Lecture 22: Arrays and Images

Oct 30, 2019

Announcements

- Lab 7
- Nice job on Lab 7!
- Make sure you commit/push before deadline
- HW 6 will be posted soon

Last Time

- We talked about internetworking
- The Internet and IP
- IP addresses (32 bit) versus Ethernet addresses (48 bit)
- Role of DHCP and ARP
  - DHCP - obtain "personal" IP address, router IP address, DNS server IP address
  - ARP - convert IP address to Ethernet address on LAN

Today's Plan

- Begin discussing arrays
  - Motivation: Images - Arrays of Brightness
  - Introduction to 1-Dimensional Arrays
  - For loops - Iteration over Arrays
  - 2-Dimensional (and larger) Arrays
Lab 7: Digital Images

A discrete collection of brightness intensity values (ints)

Start Simple: One-Dimensional Arrays

- Arrays are collections of objects (ints, Strings, JButtons, etc)
- Declaring a 1-dimensional array
  - int [] grades;
  - String [] months;
  - JButton [] buttonList;
- The first element is grades[0];
- The number of elements is grades.length
- The final element is grades[grades.length - 1]

Initializing 1-Dimensional Arrays

- int [] grades = new int [73]; // all set to 0
- int [] grades = new int [] {85, 100, 92, 97, 88};
- String [] months = new String [12]; // all null
- String [] seasons = new String [] {“Fall”, “Winter”, “Spring”, “Summer”};

Initializing 1-Dimensional Arrays

- int [] squares = new int [20]; // all 0
- int i = 0;
  - while(i < squares.length) {
    - squares [i] = i*i;
    - i = i + 1;
  - }
Initializing 1-Dimensional Arrays

```java
JButton [] buttonList = new JButton[20]; //all null

int i = 0;
while(i < buttonList.length) {
    buttonList[i] = new JButton("Button #" + i);
    contentPane.add(buttonList[i]);
    i = i + 1;
}
```

Recap: Using 1-Dimensional Arrays

```java
public int sumOfSquares(int [] squares) {
    int sum = 0;
    int i = 0;
    while(i < squares.length) {
        sum = sum + squares[i];
        i = i + 1;
    }
    return sum;
}
```

Using 1-Dimensional Arrays

```java
public void buttonClicked(JButton theBut) {
    int i = 0;
    while(i < buttonList.length) {
        if (buttonList[i] == theBut) {
            // do some action specific to this button
        }
        i = i + 1;
    }
}
```

Example

- BadStats
- BabyStats
Example

SimpleStats

A Better Loop FOR Arrays

```java
int i = 0;
while (i < grades.length) {
    total = total + grades[i];
    i = i + 1;
}
```

Example

MoreStatsWithFor

```java
for (int i = 0; i < grades.length; i++) {
    total = total + grades[i];
}
```

A Better Loop FOR Arrays

```java
initialize_loop_variable;
while (condition) {
    loop_body_statements
    advance_loop_variable;
}
```

```java
for (initialize_loop_variable; condition; advance_loop_variable) {
    loop_body_statements
}
```
Next time:
2-Dimensional Arrays

- `int [] grades;` // an array of `ints`
- `String [] months;` // an array of `Strings`
- `JButton [] choices;` // an array of `JButtons`
- `int [][] pixels;` // an array of `int[]s`! (an array of arrays!)