

プログラミング入門

第3回

箕原辰夫

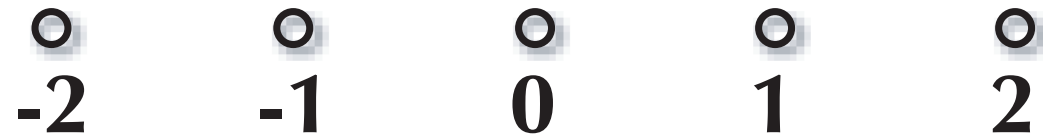
2005年秋学期

式と型

- 式の構成要素
 - 定数
 - 変数
 - 演算子
- Javaの式には、どのような型であるのかという、型の識別が伴う
 - 整数型、実数型、文字型…

整数と実数の違い

Integer(離散数)



Real(連続数)



アプリケーションで記述

- アプリケーションのテンプレート
 - Xcodeでは、JavaToolを
 - JEditでは、そのまま
- `System.out.println( );`の括弧の中に定数や式を記述していく

論理型

- 型名は、**boolean**
- 真と偽を表わす 2 つの値しか持たない
 - ➔ **true**
 - ➔ **false**

整数

- ビット数によって表わせる範囲が違う
 - **byte, short, int, long** 標準は**int**
 - 整数の定数は**int**型として扱われる
 - **long**型の整数の定数を記述するときは**L**あるいは**l**を最後につける
- 8進数、16進数も記述できる
 - 2進数3桁→8進数、2進数4桁→16進数

実数

- 単精度実数→**float**
 - 単精度実数の定数を明示するには、最後にFあるいはfをつける
- 倍精度実数→**double**
 - 実数定数をそのまま記述すると倍精度
- 指数表現と正規化

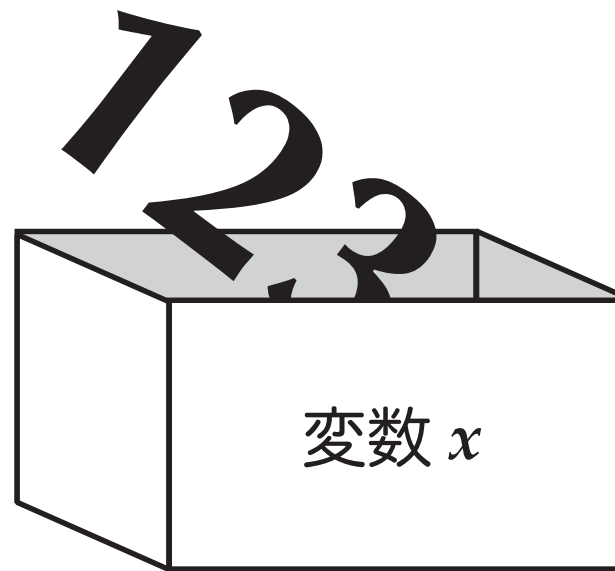
文字定数

- 1 文字を表わす **char**型
 - 定数はシングルクォート (') で囲む
 - 文字コードは、内部的にはUnicodeを使って表わされている
- Unicodeを見してみる
 - 文字コード (スクリプトメニューから)
 - UnicodeのWebサイト
www.unicode.org

文字列定数

- 文字列とは、複数の文字を連続して記述できる型である。
- 文字列はJavaでは、オブジェクトとして扱われている。Stringクラス
 - ➔ 文字列定数は、ダブルクォーテーションマーク (") で囲む

変数について



変数は、定数の値を一時的に覚えておくことができる。箱の中に入っていると考えるとわかりやすい。

変数の宣言

- 型を指定して宣言する必要がある
 - ➔ 型制約のある言語、Java, C/C++など
 - * 事前にコンパイルされるもの
- 型を指定しない言語もある
 - ➔ 型なし言語、JavaScript, AppleScript, Lisp, Prologなど
 - * インタプリタ方式で実行される

変数への代入

- 変数へは、まず最初に値を代入する必要がある
 - 変数 = 式;
- 変数を左に書く。この=は、右から左に代入が行なわれる代入演算子。
- 変数の値の参照
 - 式の中に記述できる

自己参照による代入

- それまで持っていた値を使って、変数の新しい値を代入できる
- =の左辺に変数が出現したら
 - ➔ その変数に新しく値が代入される
- =の右辺に変数が出現したら
 - ➔ その変数がそれまで持っていた値が参照される

再帰代入文

- 自己参照を行なう代入には、演算子ごとに省略形が用意されている
 - ➔ `+=`, `-=`, `*=`, `/=`, `%=`

インクリメント演算子

- +1, -1するだけは、特別な演算子がある
 - アセンブラ、C言語からの踏襲
 - ++と--
- 演算子を変数の前につけるか後ろにつけるかで副作用が異なる
 - 前（左につける：++x）
 - * 値を参照される前に更新される
 - 後（右につける：x++）
 - * 先に参照され、その後値を更新

式と評価

- 評価 (Evaluation) とは
 - 単一の値になるまで計算すること
- 式の様式に合っているか
 - 式に合っていないと文法エラー
- 式の評価の順番
 - 優先順位がある

整数演算

- 整数除算は、小数点以下が切り捨てられる
- どこに整数除算があるか重要
 - $5 / 2 * 2 \rightarrow 4$
 - $5 * 2 / 2 \rightarrow 5$
- 剰余算は、余りを計算する（実数でも可）
 - $365 \% 20 \rightarrow 5$
 - $x \% n \rightarrow 0 \sim n-1$ の数しか出てこない計算結果が0になるときは、割り切れる

型の変換

- 暗黙の型変換
 - 整数から実数へ
 - 実数が出てくると実数へ自動的に変換
- 明示的な型変換
 - キャストの書式を使う
- 文字列への変換
 - 文字列と足し算する