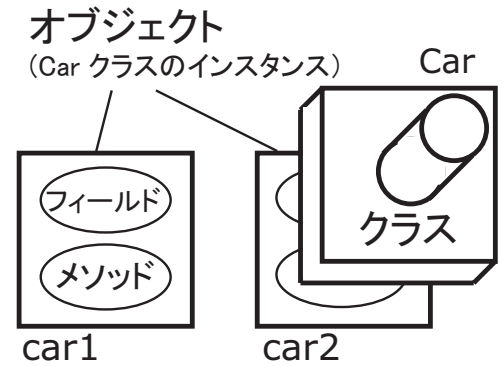


Java によるプログラミング入門 (8)

この回では、本題の Lesson6 の他に、Lesson5 の課題の解答例、Java の編集環境の紹介、今後の勉強の進め方についても簡単に触れます。

1 Lesson 6: クラスをまとめる

先の例ではオブジェクトごとにクラスを作りました。その中で TSTank1, TSTank2, TSTank3 は大変よく似た記述となっています。java ではオブジェクトを生成するために、まずクラスを記述し、そこからコンストラクタでインスタンスを生成します。このような 2 重構造になっているのは、「よく似たオブジェクトについての記述をまとめたい」からです。それにより、プログラムの負荷を減らすとともに、プログラムが分かりやすく柔軟になるからです。



○ インスタンス生成 (1) 後から変更

```
Car car1 = new Car();  
Car car2 = new Car();  
car1.setColor(RED);  
car2.setColor(BLUE);  
car2.changeEngine(JET);
```

○ インスタンス生成 (2) 生成時に

```
Car car1 = new Car(RED,DEFAULT);  
Car car2 = new Car(BLUE,JET);
```

図 1

クラス : は個々のオブジェクトを作るための「型」です .

インスタンス : は「型」であるクラスから実際に作られる個々のオブジェクトです . 一つのクラスからいくつでもオブジェクトを生成できます .

インスタンスの個別性 : 個々のインスタンスに個別性を持たせるには , フィールドの値をインスタンス毎に設定してやればいいのです .

タンクの例では , 各タンクの個別性は底面積 , 排出係数および初期水位です . 個々のインスタンスごとにこれらを設定できるようにすれば , タンクのクラスを一つにまとめてしまうことができます . 以下 , タンクとシステムのクラスをこの方針で書き換えてみましょう . 以下の実装例ではタンクの共通クラス `TSTank` のコンストラクタで個別タンクの特徴である底面積 , 初期水位 , 排出係数を設定するようにしています .

TSTank.java — タンク共通のクラス

```
1: public class TSTank {
2:     private double tankArea; //m**2
3:     private double initialTankLevel; //m
4:     private double drainageCoefficient; //m**3/s.m
5:     private double storedVolume;
6:     private double inFlow;
7:     private TSClock tsClock;
8:
9:     public TSTank(TSClock tsClock, double tankArea, double initialTankLevel,
10:        double drainageCoefficient) {
11:         this.tsClock = tsClock;
12:         this.initialTankLevel = initialTankLevel;
13:         this.tankArea = tankArea;
14:         this.drainageCoefficient = drainageCoefficient;
15:         storedVolume = initialTankLevel * tankArea;
16:     }
17:
18:     public double getTankLevel() {
19:         return storedVolume/tankArea;
20:     }
21:
22:     public void setInFlow(double inFlow) {
23:         this.inFlow = inFlow;
24:     }
25:
```

```
26: public double getInFlow() {
27:     return inFlow;
28: }
29:
30: public double getDrainageRate() {
31:     return drainageCoefficient*getTankLevel();
32: }
33:
34: public void update() {
35:     storedVolume += tsClock.getTimeStep()*(inFlow - getDrainageRate());
36:     if (storedVolume < 0) {
37:         storedVolume = 0.0;
38:     }
39: }
40:
41: public static void main(String args[]) {
42:     TSClock tsClock = new TSClock();
43:     tsClock.setTimeStep(0.1);
44:     TSTank tsTank = new TSTank(tsClock,10,1,0.5);
45:     for (int i=0;i<100;i++) {
46:         tsTank.setInFlow(1.0);
47:         System.out.println(tsClock.getTime() + "," + tsTank.getTankLevel());
48:         tsClock.update();
49:         tsTank.update();
50:     }
51: }
52: }
```

