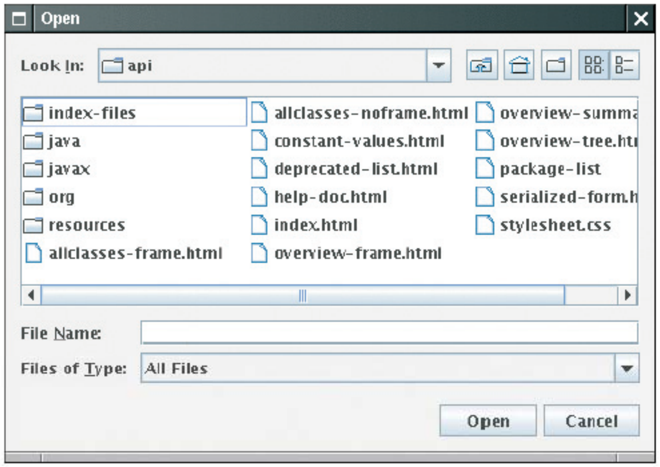
CompSci 6
Programming Design and Analysis



Look In: api

File Name:

Files of Type: All Files

A JFileChooser Dialog Box

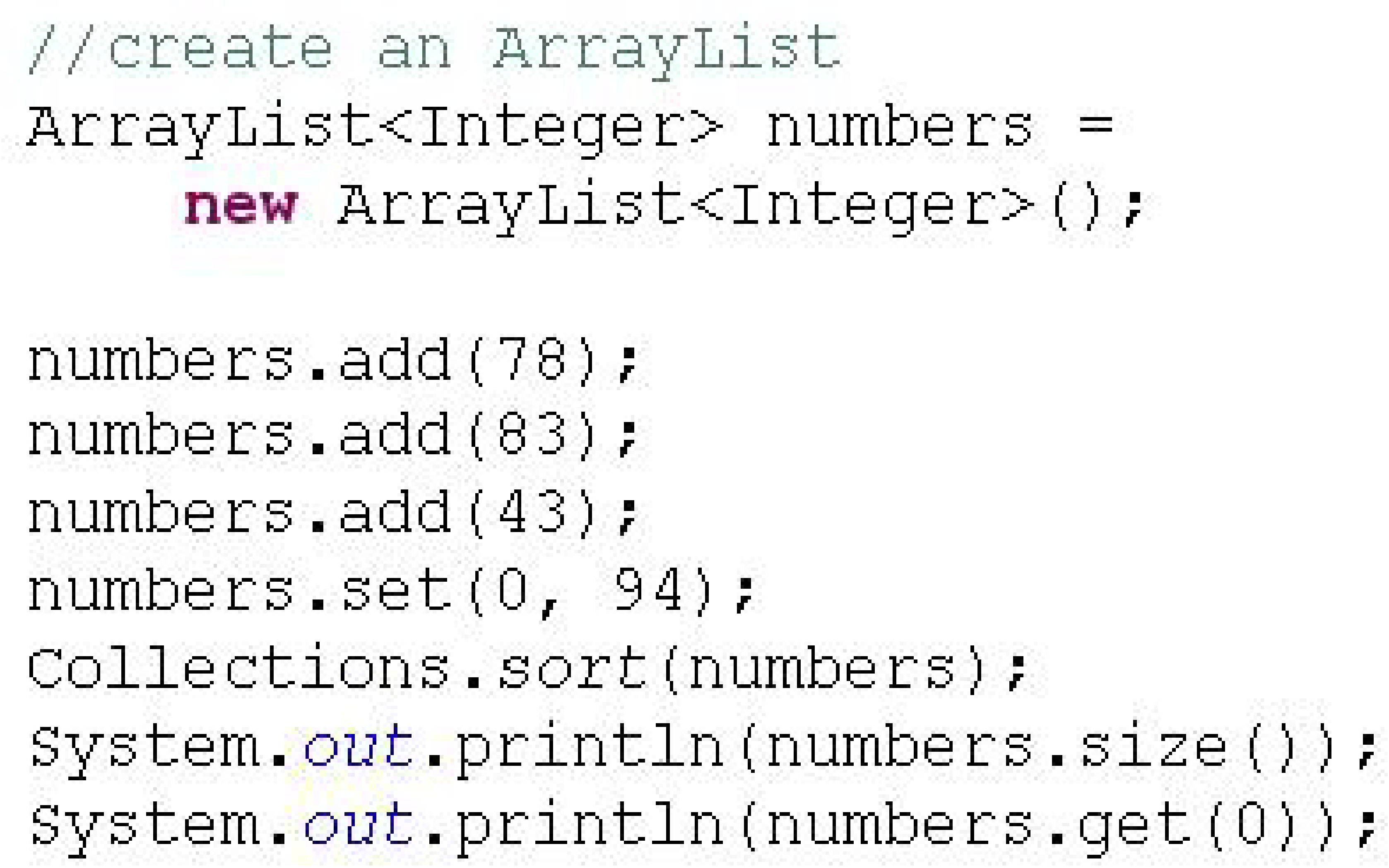
February 9, 2010

Prof. Rodger

Announcements/Review

- Assignment 4 due in a week
- Reading: Ch. 2.11-2.13, 8.1-8.8
  - Classes & OOP
- Reading Quiz next time
- Review: types of loops:
  - while, for, collection loop
- Review: Arrays:
  - array – built-in
  - ArrayList – Collection in Java

How do you use an ArrayList?
What does this code do?

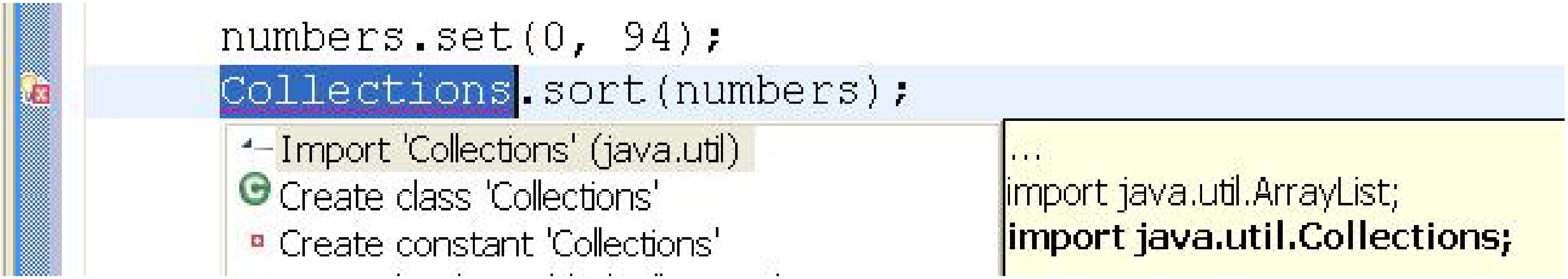


```
//create an ArrayList
ArrayList<Integer> numbers =
    new ArrayList<Integer>();

numbers.add(78);
numbers.add(83);
numbers.add(43);
numbers.set(0, 94);
Collections.sort(numbers);
System.out.println(numbers.size());
System.out.println(numbers.get(0));
```

To use other Classes

- Sometimes need to add an import

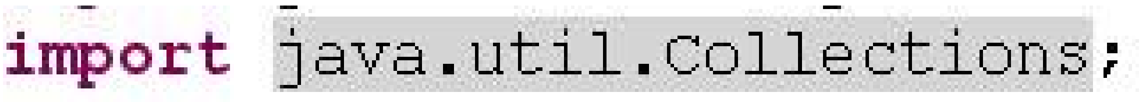


```
numbers.set(0, 94);
Collections.sort(numbers);
```

Import 'Collections' (java.util)
Create class 'Collections'
Create constant 'Collections'

```
...
import java.util.ArrayList;
import java.util.Collections;
```

- Appears at top of program



```
import java.util.Collections;
```

Strings

- String
  - a sequence of characters
  - *objects* of the String class
- String constants: `"Hello, World!"`
- All Strings are constants don’t use “new” w/ String
- String variables:

```
String message = "Hello, World!";
```

- String length:

```
int n = message.length();
```

- Empty string: `""`

- Concatenating Strings
  - Use the + operator:

```
String name = "Dave";
String message = "Hello, "
  + name;
```

- Automatic type conversion

```
String a = "Agent00";
int n = 7;
String bond = a + n;
// bond is "Agent007"
```

Strings (cont)

- How do you build a new string?
  - Initialize a string as empty
  - Use + (concatenation) to put strings together
  - Example:.

```
String dayFairStarts = “”;
String month = “Oct”;
dayFairStarts = “Friday”;
dayFairStarts += “, “ + month + “ “ + 14
  + “, “ + 2005;
```

  - Has the string been modified?

