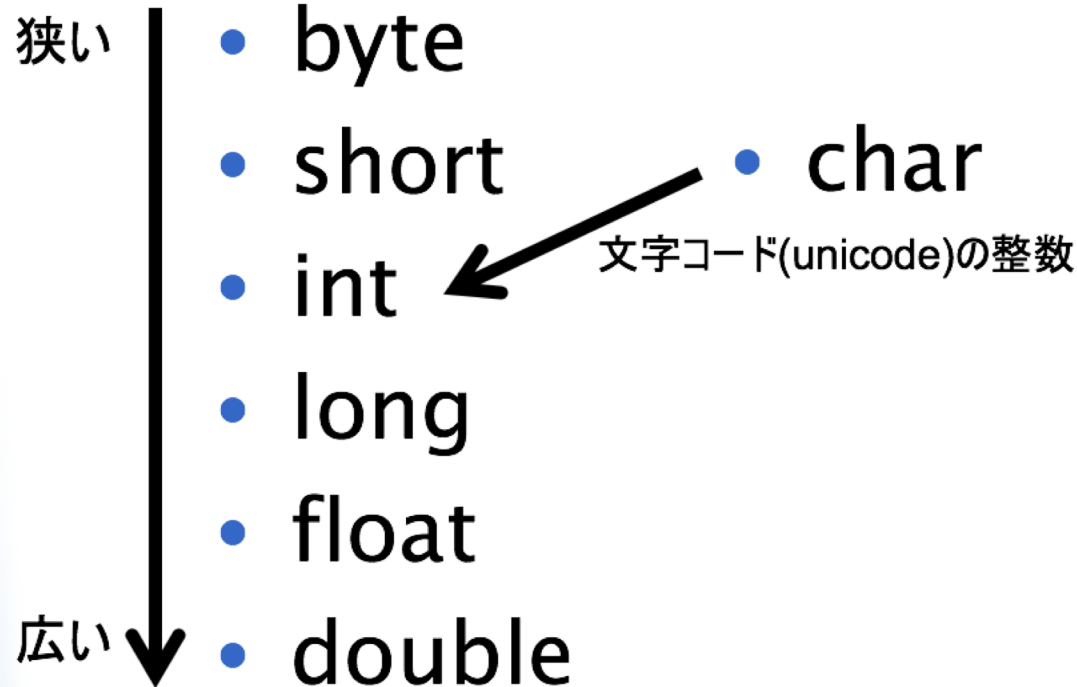
- | | |
|-----------|-----------|
| • byte | Byte |
| • short | Short |
| • int | Integer |
| • long | Long |
| • float | Float |
| • double | Double |
| • boolean | Boolean |
| • char | Character |

包囲クラスのインスタンスはその内部に対応する基本型の値を保持する

包囲クラスと基本型の自動変換

- ボクシング (boxing)
 - ◆ 基本型 → 包囲クラス
 - `Integer i1 = 100; // new Integer(100) と同等`
- アンボクシング (unboxing)
 - ◆ 包囲クラス → 基本型
 - `int k1 = i1; // i1.intValue() と同等`

広め変換(型指定不要)

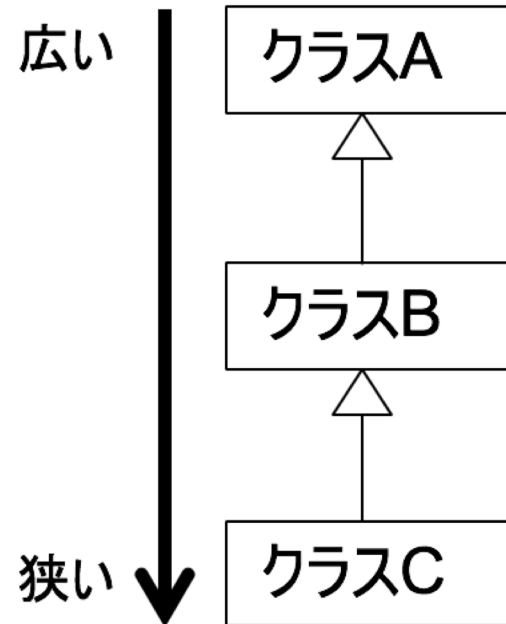
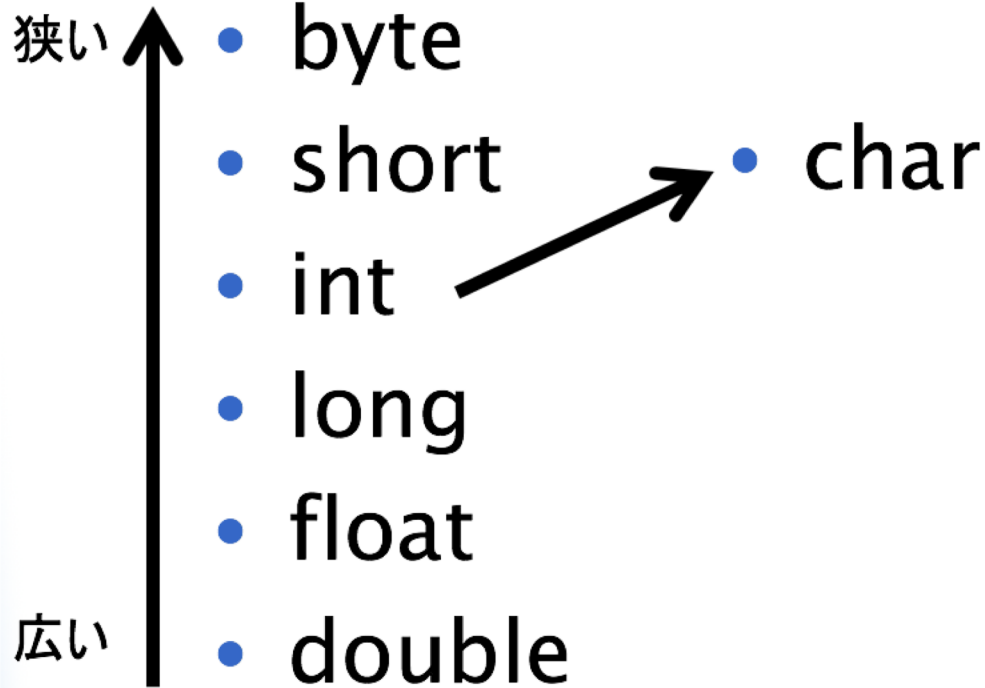


広め変換の例:

```
long x = 100;  
float f1 = 1.0f + x;
```

double型とint型の比較でも
int型がdouble型に広め変換され
比較が行われる

狭め変換(キャスト必要)



キャストの例:

```
double d1 = 3.14;  
int i1 = (int) d1;
```