TSTankSystem2.java — 共通タンクを使うシステム

```
1: public class TSTankSystem2 {
2:     public static void main(String args[]) {
3:         TSClock tsClock = new TSClock();
4:         tsClock.setTimeStep(0.1);
5:         TSWaterSupply tsWaterSupply = new TSWaterSupply();
6:         TSTank tsTanks[] = new TSTank[3];
7:         tsTanks[0] = new TSTank(tsClock,10.0,1.0,0.5);
8:         tsTanks[1] = new TSTank(tsClock,5.0,3.0,0.3);
9:         tsTanks[2] = new TSTank(tsClock,4.0,1.0,0.6);
10:        System.out.println("time,tank level 1, tank level 2, tank level3");
11:        for (int i=0; i<300; i++) {
12:            for (int j=0;j<tsTanks.length;j++) {
13:                if(j==0) {
14:                    tsTanks[j].setInFlow(tsWaterSupply.getFlowRate());
15:                } else {
16:                    tsTanks[j].setInFlow(tsTanks[j-1].getDrainageRate());
17:                }
18:            }
19:            System.out.println(tsClock.getTime() + "," + tsTanks[0].getTankLevel()
20:                + "," + tsTanks[1].getTankLevel() + "," + tsTanks[2].getTankLevel());
21:            tsClock.update();
22:            for (int j=0;j<tsTanks.length;j++) {
23:                tsTanks[j].update();
24:            }
25:        }
26:    }
27: }
```

このプログラムについては少し解説します：

- 3つのタンクを Tank クラスの配列により扱っています．クラスの配列の利用は準備のステップが少し多くなります．

1. クラスの配列 (tsTanks) の宣言．¹

2. 配列領域の確保．ここまでは int 型や double 型と同じ．

3. 配列要素の各クラスのコンストラクタによる生成と代入²．

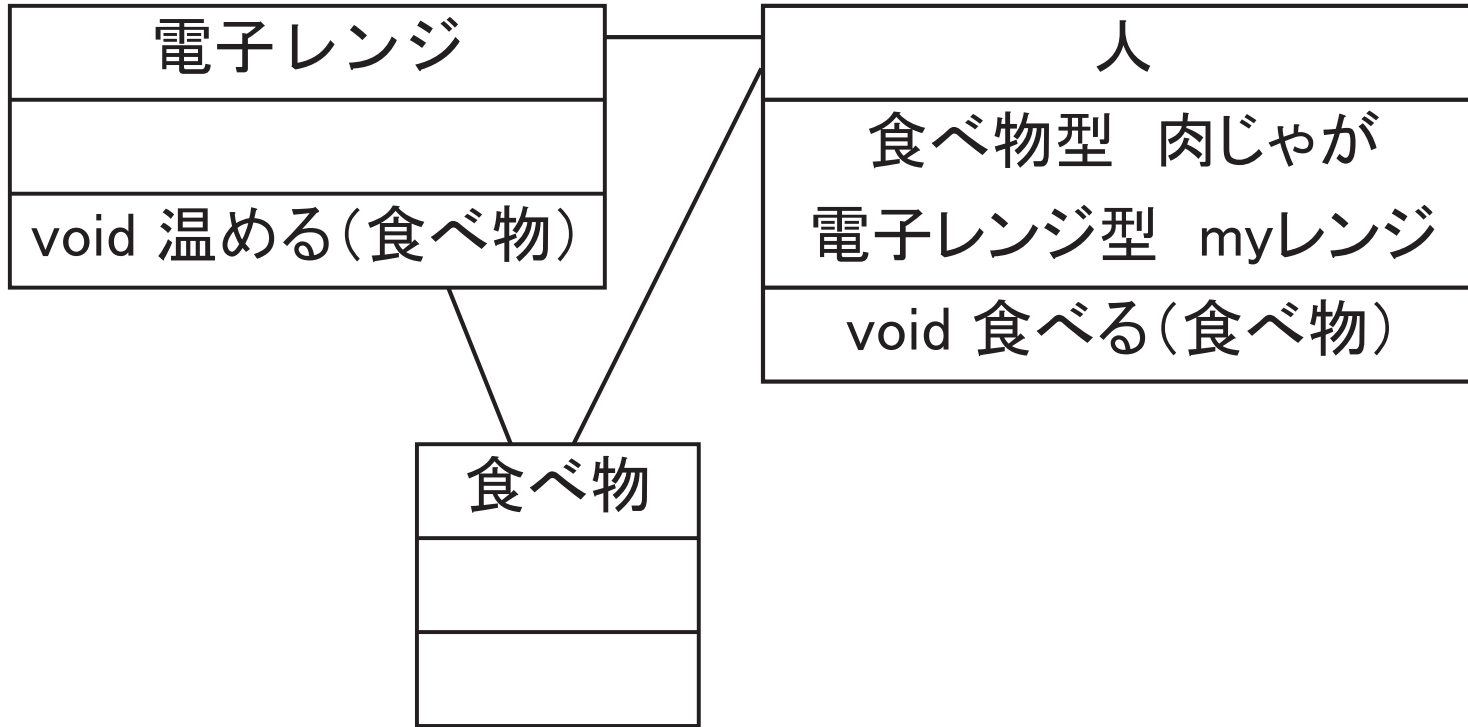
プログラム自身は機能を変えずに設計を見直ただけです³．したがって全く同じ結果が得られなければなりません．同じ結果が得られたかどうかを実行結果を比較して確認しなさい．

