Midterm Review (continued)

Bored?
An optional assignment is available on the class website. (Purely optional; not extra credit)

An assignment for next week will be available soon. Arrays, arrays, arrays (and loops).
  • Sudoku game board checker
  • Anagram solver (dictionary search)
  • A simple cipher

From Monday ...

Produce a number wedge
1
1 2
1 2 3
1 2 3 4
...
1 2 3 4 5 6 7 8 9 10

Number wedge: Nested loops

static void wedge() {
  // For each row of the wedge
  for (int width = 1; width <= 10; ++width) {
    // For each column of the wedge
    for (int i = 1; i <= width; ++i) {
      System.out.print(i + " ");
    }
    System.out.println();
  }
}
Word and line count

Count the lines and words in a file.
Output for this slide (including title) would be “9 lines, 48 words”

Assume you have a scanner f for the file, and remember you can use
f.hasNextLine(), f.nextLine(),
Scanner s = new Scanner( line ), s.next()

---

Word and Line Count

static void wordCount(Scanner f) {
  int lines = 0;
  int words = 0;
  while (f.hasNextLine()) {
    String line = f.nextLine();
    ++lines;
    Scanner l = new Scanner( line );
    while(l.hasNext()) {
      String w = l.next();
      ++words;
    }
  }
  System.out.println(lines + " lines, " + words + " words");
}

---

Sum of Maxima

/**
 * Compute sum of the maximum integers
 * on each input line.
 * @param f A scanner for a text file in which
 * each line has one or more positive integers.
 * @return The sum of the maximum integers
 * from each line.
 */

---

Sum of Maxima (notes)

You can use:
  f.hasNextLine()
  f.nextLine()
  s = new Scanner( l )
  s.hasNextInt()
  s.nextInt()

Sample input:
  5 2 7 3 4
  8 7 6
  4 2 9

Sample result:
  24
Sum of Maxima

static int sumMax(Scanner f) {
    int sum = 0;
    while (f.hasNextLine()) {
        String line = f.nextLine();
        Scanner l = new Scanner(line);
        int max = 0;
        while (l.hasNextInt()) {
            int n = l.nextInt();
            if (n > max) { max = n; }
        }
        sum += max;
    }
    return sum;
}

Control break logic

“Control break” = change from one set of like items (e.g., bananas) to another (oranges)

Very common in file scanning application
- Often with more than one level of grouping, e.g., departments within schools within cities

Harder than it looks
- Simple concept, but it takes careful design to get it right.

Summary counts (tricky!)

Given a file like this:
banaebra nano narana
apple apple orange orange
orange orange pear
pear pear

Print counts like this:
3 banana
2 apple
4 orange
3 pear
4 kinds of fruit
12 total fruit

You may assume that there is at least one fruit, and all like fruits appear together.

Fruit counter

static void fruitCount(Scanner f) {
    int totalFruitCount = 1;   // crowded to fit on slide
    int thisFruitCount = 1;   // crowded to fit on slide
    int fruitKinds = 1;
    String thisFruit = f.next();
    while (f.hasNextInt()) {
        String fruit = f.next();
        if (fruit.equals(thisFruit)) {
            ++thisFruitCount;
        } else { ++fruitKinds; // this is the “control break”
            System.out.println( thisFruitCount + " " + thisFruit );
            thisFruit = fruit;
            thisFruitCount = 1;
        }
    }
    System.out.println( thisFruitCount + " " + thisFruit);
    System.out.println( fruitKinds + " kinds of fruit");
    System.out.println( totalFruitCount + " total fruit");
}
**Things to be ready for ...**

Scope and initialization of variables  
Nested loops  
Nested “if” and “if / else if / else”  
Methods, written precisely to specification

45 minutes – strict!  
*(I will not stay to collect exams after 1:50)*

**Allowed and Disallowed**

You may bring your textbook, or another book  
You may bring written notes  
You may (and should) bring scratch paper  
and you may turn it in to show your work

The only computing device you may use is your brain. No computers, phones, calculators, music players, ...

No talking (except: raise your hand to ask questions)