Ｊａｖａを用いた雨量データベースの作成

九州大学工学部　○学生員　本間　一義
九州大学工学部　正　員　森山　聡之
九州大学工学部　正　員　平野　宗夫

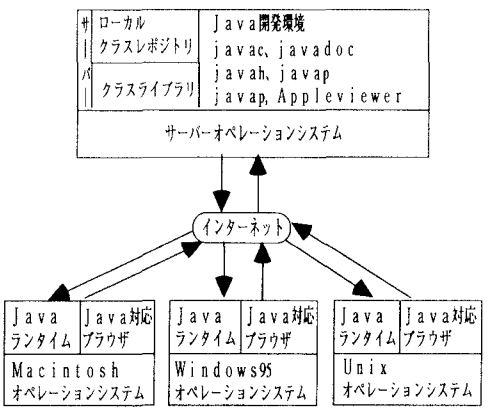
1．はじめに

近年、パソコンの普及とそれに伴なうインターネットの普及により、誰もが簡単に、様々な情報へアクセスしそこから必要な情報を入手することが可能になった。また建設省河川局でも水文情報の公開へ向けてその具体化が進められ、その一環としてレーダ雨量計の観測データをどのように公開するかについて検討がなされている。本研究は、インターネット上のWWWサーバーで標準的なプログラミング言語として普及してきているＪａｖａを用いて、レーダ雨量計の観測データを含む降水レーダデータをインターネット上にホームページとして公開できるようにデータベース化し、より簡単に降水レーダデータを入手及び使用可能にすることを最終的な目的とする。

本報では、その第一歩としてＪａｖａの可能性を探るためにＡＭＥＤＡＳデータを用いて、降雨の有無の判定を試みる。

2．Ｊａｖａ言語の特徴

Ｊａｖａは、1995年5月にサンマイクロシステムズ社より発表されて以来、急速に使用されているプログラミング言語である。Ｊａｖａの一番の利点はオペレーションシステムやコンピュータに依存しないということである。この関係を図－１に示す。Netscape Navigatorのようなｊａｖａのインタプリタを内蔵するブラウザを使用すればWindows・MacOS・Unix等のどんなオペレーションシステムでも、同じＪａｖａのアプレットを利用できる。これによりたとえば大型計算機で、ＦＯＲＴＲＡＮ等で書かれたプログラムをパソコンやワークステーションで使う場合に必要だった修正がいらなくなる。またブラウザはインターネット上のWWWサーバー上のＪａｖａソフトを自動的にダウンロードするので、いつでも最新バージョンのＪａｖａソフトを簡単に使用することが可能である。次にＪａｖａの機能の特徴については、まずＪａｖａはオブジェクト指向言語であり、これはサブルーチンとデータを一つのオブェクトというかたまりにしてモデル化し、そのオブジェクトの単位内で情報処理ができるようなメカニズムを持つことが挙げられる。このオブジェクトの継承機能を用いると、プログラムの変更がオリジナルのプログラムを変更することなく可能なため、プログラムの保守も容易になる。またＪａｖａには、グラフィク・ＧＵＩ・ネットワーク・ユーティリティ関係などのクラスライブラリが組み込まれており、この高度な機能を再利用することでプログラムの生産性を向上させることが可能である。Ｊａｖａは、ネットワーク環境で使われることを前提としているので安全性も十分考慮されており、ウイルスや、システムを破壊する恐れのあるコードからシステムを保護するためのセキュリティ機構が、いくつか実装されている。この結果、システムクラッシュ等が避けられるためデバッグが効率良く行えることも特徴として挙げられる。問題点としては、Ｊａｖａはインタプリタ言語であるため、ＦＯＲＴＲＡＮやCのようなコンパイラ言語ほど実行速度は速くなく、Ｃの1/20程度といわれている。そこで、送られてきたＪａｖａバイトコードを実行中のＣＰＵ用コードにリアルタイムに変換する『ジャスト・イン・タイム』コンパイラが作成されつつあり、この問題も近いうちに改善されると思われる。今回、Ｊａｖａの開発システムとしては、MacOS上でMetroWerks社のCodeWarrior10 Academic版を用いたが、他のOS上の他のＪａｖａ開発システム上でも同様の開発が可能である。



