付録 B　総称クラス

B.1　総称クラスの使用

総称クラス（generic class）は、型パラメータを持つクラスのことで、JDK5.0 から導入された。代表的な総称クラスの例として ArrayList, HashMap, LinkedList などがあげられる。型パラメータは<と>の間に書かれる。

ArrayList はサイズの変更が可能な配列である。ArrayList の型パラメータは要素の型を表す。（総称クラスはこのようにコレクション（データの集まり）の型に使われることが多い。）例えば、String 型を要素とする ArrayList はArrayList<String>となり、次のように使用する。

```
ArrayList<String> arr1 = new ArrayList<String>();    // 空の ArrayList 作成
arr1.add("aaa");  arr1.add("bbb");  arr1.add("ccc"); // データ追加
String s = arr1.get(1);                              // データ取出し
```

add メソッドでデータを追加し、get メソッドでデータを取り出すことができる。

int, double のようなプリミティブ型は総称クラスの型パラメータになることができないという制限があるので注意が必要である。このときは Integer, Double などの対応するラッパークラスと呼ばれるクラスを利用する。ラッパークラスとプリミティブ型の変換はほとんどの場合、自動的に行われる（オートボクシング）ので、int の代りに Integer と書く以外は通常のクラス型をパラメータとするときと変わらない。例えば次のように書くことができる。

```
ArrayList<Integer> arr2 = new ArrayList<Integer>();  // 空の ArrayList 作成
arr2.add(123);  arr2.add(456);  arr2.add(789);       // データ追加
int i = arr2.get(1);                                 // データ取出し
```

ArrayList<String>に int 型の要素を add したり、ArrayList<Integer>から String 型の要素を get したりするのは、当然型エラー（コンパイル時のエラー）になる。

```
ArrayList<String> arr1 = new ArrayList<String> ();
arr1.add(333);          // 型エラー

ArrayList<Integer> arr2 = new ArrayList<Integer> ();
...
String t = arr2.get(2)  // 型エラー
```

このような型エラーをコンパイル時にちゃんと発見したい、というのが、総称クラスの導入のそもそもの動機である。

API ドキュメントの中では、型パラメータは E のような仮のクラス名が使われ、

java.util.ArrayList<E>クラス:

```
public boolean add(E e)
```

リストの最後に、指定された要素（e）を追加する。

```
public E get(int index)
```

リスト内の指定された位置（index）にある要素を返す。

のように書かれる。

例題 B.1.1 *ArrayList* クラス

ファイルから読み込んだデータの保存に ArrayList を使用する例である。init メソッドでファイルから読み込んだデータを保存し、doGet メソッドでそれを利用している。なお、配列型も総称クラスの型パラメータとして問題なく使用することができる。この例の場合、ファイルの行数が前もってわからないので、配列ではなく ArrayList を使用している。

また、この例では doGet メソッドの中で、拡張 **for** 文（**for-each** 文）を使用している。

ファイル ArrayListTest.java（前半）

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.util.ArrayList;

import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ArrayListTest extends HttpServlet {
  private ArrayList<String[]> questions;

  @Override
  public void init(ServletConfig config) throws ServletException {
    questions = new ArrayList<String[]>();
    try {
      File f = new File(config.getServletContext().getRealPath("/quiz.txt"));
      BufferedReader in = new BufferedReader(
                              new InputStreamReader(
                                 new FileInputStream(f), "Windows-31J"));
      String line="";
      while((line=in.readLine())!=null) {
        line = line.trim();
        if(line.equals(""))
                continue;
        questions.add(line.split("¥¥s+"));
      }
      in.close();
    } catch (IOException e) {
    }