¹ 複数のタンクなので tanks と複数形にしています．

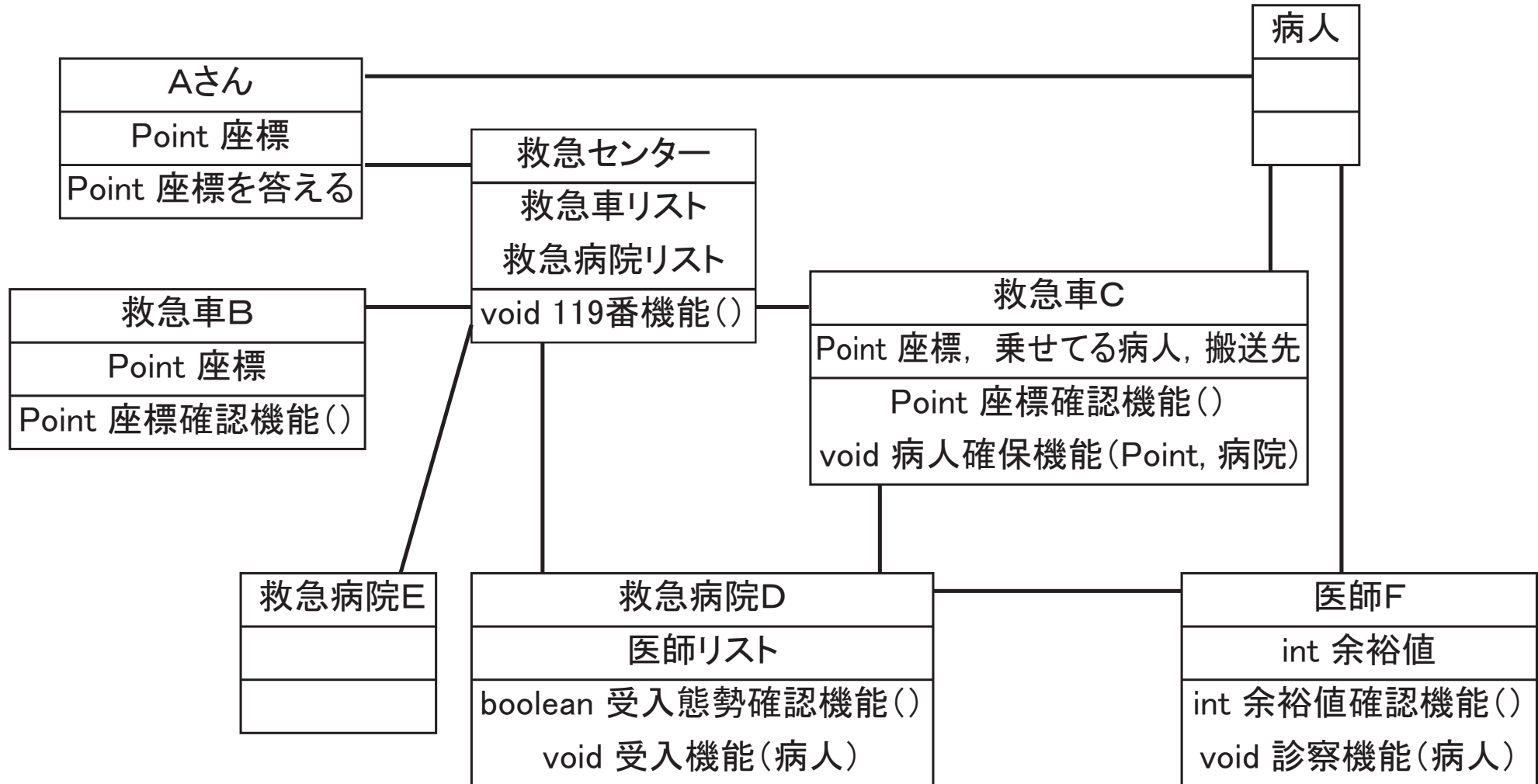
² オブジェクトの配列確保は単に要素となるオブジェクトの参照の入れ物の確保ですから個々のオブジェクトを生成して代入しなければなりません．

