An Object Example

CS 1025 Computer Science Fundamentals I

Stephen M. Watt

University of Western Ontario
Objectives

• Cement the idea of objects.

• Become comfortable with objects *in Java*.

• Practice with arrays.

• See a dynamic data structure.

• Learn how to run a Java program in Eclipse.
Data Structures

• For most problems there are *choices* for how to organize the data.

• Example an m×n table of numbers:
  
  Use *one* array with the numbers for row \( i \) using slots 
  
  \[ i \times n + 0, i \times n + 1, i \times n + 2, \ldots, i \times n + (n-1) \] ,
  
  **OR** use an array with \( m \) entries, each of which is 
  
  an array of size \( n \).

• Example a list of bank deposits:
  
  Use an array of floating point numbers (§ and ø, 23.10) **OR** 
  
  use an array of integers (value × 100, 2310).

• The choice gives a data representation, or data structure 
  (you’ll see *much* more about this later).
An Example: Statistical Data

• A recording engineer is collecting data on the length of songs, represented as floating point numbers (number of seconds).
• There will be multiple data sets, each representing a playlist.
• It is not known in advance how many songs will be in each list.

• The lengths of the songs will be entered one at a time.
• At any point, it should be possible to ask for
  – the number of songs in a playlist
  – the total playing time of the playlist (given the inter-song gap time)
  – the average length of a song.
How to Represent the Data?

• Obviously there will be an array involved, but how big should it be?

• To start: use one that should “always be big enough”

• Keep track of how many slots are actually used.
class DataSet {
    private double[] data = new double[100];
    private int nused = 0;

    public void addValue(double val) {
        data[nused++] = val;
    }

    public int count() {
        return nused;
    }

    public double totalDataLength() {
        double tot = 0.0;
        for (int i = 0; i < nused; i++)
            tot += data[i];
        return tot;
    }

    public double averageDataLength() {
        return totalDataLength() / nused;
    }

    public double totalPlayLength(double gapLength) {
        return totalDataLength() + (nused - 1) * gapLength;
    }
}
class RecordingSessionOne {
    public static void main(String[] args) {
        DataSet songs = new DataSet();
        DataSet sounds = new DataSet();

        songs.addValue(90.4);   songs.addValue(102.3);
        songs.addValue(60.5);

        sounds.addValue(3.4);   sounds.addValue(8.3);
        sounds.addValue(1.5);   sounds.addValue(2.0);

        System.out.println("Average song length is "+
                           songs.averageDataLength());

        System.out.println("Average sound length is "+
                           sounds.averageDataLength());
    }
}
What If We Have More Than 100 Songs?

• Could use an array of size 1000. Or 1,000,000.
• That wastes a lot of space.

• One idea is to enlarge the array when needed.
• Then copy data from old array to new array.
• Forget about old one. It will be garbage collected.
  (In some languages you have to deallocate it.)

• Because the array is private to the object, the program that uses the object need not be changed.
class DataSet {
    private double[] data = new double[20];
    private int nused = 0;

    public void addValue(double val) {
        if (nused == data.length) {
            double[] newData = new double[2*data.length];
            for (int i = 0; i < data.length; i++)
                newData[i] = data[i];
            data = newData;
        }
        data[nused++] = val;
    }

    // All the rest is the same ......
How to Run this in Eclipse
Step 1. Create a Project
Step 1. Create a Project (contd)
Step 2. Create DataSet
Step 2. Create Data Set (contd)
Step 3. Enter the Code for DataSet

```java
// This class represents a collection of song lengths.
// The recording times are given in seconds.
// Initially space for 20 songs is allocated, and this
// is grown as necessary.

class DataSet {
    private double[] data = new double[10];
    private int nused = 0;

    public void addValue(double val) {
        if (nused == data.length) {
            double[] newData = new double[2*data.length];
            for (int i = 0; i < data.length; i++)
                newData[i] = data[i];
            data = newData;
        }
        data[nused++] = val;
    }

    public int count() { return nused; }

    public double totalDataLength() {
        double tot = 0.0;
        for (int i = 0; i < data.length; i++)
            tot += data[i];
        return tot;
    }
}
```
Step 3. Create the Main Program
Step 3. Create the Main Program (contd)
Step 4. Fix Errors
Step 5. Run