CSCI1530 Computer Principles and Java Programming

Tutorial 13
Asg 6: Statistics on Array Data and File I/O

Zheng Qingqing
SHB 911
qqzheng@cse.cuhk.edu.hk

Task && Aim

- Task:
  - Read raw HSI trading data from URL;
  - Finding some statistics of Hang Seng Index (HSI) daily close.

- Aim:
  - Practice using looping constructs on an array;
  - Work with file I/O and exception handling

Samples run

- User input is highlight in **red**

First sample

```
Start Date: 20001215
HSI from 20001215 to 20150409
Number of records: 3572
Max: 31638.22
Min: 8409.01
Average: 17924.20284994405
Median: 19050.449999999997
```

Second sample

```
Start Date: 20130822
HSI from 20130822 to 20150409
Number of records: 403
Max: 26944.39
Min: 21182.16
Average: 23399.991488833755
Median: 23291.04
```

Skeleton

HSI.java

1. Read the raw HIS trading data from the URL;

2. Ask the user for a single input value (an 8-digit interger in YYYYMMDD format) ;

1. Process the raw data and do some statistics.
   1. Number of records between specific dates
   2. Average/Max/Min/Median of HIS

File I/O: read data from URL

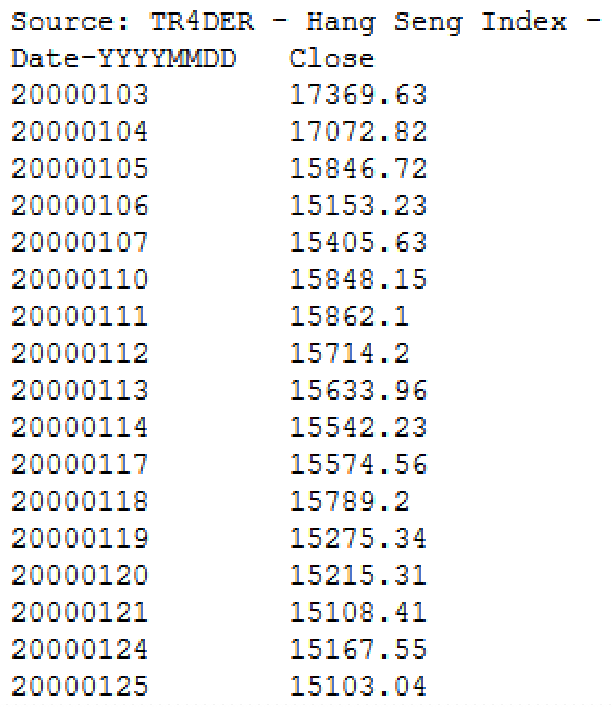
- Uniform Resource Locator (URL object) allows us to access to data from web pages:
  - http://www.cse.cuhk.edu.hk/csci1530/assignment/HSI.txt

- To read data from or write data to a file, we must create or obtain a Java stream object which attaches to the file.

- To read data from a text file, we use a `Scanner` object.

File I/O: read data from URL

- Text feature:
  - Fist two lines are not data
  - First column: ascending time stamps (in YYYYMMDD format)
  - Second column: HSI closes
- Extract data:
  - Skip the first two lines
  - Read “Date” into an integer array.
  - Read “Close” into a double array.
- Please note:
  - calendar dates are not consecutive ! The number of row counts but not the calendar days spanned.
  - At most 4000 records to process



```
Source: TR4DER - Hang Seng Index -
Date-YYYYMMDD   Close
20000103        17369.63
20000104        17072.82
20000105        15846.72
20000106        15153.23
20000107        15405.63
20000110        15848.15
20000111        15862.1
20000112        15714.2
20000113        15633.96
20000114        15542.23
20000117        15574.56
20000118        15789.2
20000119        15275.34
20000120        15215.31
20000121        15108.41
20000124        15167.55
20000125        15103.04
```

File I/O: read data from URL

```java
//declare a String variable to store the url
String addr
="http://www.cse.cuhk.edu.hk/csci1530/assignment/HSI.txt";
//New a URL object
URL link = new URL(addr);
//Associate the scanner object to stream object
Scanner dataStream = new Scanner( link.openStream() );
//Read a whole line
String line1 = dataStream.nextLine();
String line2 = dataStream.nextLine();
int[] day = new int[4000];  //declare & initialize an int array
double[] his = new double[4000];
if ( dataStream.hasNextInt() ) { //check for end-of-data
    int day2 = dataStream.nextInt();
    double hsi2 = dataStream.nextDouble();
}
```

User input as start Date

- Declare an integer to store “start Date”.
- Ask the user to input an 8-digit integer on console.

```
int startDate;
Scanner userInput = new Scanner(System.in);
int startDate = userInput.nextInt();
```

Data Process and Statistics

- Start Date && End Date:
  - Start Date is user specified
  - End Date is in the last line of data (Hint: when read the data from url, keep the last day as End Date)
- Number of Record:
  - number of record =

    (End Date index) – (Start Date index) +1
  - Hint: for loop to traversal the day array, if-else statement to search the index of Start Date/ End Date.

Data Process and Statistics

- Create a new array to store the HSI from Start Date to End Date.
  - Array length is number of record
  - Sample code

```
for (int k =0; k<num_record; k++){
    record[k] = hsi[k+startindex];
    sum = sum + record[k];
}
```

Data Process and Statistics

- Sort the record array in order to find Max/Min/Median
  - Select sort sample code(sort array N):

```
int N[] = {4, 5, 3, 1};
int round, pt, size = 4;
for (round = 0; round < size - 1; round++)
   for (pt = round + 1; pt < size; pt++)
      if (N[pt] < N[round]) {// a new min found
         int temp = N[pt];    // exchange their
         N[pt] = N[round];    // position
         N[round] = temp;
      }
```

Data Process and Statistics

- After sorting:
  - Max: the last element of the record array
  - Min: the first element of the record array
  - Average: $\frac{sum}{num\_record}$
  - Median:
    - Number of record is odd, median is the middle value;
    - Number of record is even, median is the mean of middle two value.
    - [Hint: (number of record) % 2 determine a number is odd or even (number of record) /2 to get the middle index if odd]

Exception Handling

- Terminate the program if any exceptions/ unexpected conditions occur.

```
public static void main(String[] args) throws Exception{
try {
      ...//Write your code here
   }catch (FileNotFoundException e) {
   System.out.println("File cannot be opened!");
   } catch (IOException e) {
   System.out.println("I/O error! Program exit.");
   }
}
```

Q & A