³ この種の改善のことを「リファクタリング」と呼びます．

電子レンジの課題のできあがりの例です。



病院の課題のできあがりの例です。



2 さらに学習のために

Java でのプログラミングをさらに学習してゆく方向について示します。

- 多様なデータの処理
 - 文字列の処理 (String クラスなどの取り扱い)
 - コレクション⁴
- 入出力：これまでに学んだ入出力はあまりに貧弱でパッとしません。以下のことを学んでゆくと良いでしょう。グラフィックユーザインターフェイスを学ぶ際には人間が行う様々な操作 (イベント) に対してプログラムがどのように応答するのかという書き方をします⁵。
 - コマンドラインパラメータの処理
 - ネットワークの利用
 - グラフィックユーザインターフェイス (Swing)
- プログラムの制御の流れの並行実行 (スレッド機構)
- Java のフレームワーク (Swing によるアプリケーションや Applet, Servlet, JSP などの Web プログラミング)
- データベースとの連携
- 大きく複雑なプログラムを設計し実装するための技法
 - リファクタリングやデザインパターン
 - UML の活用
 - 各種プログラミングツール (統合開発環境, テスト環境など) の活用

⁴ 配列より柔軟なデータの取り扱いのための API が Java には用意されています。

⁵ イベントドリブン型のプログラミング

3 CPad for Java2 SDK を使う

前節のようにエディタを使ってプログラムを書き，コマンドプロンプトを使ってコンパイラやバーチャルマシンを起動する方法もありますが，統合的な開発環境を使って編集，コンパイル，実行を一貫して行う方法もあります．ここでは初心者にも使いやすいものとして CPad for Java2 SDK というソフトを利用する例を示します⁶．

CPad for Java2 SDK のインストール まずは CPad for Java2 SDK をインストールします．

【家・研究室のPCにインストールする場合】CPad for Java2 SDK 以下の URL にあります．

<http://www.vector.co.jp/soft/win95/prog/se153698.html>

インターネットエクスプローラなどの Web ブラウザを使ってダウンロードしてください．ダウンロードしたファイル (jcpad231.lzh) はダブルクリックすればデスクトップ上で解凍されます．できた jcpad231 というフォルダをマイドキュメントにコピーしてください．その中に jcpad.exe ほかいくつかのファイルがあれば ok です．

【教育用端末で使う場合】C:¥jcpad231 というフォルダをMドライブにまるごとコピーして使ってください．

⁶ 最近ではより本格的な統合環境として Eclipse (<http://www.eclipse.org/>がよく使われます)

【実行と設定】

中の jcpad231.exe をダブルクリックして起動してください。起動時に「設定」画面が現れたら「コンパイラのパス」として

C: ¥ j2sdk1.4.2_12¥ bin¥ javac.exe

また、標準のデータ保存フォルダとして

M: ¥ ¥ Java

と入力してください。入力後「ok」ボタンを押して準備は完了です。

プログラムの作成、編集、実行 では CPad for Java2 SDK を使ってみましょう。

CPad for Java2 SDK のメニューから [ファイル] [新規作成] と選び、Sample.java のコードを入力し、[メニュー] [名前を付けて保存] を選んで、フォルダ java の中に Sample.java として保存します。

次に Sample.java について [実行] [コンパイル] を選ぶとコンパイルが実行されます。画面の下側がコンパイラのメッセージです。今度は [実行] [実行] と選ぶとコンパイルしたプログラムが実行されます。コマンドプロンプトウィンドウが生成され、実行結果が表示されます。実行結果を確認した後、どれかキーを押せばウィンドウが閉じます。

実行結果をリダイレクトするにはメニューから [実行] [実行時パラメータ] と選んで出される設定画面のうち、実行時パラメータに > result.txt と設定したのち実行します。