

教科書第4章「式と演算子」を理解し、次の問に答えられるようにする。

- 式の形(書き方)のルールを理解する。式の値(評価)の考え方を理解する
- どういう演算子があるか? 数学では見かけない数値の演算子や、論理演算子の機能・動作を理解する
- 演算子の優先順位に気を遣うことを理解する
- 型の変換について、自動的に起こる場合・プログラムで(明示的に)起こす場合、不整合の場合の動作を理解する

1. 前回の小テストのコメント

- 1) ② float型の制限? どの型もサイズが決まっており、そのサイズを超えた大きさの値をいれることはできない。

int型の場合、教科書47ページにあるように-2147383648~2147483647より外側の数は、整数であっても入れられない。それは箱の大きさが決まっているからと考えればよい。同様に、float型にも大きさの制限があるが、「浮動小数点数」の表現方法はやや複雑で、単純に「桁数が...」と言えないところがある。だから教科書にもそう書いていない。→ 1年秋学期の「コンピュータアーキテクチャ」で浮動小数点小数のコンピュータ内部での表し方を学び、この疑問を解決する。

- 1) ③ char型は、2バイトと教科書に書いてある。(この授業では扱わないが) C言語では1バイトだったりするので、将来注意が必要。英数字(アルファベットと数字)は従来は1バイト(8ビット)で表現されてきた(ASCIIコード)が、Javaや最近の言語ではユニコードで表すので英数字は2バイトで表される。
- 2) ポイントは、変数chがchar型と宣言されているのに、次の行の代入文ch = 3.14では「3.14」という数値(小数)を代入しようとしている。このため、型が合わないというエラーが起きる。(ぜひ自分でJCPad上で試してみたい)

解答の書き方として、「ch = 3.14がまちがい」「ch = 3.14でエラーを起こす」よりは、もう少し理由を説明して欲しい。つまり、変数chが文字型であること、そこへ(浮動小数型の)数値3.14を代入しようとしていること、そのため型の不整合のエラーとなること、を指摘して欲しいのである。

(注意) 変数の名前chを、小数向けの名前dbに変えるべし、というコメントを書く人がいるが、これは

*見やすくする、見慣れたものにする、という意味で、間違いではない(正しい)のだが、

*プログラム言語から見ると、変数の名前は(浮動小数型であろうと)chのまま構わない。chを必ず文字型に使う、とか文字型にしか使ってはならない、とかいうルールはない。もちろん、直しても構わないのだが.....

- 3) 教科書p 64のSample5.java をマネして書けばよい。念のためp 62のSample4.javaとの違いをよく見ておいて欲しい。p 62に比べてp 64は、下から4行目の int num = Integer.parseInt(str); の1行が多い。その前の行から見ると、

```
String str = br.readLine();           ← ①
int num = Integer.parseInt(str);      ← ②
```

となっていて、①の行でキーボードから文字(列)を読み取り、String型の変数strへ代入する。たとえばキーボードから"18"という文字列(引用符"は入力しないが)を入力すると、変数strには文字列"18"が入る。

次に②でそのstrに対してInteger.parseInt(str)という操作をする(関数を適用する)。この関数は、与えられた文字列の中身が「整数の数値」を表す文字列(たとえば"18"とか"-345"とか)であれば、それに対応する整数値に変換して返す。プログラムではその返された値を(整数型の)変数numに代入しているため、変数numの値はキーボードから入力した数字に対応する整数になる。たとえば18と入力すれば、numの値は整数の18になる。だから、次の行で2019-numを計算すると、生まれ年になる、というわけである。

(注意) なお、int num = Integer.parseInt("18");と書いた人がいるが、こうするとキーボードから読み出した文字列を渡しておらず、決まった文字列"18"を渡しているため、プログラムは動作するが、表示はキーボード入力に依らず同じ2001を表示するだろう。たとえば、私の年齢66を入れてみて欲しい。1953年と出てくれば正解である。

2. 式の形、式の値

- 教科書の図4-1を参照 ⇒ たとえば、 <オペランド> (演算子) <オペランド>
1 + 2 については、オペランドは () と () で、演算子は ()
それでは、 1 + 2 + 3 では? (1 + 2) × 5 では?
- 「式」は、(計算できて)、値が定まる。 式の値を計算することを、式を () する という。
- いろいろな演算: 演算の対象(オペランド)はいろいろなものがある。
1 + 2 オペランド1や2は ()。この時はその値(書かれている値)を使う。
num1 + num2 オペランドnum1やnum2は ()。この時は () を使う。
では、 num1 + (num2 + 5) はどう解釈しようか? (教科書のp 77の囲み記事)
- 教科書p 74のSample2.javaを解釈・説明してみよう。
- 「変数は (その時々で) いろいろな値を格納」 ⇒ その変数を使った式の値も変わる
- 教科書p 76のSample3.javaを解釈・説明してみよう。
[演習] Sample3.javaを実行してみよう。

3. 演算子の種類

- 教科書表4-1を眺めてみよう。
それぞれの演算子の意味は、数学で使うものと同じもの、そうでないものがある。そうでないものは順に理解しよう
オペランドが1つの演算子=単項演算子 (たとえば -xのマイナスとか)。 2つの演算子=2項演算子。
オペランドの種類(型)によって意味が違う演算子もある。特に () 演算は数値と文字列で意味が異なる。
- 数学では出てこない演算子のいくつかを見ておこう。

