

通過テスト14 解答

14

コレクションと総称型

1. ① K ② G ③ D ④ L

2.

```
package e2;
import java.io.*;
import java.util.ArrayList;
import java.util.Collections;
public class Exec {
    public static void main(String[] args) {

        ArrayList<String> ls = new ArrayList<String>();
        try {
            BufferedReader in =
                new BufferedReader(new FileReader("book.txt"));
            String s;
            while((s=in.readLine())!=null){
                ls.add(s);
            }
            in.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
        //
        Collections.sort(ls);
        try {
            PrintWriter out =
                new PrintWriter(new BufferedWriter
                    (new FileWriter("book.txt")));
            for(int i=0; i<ls.size(); i++){
                out.println(ls.get(i));
            }
            out.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
        System.out.println("タイトル数="+ ls.size()+"件");
    }
}
```

【解説】 `ArrayList` はコレクションの中では最も良く利用されるクラスでオブジェクトを格納する配列の代わりに使用する。要素数を気にしなくてよい利点がある。問題のように `add(s)` として末尾に追加するほかに `add(i, s)` のように添え字を使って `i` 番目に挿入するという使い方もできる。配列よりも高機能である。413 頁の主なメソッドを参照しておくこと。

3.

```
package e3;
import lib.Input;
public class Exec {
    public static void main(String[] args) {
        Car[] cars = new Car[100];
        int n=0;
        String name;
        while((name=Input.getString())!=null){
            cars[n++] = new Car(name);

        }
        for(int i=0; i<n; i++){
            System.out.println(cars[i]);
        }
    }
}
```

【解説】 オブジェクトの配列も使用されないわけではない。 `new` で要素オブジェクトを作成して、配列要素に代入する点がポイントである。

《注》 `Input` クラスの使用について

`Input` クラスはサポートデータに含まれている。`lib` パッケージを作成してその中に置くと使える。著者の HP からダウンロードできる。`getString()` では、Enter キーだけをタイプすると `null` が入力される。

4.

```
package e4;
import java.util.HashMap;
import java.util.Map;
import lib.Input;
public class Exec {
    public static void main(String[] args) {
        Map<Integer, String> map = new HashMap<Integer, String>();
        map.put(110, "田中 宏");
        map.put(111, "前田 五郎");
        map.put(112, "鈴木 愛");
        map.put(113, "佐藤 恵子");
        map.put(114, "村田 美紀");
        map.put(115, "麻生 二郎");
        int n;
        while((n=Input.getInt("番号"))!=0){
            System.out.println(map.get(n));
        }
    }
}
```

【解説】 110 のような整数を Map などコレクションの要素として格納する場合、自動ボックス化・自動ボックス化解除の機能によりラッパークラス型のオブジェクトとして格納されることに注意する。コレクションの各クラスが扱えるのは、オブジェクトだけで基本データ型のデータは格納できない。java1.6 から自動ボックス化・自動ボックス化解除により、この問題のように変換を気にしなくてもよくなった。ただし、総称型として Map<Integer, String>と書くことを忘れないようにする。

5. C

【解説】 sh は基本データ型の short なので、Integer 型に自動型変換されない

6. B

【解説】 peek はキュー構造でリストの先頭の要素を取り出すメソッド。Queue インタフェースや Li ndedLi st にある。ただし、ArrayLi st は Queue を継承していないので D は誤り。正解は B。