10.8 練習問題の解答

練習 10-1 の解答

Main.java

```
1  import java.io.*;
2  import java.util.*;
3
4  public class Main {
5    public static void main(String[] args) throws Exception {
6      Reader fr = new FileReader("c:¥¥pref.properties");
7      Properties p = new Properties();
8      p.load(fr);
9      System.out.println(p.getProperty("aichi.capital") + ":" +
           p.getProperty("aichi.food"));
10     fr.close();
11   }
12 }
```

練習 10-2 の解答

Department、Employee の宣言に「implements java.io.Serializable」を追加した上で、以下のクラスを作成します。

Main.java

```
1  import java.io.*;
2
3  public class Main {
4    public static void main(String[] args) throws Exception {
5      Employee tanaka = new Employee();
6      tanaka.name = "田中一郎";
```



```
7      tanaka.age = 41;
8      Department soumubu = new Department();
9      soumubu.name="総務部";
10     soumubu.leader = tanaka;
11     FileOutputStream fos =
           new FileOutputStream("c:¥¥company.dat");
12     ObjectOutputStream oos = new ObjectOutputStream(fos);
13     oos.writeObject(soumubu);
14     oos.flush();
15     oos.close();
16   }
17 }
```

ファイナライザで後片付けしない

あらゆるクラスではObjectクラスが備えるfinalize()メソッドをオーバーライドすることができます。**ファイナライザ**(finalizer)と呼ばれるこのメソッドは、利用済みのオブジェクトがガベージコレクション(GC)で掃除される際に自動的に呼び出されます。そのため、ファイナライザに次のようなコードを記述すれば「ファイルの利用後、明示的にclose()を呼ぶ必要がなくなる」と思うかもしれません。

```
protected void finalize() throws Throwable {
  this.stream.close();
}
```

ですが、**ファイナライザをファイルの後片付けに使ってはいけません。**なぜならGCがいつ行われるかはわからないからです(13.6.4節参照)。GCが行われなければファイルはいつまでも開いたままになり、最悪の場合にはファイルが壊れる恐れがあります。

第11章で学ぶネットワークや第12章で学ぶデータベースも、利用後にはclose()を明示的に呼び出す必要があります。一見、便利で使いたくなるファイナライザですが、実際にはほとんど使う状況はない道具だと覚えておきましょう。

第11章

ネットワーク通信

オンラインゲームやSNSのような私たちが身近に使っているサービスはもちろん、企業で使用される業務システムのために作られるプログラムでも、ネットワーク通信は欠かせないものとなっています。

私たちが学んでいるJavaは、ネットワークの利用を前提としたさまざまなしくみが用意されている言語です。

この章でネットワークに関するAPIを学べば、Javaでいかに手軽にネットワーク通信を実現できるかが実感でき、プログラミングの幅も広がるに違いありません。