インクリメント・デクリメント演算子

インクリメント: 書き方 () 意味・動作 ()
デクリメント: 書き方 () 意味・動作 ()
前置と後置の違い 前置 b = ++a; ()
後置 b = a++; ()

代入演算子 ① 代入すること自体が演算である ② 他の演算と組み合わせた使い方がある

組み合わせた代入演算: a += b; 意味は ()

もし aの値が3、bの値が2であれば、a += b; を実行した後の aの値は ()

(注意) 教科書 p 86 の最後の行: ()

シフト演算子 (2進数だと見なして) 桁をずらす(シフトする)

左シフト

5 = () (2進表記で。面倒なので16桁だけ書こう~本当は32桁)

5 << 2 = () = (10進に直すと) ()

(注意) 右側にできた「空き」桁には0を入れる。

右シフト

5 = () (2進表記で。面倒なので16桁だけ書こう~本当は32桁)

5 >> 2 = () = (10進に直すと) ()

(注意) 左側にできた「空き」桁には「正の数なら0を、負の数なら1を入れる」 ← 2進負数の説明が必要

[考えてみよう] 教科書 p 98 の囲み記事

2進数で左へ1桁ずらす(シフトする)ということは、数を () 倍するということ。

← (証明は自分で考えておく)

上の例では、左へ2桁ずらす(シフトする)のだから、数を () 倍するということ。上の結果で確認せよ

4. 演算子の優先順位 (優先度)

- 基本は、数学の式での優先順位と同じ考え
 $2 + 3 \times 5 = 2 + (3 \times 5)$ とみなす。 $(2 + 3) \times 5$ ではない。
- 表4-3を確認しておこう。(覚える必要は無い)
- 結合規則の「右」と「左」について: 教科書 p 93~94の説明を参照

5. 型変換

- 型変換(特に自動変換)は、プログラミング言語によっても違うし、うっとうしい話題ではあります。
- ① (式の) 値の型とは違う変数(入れ物)に代入したり、違う型間で演算したいことがある。どうするか?
- ② 型が合わなければ原則はエラーになる。
- ③ 意識して変換する場合 vs 自動的に変換してくれる場合 がある。

自動的に変換してくれる場合、(たいていは思った通りに変換されるが) 場合によって思っていない変換をする。

⇒ (山内のお勧めは、自動変換に頼らず、自分で丁寧に型を変換するのが確実)

- 例

- ① 小さいサイズ→大きいサイズに代入 (int型→double型) ⇒ 自動で(何も書かなくても)型変換できる
大きいサイズ→小さいサイズに代入 (double型→int型) ⇒ (コンパイル時の)エラーになる

教科書 Saaple8.java inumはint型の変数であるが、その内容をdouble型の変数dnumに代入する
このとき、整数160が浮動小数160.0に変換される

教科書 p 47の表3-1を見直してみよう。 ⇒ [演習] (時間があれば) JCPad上で試してみよう

int型 → short型は () int型 → long型は () int型 → byte型は
()

int型 → double型は () int型 → float型は ()

double型 → float型は () float型 → double型は ()

double型 → int型は () float型 → int型は ()

- ② キャスト演算子による明示的な型変換 教科書 p 98 `int inum = (int)dnum;`
- ③ 異なる型同士の(2項)演算 教科書 p 100 (自動的に) 大きなサイズの型に統一(変換)して計算する
- ④ 教科書 p 102 の場合 演算の時は同じint型同士、代入の時にdouble型に変換するから、こうなる。

6. まとめ

教科書 p 105を見直す

第4回 小テスト (今日の復習)

今回は教科書p 106~107の練習問題1、2、4をやってみます。 回答欄を用意するので、ここに書き込んでください。

1. ① 小さなサイズの型の数値を大きなサイズの型に変換するには、
()
- ② 大きなサイズの型の数値を小さなサイズの型に変換するには、
()
- ③ 異なる型の数値どうしの演算は、()
- もし「できるわけではない」と答えたら、どういう場合に格納できないのか説明してください。

2. 次の計算をするコードを書いてください。 (Sample4などを参照)

- ② 3.14×2 (6.28になるように) ③ $5 \div 3$ (1.666...7になるように)

- ④ $30 \div 7$ の余りの数 (2になるように) ⑤ $(7+32) \div 5$

3. 三角形の高さと底辺を整数で入力し、面積 = (底辺×高さ)÷2 を出力するコードを書いてください。

(入力のあたりはSample3を参照)

第4回 復習の課題

第3回で学んだことを整理しておきましょう。以下の問に答える形で、(簡潔に)整理してみてください。

1. 変数とは何か、どういうものか、どうするものか? (漠然とした問い方なので、答えにくいですが、自分が理解したことを整理して書いてみましょう。

2. 変数の名前付けのルールはどんなものか? (いくつかルールがありましたが、列挙してみましょう。)

3. a) (変数の)型とは何か? (何をするものか? どう使うのか?)
b) 具体的にどういう型があってそれぞれの意味は何か? (授業で出てきた型を書いてみてください。)

4. a) 「変数に代入する」とはどういうことか? (代入するとどういう効果があるのか?)
b) 変数 x に、値 5 を代入したいときにはどうするか?

5. a) 「変数の値を参照する」とはどういうことか? (うまく表現できますか? 書くのが難しいかもしれない)
b) 変数 y の値を参照するときにはどうすればよいか?

6. キーボードから文字列・数値を読み込むにはどのようなプログラムを書けばよいか? 具体的に書いてみよ。