情報処理技法
(Javaプログラミング)1

第3回
コンピュータが情報を扱うには?
(変数, データ型, 代入)(続き)

人間科学科コミュニケーション専攻
白銀 純子

第3回の内容

- プログラムで扱うデータのおはなし(続き)

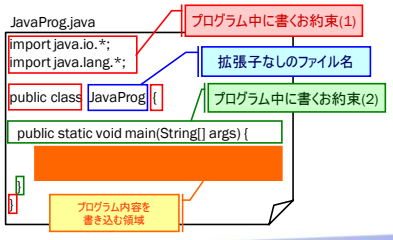
前々回の復習問題の解答

- プログラムの記述・コンパイル・実行は、それぞれどのようなソフトウェアを使って行うか、また、コマンドを使う場合にはどのようなコマンドを使うかを解答しなさい。
  - 授業で説明したソフトウェアでなくても、記述・コンパイル・実行ができるソフトウェアであればかまいません。

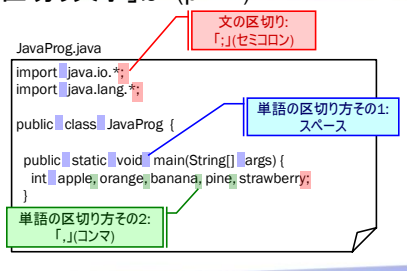
解答例
記述には、JeditやEmacsなどのテキストエディタを使う。コンパイルは、
javac Javaファイル名(拡張子つき)
というコマンドで行い、実行は
java Javaファイル名(拡張子なし)
で行う

前回の復習

プログラムの「カタチ」は?(p. 38)



Javaの「区切り文字」は?(p. 44)



プログラミングでのエラー(p. 49)

- プログラム作成時に、エラーでうまくいかないことも多い
  - コンパイル時に表示されるエラー: コンパイルエラー
    - スペルミスをした
    - カッコを開き忘れ・閉じ忘れた
    - 必要な場所に必要な命令を書いていなかった, etc.

    プログラム中の文法間違い、という意味のエラー
  - コンパイル後、実行時のエラー: 例外
    - 数を0で割ろうとした
    - 使ってはならない番号を使おうとした(配列など), etc.

    プログラムに文法間違いはないが、何らかのミスでそれ以上実行できない、という意味のエラー

コンパイルエラーの基本形(p. 49)

基本的なコンパイルエラーのメッセージの形

XXX.java:*n*:メッセージ
プログラム中の文
^

XXX.java: コンパイルしたファイル名
n: エラーが見つかった行数(「n行目にエラーがある」という意味)
^:「プログラム中の文」の中のあやしい部分(間違っていそうな部分)

コンパイルエラーへの対処の基本(p. 49)

- コンパイルエラーには一番上から順に対処すること
  - コンパイルエラーがたくさん出てきたときは、多くの場合、上の方に出ているメッセージがより適切な意味
  - 1つのまちがいが影響していろいろな部分のメッセージを出すことも
    - 例えば、宣言していない変数を5箇所で使っていたら、5つエラーメッセージが出てくる
- 「メッセージ」の部分をよく読み、エラーの意味を理解すること
- Jeditで、エラーが出た行番号のところをよく見て、ミスを探すこと

プログラムで扱うもの~データ型~(p. 56)

- 整数 → プログラムでの表現: int
- 小数 → プログラムでの表現: float または double
- 文字 → プログラムでの表現: char

プログラムでのデータは、どの系統になるか決めておく必要あり

「データ型」と呼ぶ

データの「種類」は?(p. 57)

- 1つ1つのデータにそれぞれ名前をつける

例えば...

「変数」と呼ぶ

購入するりんごの数(int): apple
量った牛肉の分量(float): meat
自分が働いている陳列棚のエリア(char): area

系統(データ型): int
種類(変数名): apple

系統(データ型): float
種類(変数名): meat

系統(データ型): char
種類(変数名): area

「変数」= データを入れるための箱

データは原則として、必ず箱の中に入れて扱う

変数の宣言(p. 59)

- 変数を使う(データを入れるなど)前に、変数を準備する必要

変数を「宣言する」という
= それぞれの箱が、「肉・魚系統」か「野菜系統」か 「飲み物系統」かをコンピュータに知らせ、箱を準備する

スペース

準備(宣言)例

```
int apple, orange, banana;
float meat, chicken;
char area, register;
```

変数の系統(データ型)を先頭に書く

「,」で区切って複数の変数を予告(宣言)できる

変数の値(p. 62)