図－１．Ｊａｖａとインターネットの関係

3．使用データの概要

今回用いたデータは、九州北部40地点の1993年6月1日1時ら1993年９月24日24時の観測データである。図－２に

その一部を示す。1行目は、地点番号および観測開始と終了日時、2行目から時間雨量が24個すなわち1日分のデータが各行に記録されている。

4. 計算の結果と整理

今回は、WWWブラウザで動作するアプレットではなく、図－3に示すようなアプリケーションプログラムとして作成する。

図－2に示した様なフォーマットのAMEDASデータを用いて観測期間中の各時間毎の降雨判定が行えるようなプログラムを作る。以下に簡単にその過程を示す。

1. まず観測期間を時間単位で表わすベクトルTimeListをつくる。TimeListの長さは116（日）×24（時間）＝2784時間であり、すべての要素に初期値0を代入する。
2. AMEDASデータをファイルから一行読み取り、入した行ごとにプログラム中で雨量データかどうかは行数で判定する。図－2の1行目のように地点番号、観測期間を示す行は、そのまま出力する。
3. 2行目以下の様に雨量が並んでいる時は、その行にある数値を1つずつ判定する。図－2の3行目を例にとると1番目と2番目の数値は0であり、この時は無降雨であるのでプログラムは何もせず次の数値を読む。3番目に数値1がでてくるが数値が0でないとき時は、降雨が存在するのでプログラムはTimeListのcount＝27番目（24+3時間）の要素を0から1に書き換える。このようにしてプログラムはAMEDASデータファイルを上記2.3.に従い順に判定していく。
4. すべての判定が終わるとTimeListには、0か1が各要素にセットされているのでTimeListの各要素にその要素が示す日付と時刻を付け足して、TimeListを出力し、プログラムを終了する。

図－4に出力の一部を示す。これにより1993年8月27日には、13時と15時に降雨があったことが確認できる。

```
Fri Aug 27 11:00:00  1993:0
Fri Aug 27 12:00:00  1993:0
Fri Aug 27 13:00:00  1993:1
Fri Aug 27 14:00:00  1993:0
Fri Aug 27 15:00:00  1993:1
```

図－4. 出力結果

5. おわりに

今回、時間毎の降雨判定プログラムの動作が確認できたが、今後この結果をふまえ、引き続きＪａｖａの機能を活かした雨量データベース作成を行いたい。その際は、今回示したようなＦＯＲＴＲＡＮ的なプログラミングだけでなく、オブジェクト指向言語の特徴を生かしたプログラミングを行う予定である。今後も研究を進めインターネット上でデータベースプログラムとして公開できるように改良していきたい。

```
10 27 1993 6 1 1 1993 9 24 24
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 3 3 9 9 5 7 3 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 1 3 2 2 5 6 3 4 4 4 3 2 2
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

図－2. AMEDASデータ

```
import java.io.*;
import java.util.*;
import java.lang.*;

public class HANTEI {
  public static void main(String args[])   {
    Vector TimeList = new Vector();
    int Time = 2784;
    for(int i = 0 ; i < Time ; i++) {
      TimeList.addElement("0");
    }
    int count = 0;
    try {
     FileInputStream f = new FileInputStream("AMEDAS.P");
     DataInputStream in = new DataInputStream(f);
     FileOutputStream fout = new FileOutputStream("DATA");
     PrintStream out = new PrintStream(fout);
     String line;
     int lineno = 0;
     while((line = in.readLine()) != null) {
       if(lineno == 0){
         System.out.println(line);
       }
       else if(lineno > 0){
         StringTokenizer st = new StringTokenizer(line);
         while(st.hasMoreTokens()) {
           String str = st.nextToken();
           int x = Integer.parseInt(str);
           if(x != 0) {
             TimeList.setElementAt("1",count);
           }
           if(count >= (Time-1)) count = 0;
           else count++;
         }
       }
       if(lineno ==116) lineno = 0;
       else lineno++;
     }
     Object a[] = new Object[TimeList.size()];
     TimeList.copyInto(a);
     int j = 1, k = 0;
     for(int i = 0 ; i < a.length ;i++){
       Date d = new Date(93,5,j,k,00);
       if(k == 23){
         k = 0;
         j++;
       }
       else k++;
       out.println(d+":"+a[i]);
     }
     in.close();
     out.close();
    } catch(Exception e){
      System.out.println(e);
    }
  }
}
```

図－3プログラムリスト

＜参考文献＞

1）森山　聡之、平野　宗夫：「建設省レーダデータに関する認識について」水文・水資源学会要旨集（1996）

2）青柳　龍也（1996）：ＪａｖａＡＰＩプログラミングガイド，工学図書