What can you do with strings?

- Look at API
- int length()
  - Returns length of string
- String substring(int beginIndex)
  - Returns substring from beginIndex to end of string
- String substring(int beginIndex, int endIndex)
  - Returns substring from beginIndex to endIndex -1

Example

```
String one = “ferriswheel”;
String two = one.substring(5);
String three =
  one.substring(4,6);
```

What are two and three?

Finding substrings in strings

- int indexOf(String str)
  - Returns first position of str in the string
  - First position in a string is 0
- int indexOf(String str, int fromIndex)
  - Returns first position of str starting at fromIndex

Example

```
String one = “Cotton Candy”;
String two =
  one.substring(indexOf(“Can”),
      indexOf(“Can”)+4);
```

What is two?

Review Strings

- `String word = “CompSci 6”;`
- `word.length()` – returns length of string
- `word.toCharArray()` – returns string as an array of characters
- `word.charAt(5)` – returns character at position 5
- Loop over characters in a string

```
for (char ch:
word.toCharArray())
   {
   }
```

Classwork Birthday

- Convert a String that represents a number to an int

Example (hint):

```
String numString = “87”;
   int num =
Integer.parseInt(numString);
   // num has int value 87
```

Comparing Strings and Objects

```
String one = "computer";
String two = "com" + one.substring(3,8);
String three = two;
String four = "science";

System.out.println(two);
if (one == two)                     // don’t do
        System.out.println("A");
if (two == three)                   // don’t do
        System.out.println("B");
```

Comparing Strings (cont)

```
if (one.equals(two))
        System.out.println("C");
if (two.equals(three))
        System.out.println("D");
if (one.compareTo(four) < 0)
        System.out.println(“E");
```

Reading from Files

- import java.io.File;
- Declare a file

```
File fileOfCats = new File(“cats.txt”);
```

- Use file – pass it as an argument to a Scanner

```
  Scanner in = new Scanner(fileOfCats);
```

Using Scanner class to read

- Import java.util.Scanner;
- Declare Scanner and bind it to a file (last slide)
- Make sure there is input still to read

```
while (in.hasNext())
```

- Read next line

```
String line = in.nextLine();
```

- Read next word/token

```
String word = in.next();
```

- Read next integer

```
String word = in.nextInt();
```

LineNumberer

- Reads all lines of a file and sends them to the output file, preceded by line numbers
- Sample input file:

```
Mary had a little lamb
Whose fleece was white as snow.
And everywhere that Mary went,
The lamb was sure to go!
```

- Program produces the output file:

```
/* 1 */ Mary had a little lamb
/* 2 */ Whose fleece was white as snow.
/* 3 */ And everywhere that Mary went,
/* 4 */ The lamb was sure to go!
```

Copyright © 2008 by John Wiley & Sons.
All rights reserved.

ch11/fileio/LineNumberer.java

```java
import java.io.FileReader;
import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.util.Scanner;

public class LineNumberer
{
   public static void main(String[] args)
         throws FileNotFoundException
   {
      Scanner console = new Scanner(System.in);
      System.out.print("Input file: ");
      String inputFileName = console.next();
      System.out.print("Output file: ");
      String outputFileName = console.next();

      FileReader reader = new FileReader(inputFileName);
      Scanner in = new Scanner(reader);
      PrintWriter out = new PrintWriter(outputFileName);
      int lineNumber = 1;
```

Continued

Copyright © 2008 by John Wiley & Sons.
All rights reserved.

ch11/fileio/LineNumberer.java (cont.)

```

      while (in.hasNextLine())
      {
         String line = in.nextLine();
         out.println("/* " + lineNumber + " */ " + line);
         lineNumber++;
      }

      out.close();
   }
}
```

Big Java by Cay Horstmann
Copyright © 2008 by John Wiley & Sons.
All rights reserved.

Classwork

- Write a method to print contents of file, one word at a time
- Write a method to calculate the *mode* of an array of integers
- Write a method to calculate the *mode* of a file of integers