- 変数(箱)に値(データ)を入れて扱う
- 「=」で値を決める ―「代入する」という = 箱の中に具体的なデータを入れること
  ※「=」の左側は、必ず変数1つだけ
  - 用意した変数に初めて値を入れること: 初期化

購入したりんごの数が10個だった場合

apple = 10; → 10

牛肉の分量を量ったら200.5gだった場合

meat = 200.5; → 200.5

数の計算(p. 66)

- 変数(箱)の中には、計算結果や処理結果を入れることもできる
  - 足し算: +
  - 引き算: -
  - かけ算: *
  - 割り算(商): /
  - 割り算(余り): %

例えば…代金計算
(支払い金額: result)

100円のりんごを10個買った場合

apple = 10;
result = apple * 100;

「result」には、「1000」という結果が入る

100円のりんごを10個,
150円のバナナを5個買った場合

apple = 10;
banana = 5;
result = apple * 100 + banana * 150;

「result」には、「1750」という結果が入る

変数の使いまわし

- 変数は使いまわし可能
  - 同じ変数の宣言は1度だけで良く、何度も宣言する必要はなし
  - データ型の変更は不可

データ型: int
変数名: apple

- 宣言直後: りんごの値段として「100」円を代入
- しばらく後: りんごの値段として「80」円を代入
- さらに後: りんごの「個数」として利用

ただし…すでに値が入っている変数に別の値を代入すると…

100 → 80

データ型: int
変数名: apple

もともと入っていたデータは消える

代入の不思議(p. 67)

「milk」を、店にある牛乳の在庫のパック数と考えると…
トラックが来る前: 在庫のパック数は30

milk = 30;

トラックが牛乳を50パック運んできた
この後の店の在庫数の計算は?

milk = milk + 50;

- milk = milk + 50; の左辺: トラックが来た後の在庫数
- milk = milk + 50; の右辺: トラックが来る前の在庫数

- 「=」より後の変数は、直前までに代入されていた値
- 「=」より前の変数には、「=」より後の計算結果を代入
  (値が新しいものに更新される)

変数宣言の注意(p. 61)

- 同じ名前の変数は、1回しか宣言できない

```
public static void main(String[] args){
  int abc;
  ……
  int abc = 10;
}
```

コンパイルエラーが出る(一度宣言した変数は何回でも使えるので、「int abc = 10;」の「int」は不要)

- 変数には、宣言と同時に値を代入してよい

```
int abc;
abc = 10;
```

```
int abc = 10;
```

同じ意味を表す

変数の宣言忘れに注意(p. 61)

- どの変数であっても、宣言していなければ使えない

```
int result;
result = banana + 10;
```

変数「banana」の宣言をしないまま、「banana」のデータを使って計算しようとしている

「シンボルを処理解釈できません」というエラーメッセージ
宣言していない変数はすぐ後に書かれているので、よくメッセージを読んで宣言をすること
※スペルミスの可能性もあるので、要注意

代入と参照の注意(p. 67)

- 初期化をしないと、変数を参照できないので注意
  - 箱の中にデータが入っていないので、存在しないデータを使って計算などはできないため

```
int  banana, result;
result = banana + 10;
```

変数「banana」の初期化をしないまま、「banana」の中からデータを取り出して計算しようとしている

「変数bananaは初期化されていない可能性があります」というエラーメッセージ

初期化が必要な変数(この変数を初期化すること)
※どのような値を代入すれば良いかはそのときどきでよく考えること

プログラムの記述とコンパイル・実行

- プログラムの内容
  - JeditまたはEmacsでなどのエディタ記述し、ファイルとして保存
- コンパイル・実行
  - ターミナルで、保存したファイルを指定

エディタとターミナルは、どちらをどのように使うか、きちんと区別しよう!

コンパイル・実行時の注意(1)

- ターミナルでのカレントフォルダを、Javaファイルを保存しているフォルダに設定すること
  - カレントフォルダ: ターミナルでの、現在の作業フォルダ
  - ターミナルを起動したとき: カレントフォルダはホームフォルダ

コンパイル・実行時の注意(2)

- カレントフォルダの変更のコマンド:
  - コマンドの入力
    % cd ***ホームフォルダからの相対パス***
  - Ex1. ホームフォルダの中で、「Desktop」→「Java」→「chap」に保存してある場合(相対パス: Desktop/Java/chap):
    % cd ***Desktop/Java/chap***
  - Ex2. ホームフォルダの中で、「Download」→「chap」→「chap01」に保存してある場合(相対パス: Download/chap/cha01):
    % cd ***Download/chap/cha01***

やってみよう!(2)

- 教科書p. 76の例題01-07をやってみよう