  }
```

ファイル ArrayListTest.java（後半）

```
  @Override
  public void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
    response.setContentType("text/html; charset=Windows-31J");
    PrintWriter out = response.getWriter();
    out.println("<html><head></head><body>");
    out.println("<table border='1'>");
    out.println("<tr><th>問</th><th>1</th><th>2</th><th>3</th><th>答</th></tr>");
    for (String[] line : questions) {
      out.printf("<tr><td>%s</td><td>%s</td><td>%s</td><td>%s</td><td>%s</td></tr>%n",
                 line[0], line[1], line[2], line[3], line[4]);
    }
    out.println("</table>");
    out.println("</body></html>");
    out.close();
  }
}
```

例題 B.1.2 *HashMap* クラス

HashMapは連想配列と呼ばれるデータ構造である。通常の配列と異なり、int型だけではなく、任意の型（String型など）をキー（添字）として、値を格納・検索することができる。HashMapの型パラメータは2つあり、1つめがキーの型、2つめが値の型である。下の例では、HashMap<String, Integer>、つまりキーがString型で値がInteger型の連想配列を用いている。値の格納にはputメソッド、検索にはgetメソッドを用いる。

java.util.HashMap<K,V>クラス:

```
  public V put(K key, V value)
```

指定された値（value）と指定されたキー（key）をこのマップに関連付ける

```
  public V get(Object key)
```

指定されたキー（key）がマップされている値を返す。

Object（java.lang.Object）クラスはJavaのすべてのクラスのスーパークラスとなる、クラス階層のルートクラスである。

ファイル HashMapTest.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.util.HashMap;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class HashMapTest extends HttpServlet {
  HashMap<String, Integer> colors;

  @Override
  public void init() throws ServletException {
    colors = new HashMap<String, Integer>();
    colors.put("鴇", 0xf7acbc); colors.put("赤", 0xed1941);
    colors.put("朱", 0xf26522); colors.put("桃", 0xf58f98);
    colors.put("緋", 0xaa2116); colors.put("肌", 0xfedcbd);
    colors.put("橙", 0xf47920); colors.put("褐", 0x843900);
    colors.put("黄", 0xffd400); colors.put("鶸", 0xcbc547);
    colors.put("鶯", 0x87943b); colors.put("緑", 0x45b97c);
    colors.put("鉄", 0x005344); colors.put("水", 0xafdfe4);
    colors.put("青", 0x009ad6); colors.put("藍", 0x145b7d);
    colors.put("紺", 0x003a6c); colors.put("菫", 0x6ff0aa);
    colors.put("藤", 0xafb4db); colors.put("紫", 0x8552a1);
    colors.put("白", 0xfffffb); colors.put("灰", 0x77787b);
    colors.put("黒", 0x130c0e);
  }

  @Override
  public void doPost(HttpServletRequest request, HttpServletResponse response)
           throws ServletException, IOException {
    request.setCharacterEncoding("Windows-31J");
    String cn = request.getParameter("colorName");
    Integer cv = colors.get(cn);

    response.setContentType("text/html; charset=Windows-31J");
    PrintWriter out = response.getWriter();
    out.println("<html><head></head><body>");

    if(cv!=null) {
      out.printf("%s 色は<span style='color:#%06x'>こんな色</span>です。", cn, cv);
    } else {
      out.printf("%s 色は見つかりません。", cn);
    }
    out.println("</body></html>");
    out.close();
  }
}
```

例題 B.1.3 *LinkedList* クラス

LinkedList は ArrayList と同じように要素を付け足していくことができるコレクションの型だが、先頭からも末尾からも要素を追加したり削除したりできる。

ファイル LinkedListTest.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.util.LinkedList;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class LinkedListTest extends HttpServlet {
  private int i=1;

  @Override
  public void doGet(HttpServletRequest request, HttpServletResponse response)
  throws ServletException, IOException {
    response.setContentType("text/html; charset=Windows-31J");
    try {
      // デバッグ用
      i = Integer.parseInt(request.getQueryString());
    } catch (Exception e) {}

    PrintWriter out = response.getWriter();
    out.println("<html><head></head><body>");
    LinkedList<Integer> xs = new LinkedList<Integer>();
    int j=i;
    while(j>0) {
      xs.addFirst(j%10);
      j/=10;
    }
    out.printf("あなたは ");
    for(int k : xs) {
      out.printf("<img src='Images/%d.png' alt='%d'>", k, k);
    }
    out.printf("番目の来訪者です。%n");
    out.printf("</body></html>");
    out.close();    // close を忘れない
    i++;
  }
}
```

キーワード :

総称クラス, ArrayList クラス, HashMap クラス, LinkedList クラス

メモ: