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×n+0, i×n+1, i×n+2, ..., i×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 ¢, 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 play list.
- 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 play list
  - the total playing time of the play list (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.

A First Implementation

```
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;
      }
  }
```

Using the Objects

```
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.

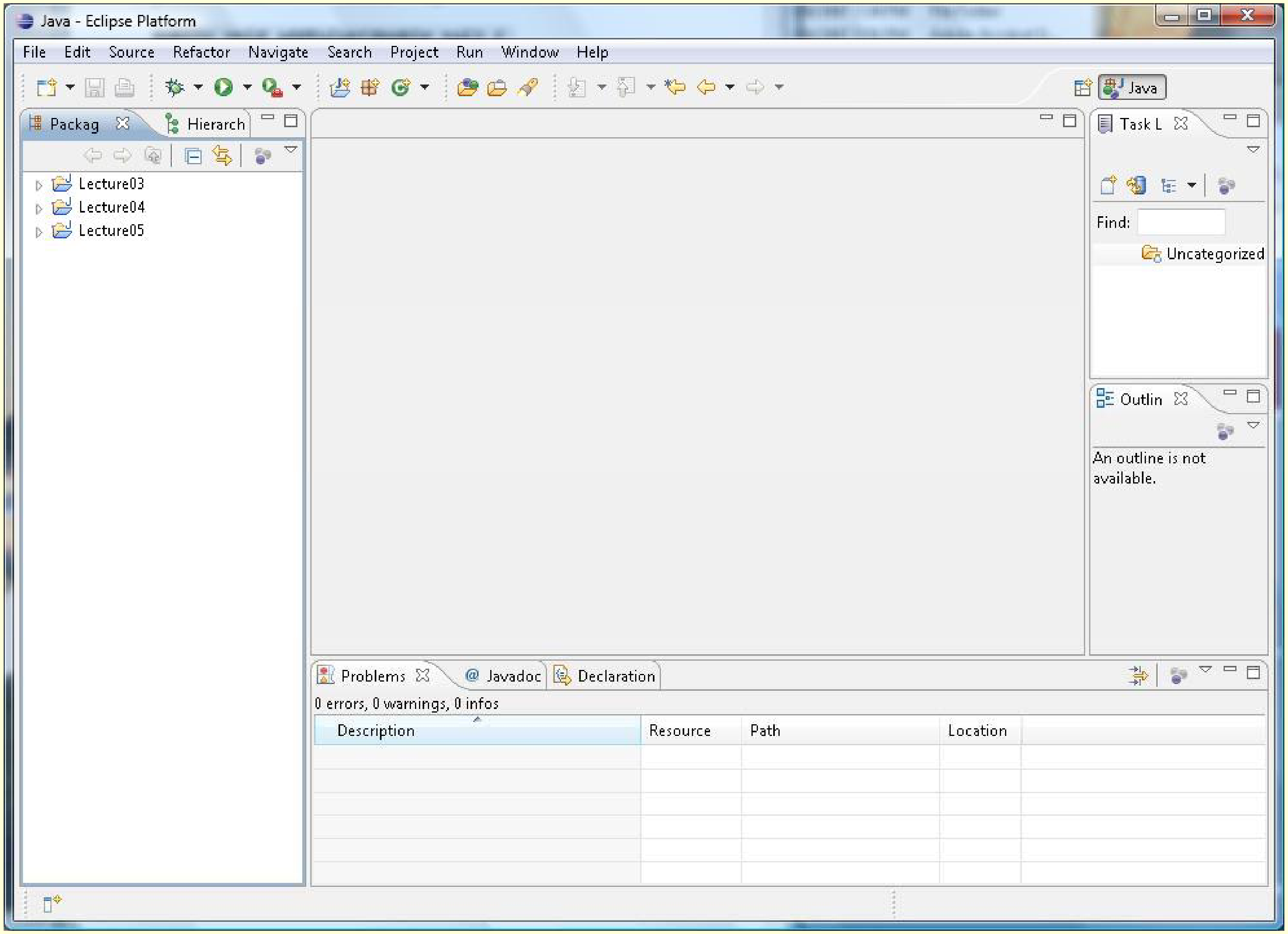
- One idea is to enlarge the array when needed.
- Then copy data from old array to new array.
- Forget about old one. It will be *garbage collected.* (In some languages you have to *deallocate* it.)

- Because the array is *private* to the object, the program that *uses* the object need not be changed.

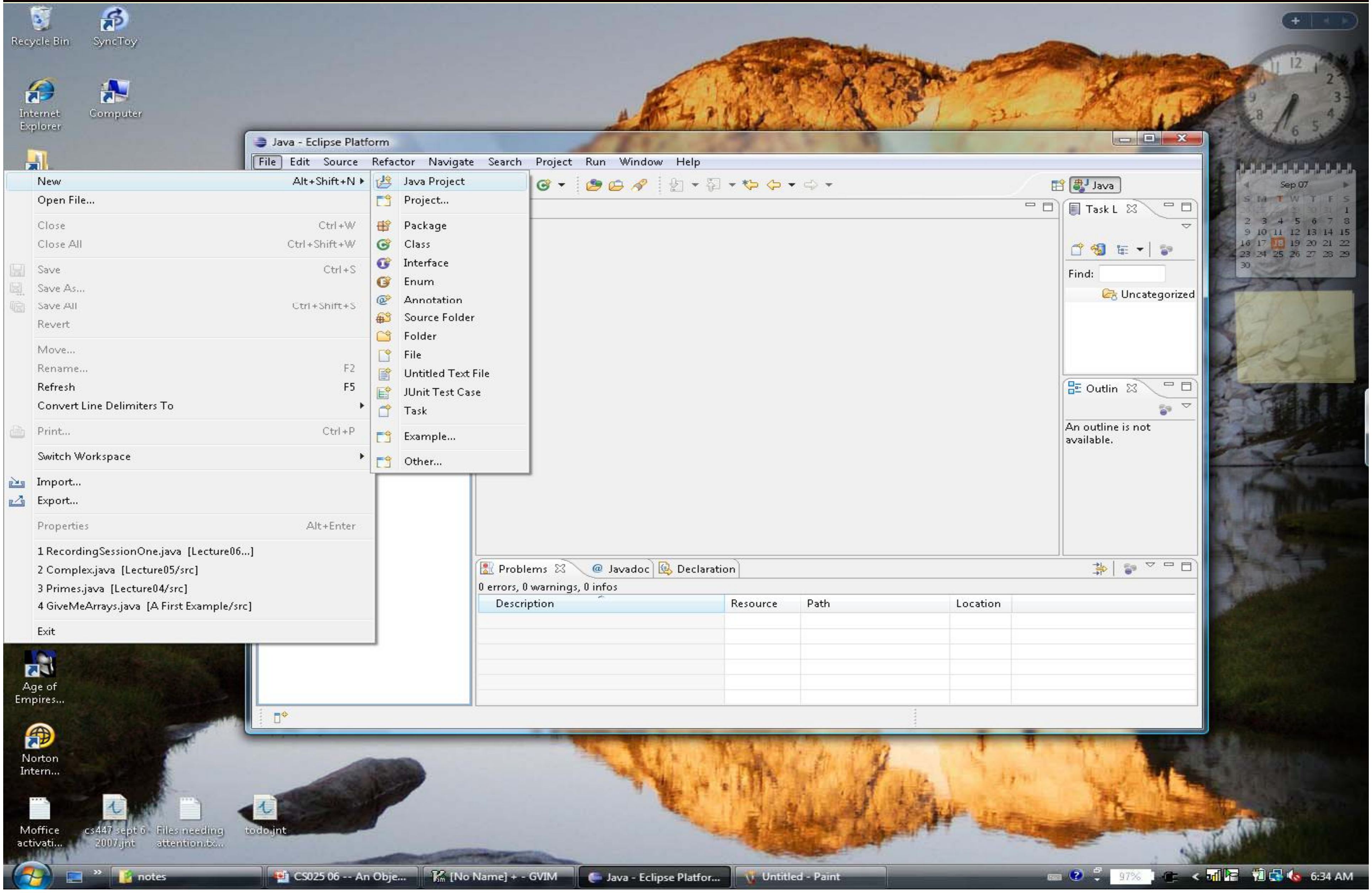
Only one method need be changed...

```
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 .....
  }
```

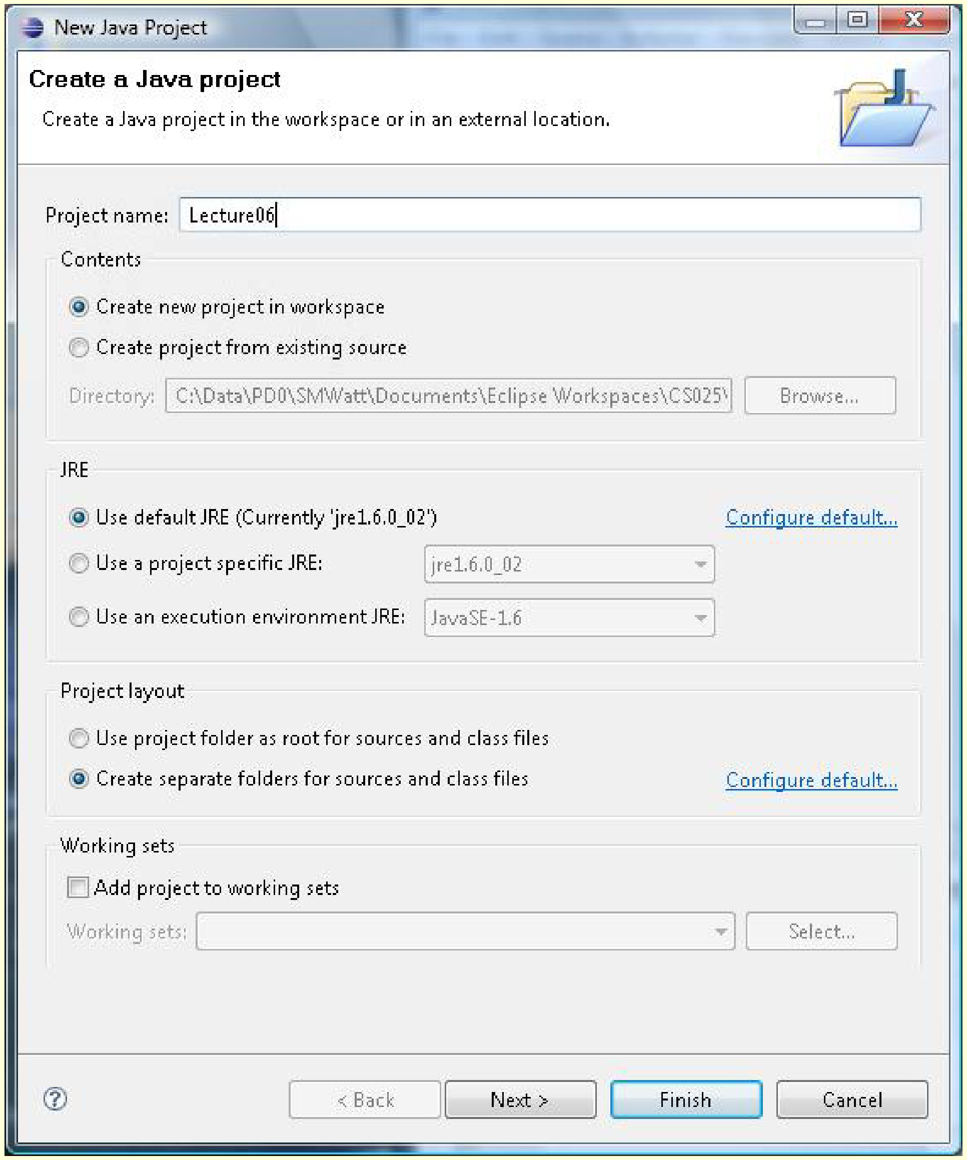
How to Run this in Eclipse



Step 1. Create a Project



Step 1. Create a Project (contd)



New Java Project

Create a Java project

Create a Java project in the workspace or in an external location.

Project name: Lecture06

Contents

- (●) Create new project in workspace
- ( ) Create project from existing source

Directory: C:\Data\PD0\SMWatt\Documents\Eclipse Workspaces\CS025\ [Browse...]

JRE

- (●) Use default JRE (Currently 'jre1.6.0_02') — Configure default...
- ( ) Use a project specific JRE: jre1.6.0_02
- ( ) Use an execution environment JRE: JavaSE-1.6

Project layout

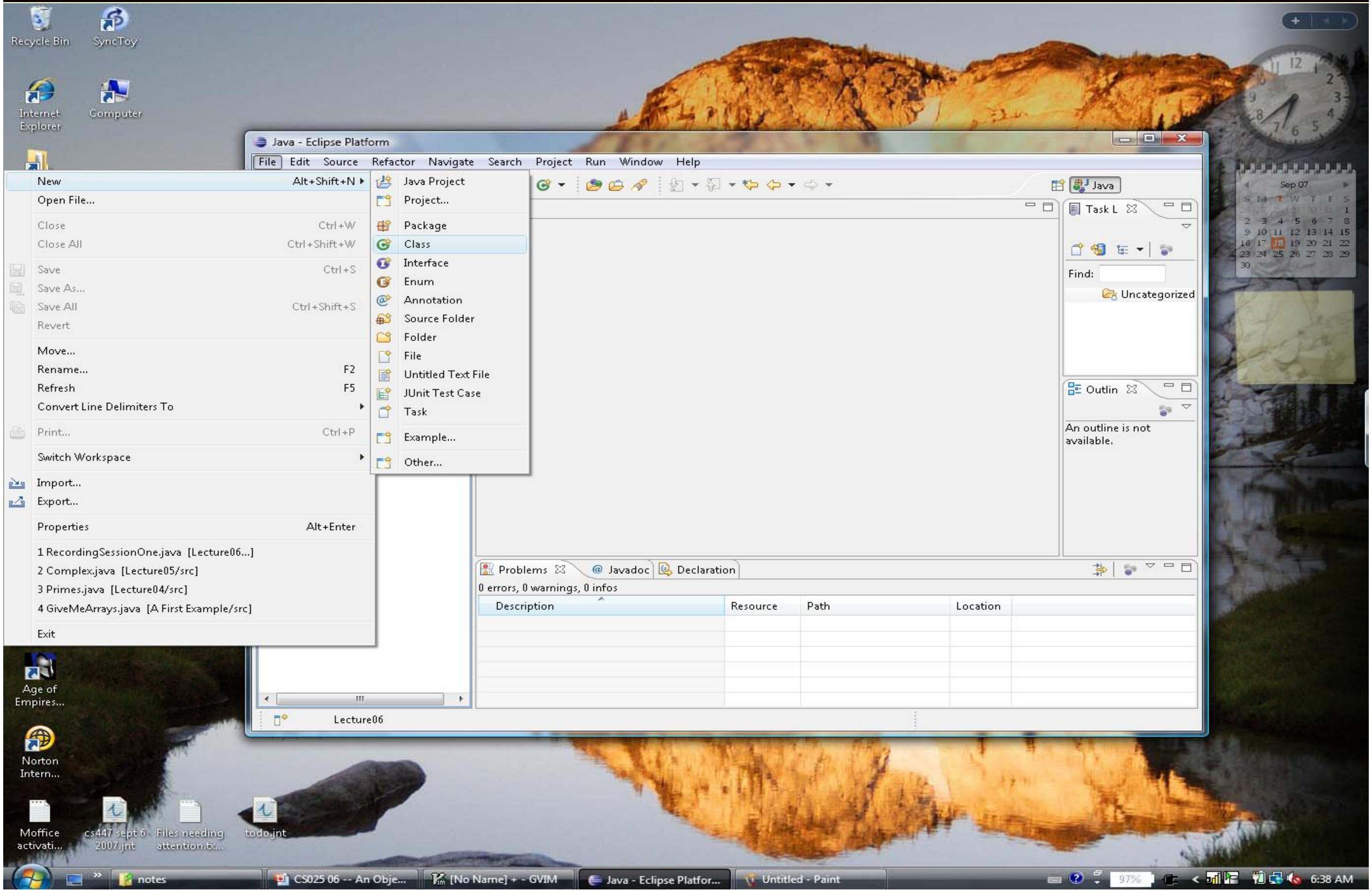
- ( ) Use project folder as root for sources and class files
- (●) Create separate folders for sources and class files — Configure default...

Working sets

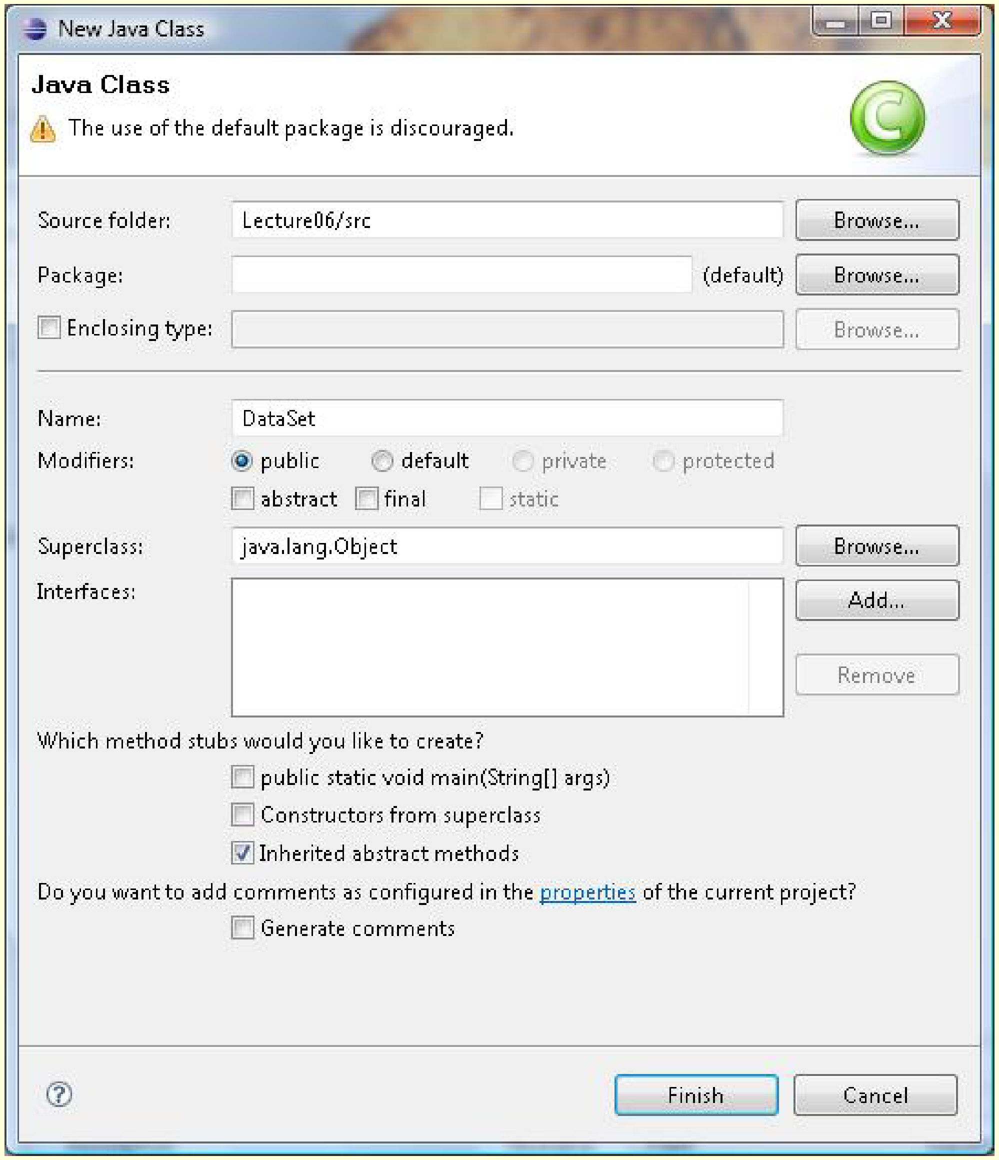
- [ ] Add project to working sets

Working sets: [Select...]

Step 2. Create DataSet



Step 2. Create `DataSet` (contd)


New Java Class

Java Class

The use of the default package is discouraged.

Source folder: Lecture06/src

Package: (default)

Enclosing type: [ ]

Name: DataSet

Modifiers: [✓ public] default private protected

[ ] abstract [ ] final [ ] static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

[ ] public static void main(String[] args)

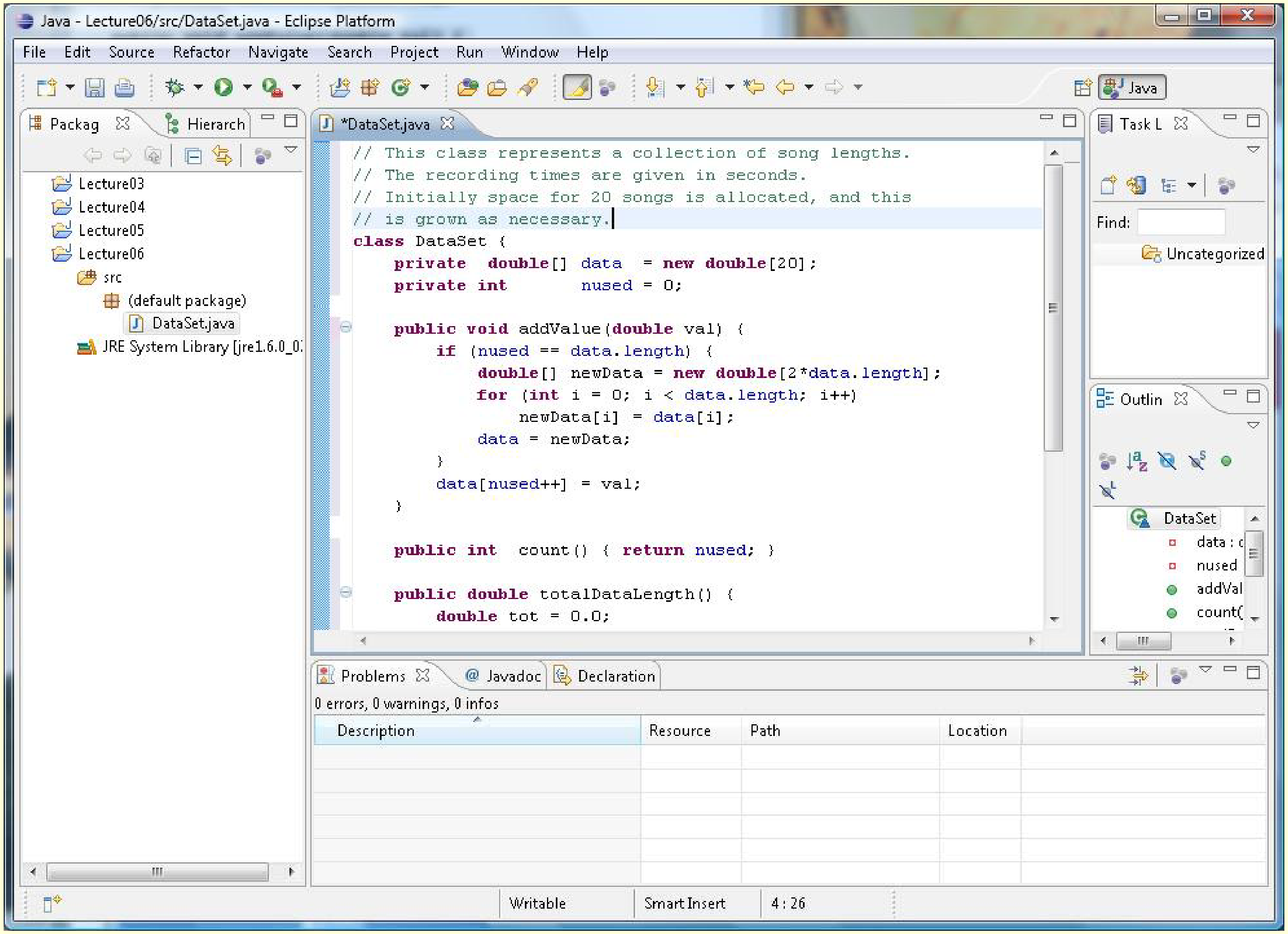
[ ] Constructors from superclass

[✓] Inherited abstract methods

Do you want to add comments as configured in the properties of the current project?

[ ] Generate comments

Step 3. Enter the Code for `DataSet`

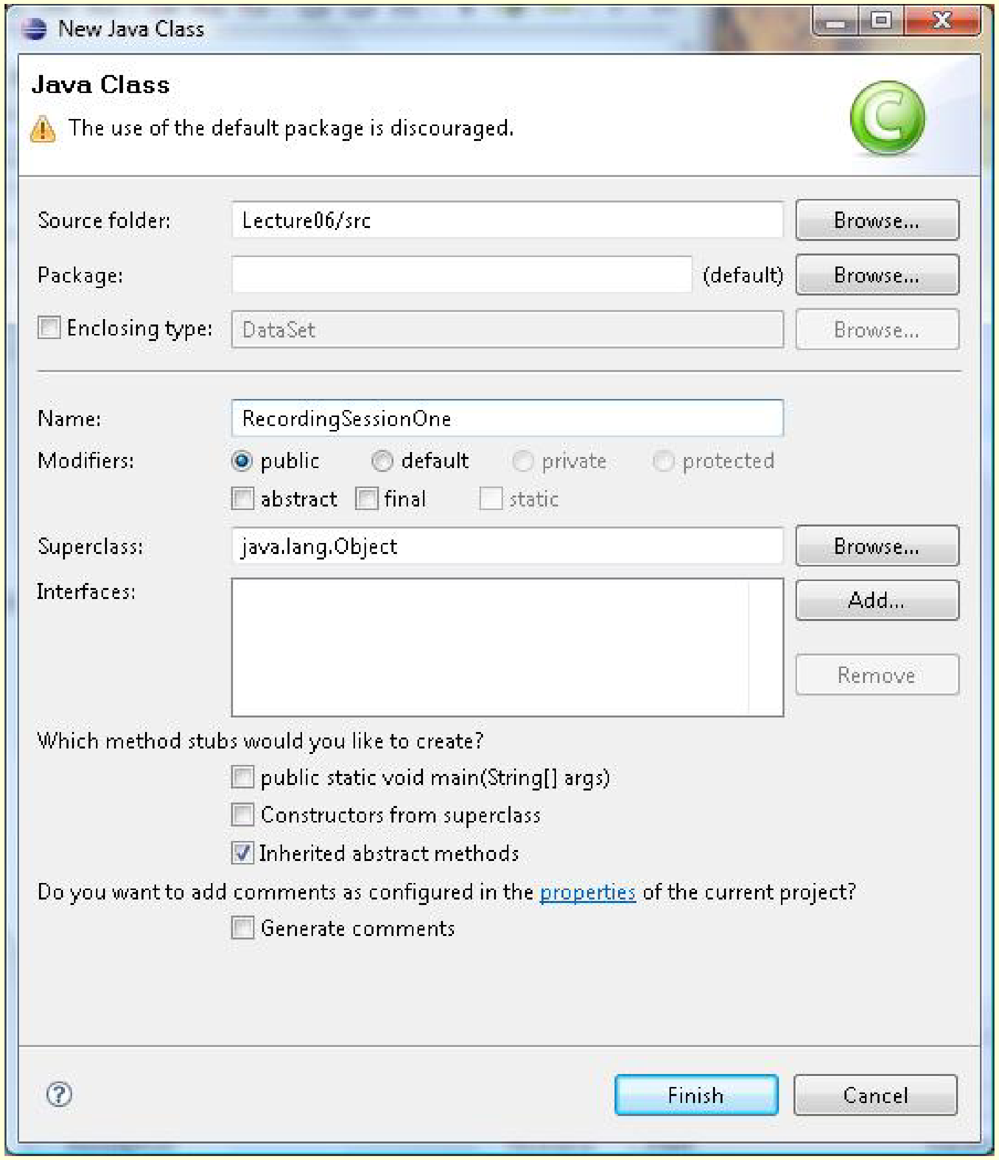


```
// This class represents a collection of song lengths.
// The recording times are given in seconds.
// Initially space for 20 songs is allocated, and this
// is grown as necessary.
class DataSet {
    private  double[] data  = new double[20];
    private int       nused = 0;

    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;
    }

    public int  count() { return nused; }

    public double totalDataLength() {
        double tot = 0.0;
```

0 errors, 0 warnings, 0 infos

Description	Resource	Path	Location

Step 3. Create the Main Program



New Java Class

Java Class

The use of the default package is discouraged.

Source folder: Lecture06/src

Package: (default)

☐ Enclosing type: DataSet

Name: RecordingSessionOne

Modifiers: [● public] ○ default ○ private ○ protected

☐ abstract ☐ final ☐ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

☐ public static void main(String[] args)

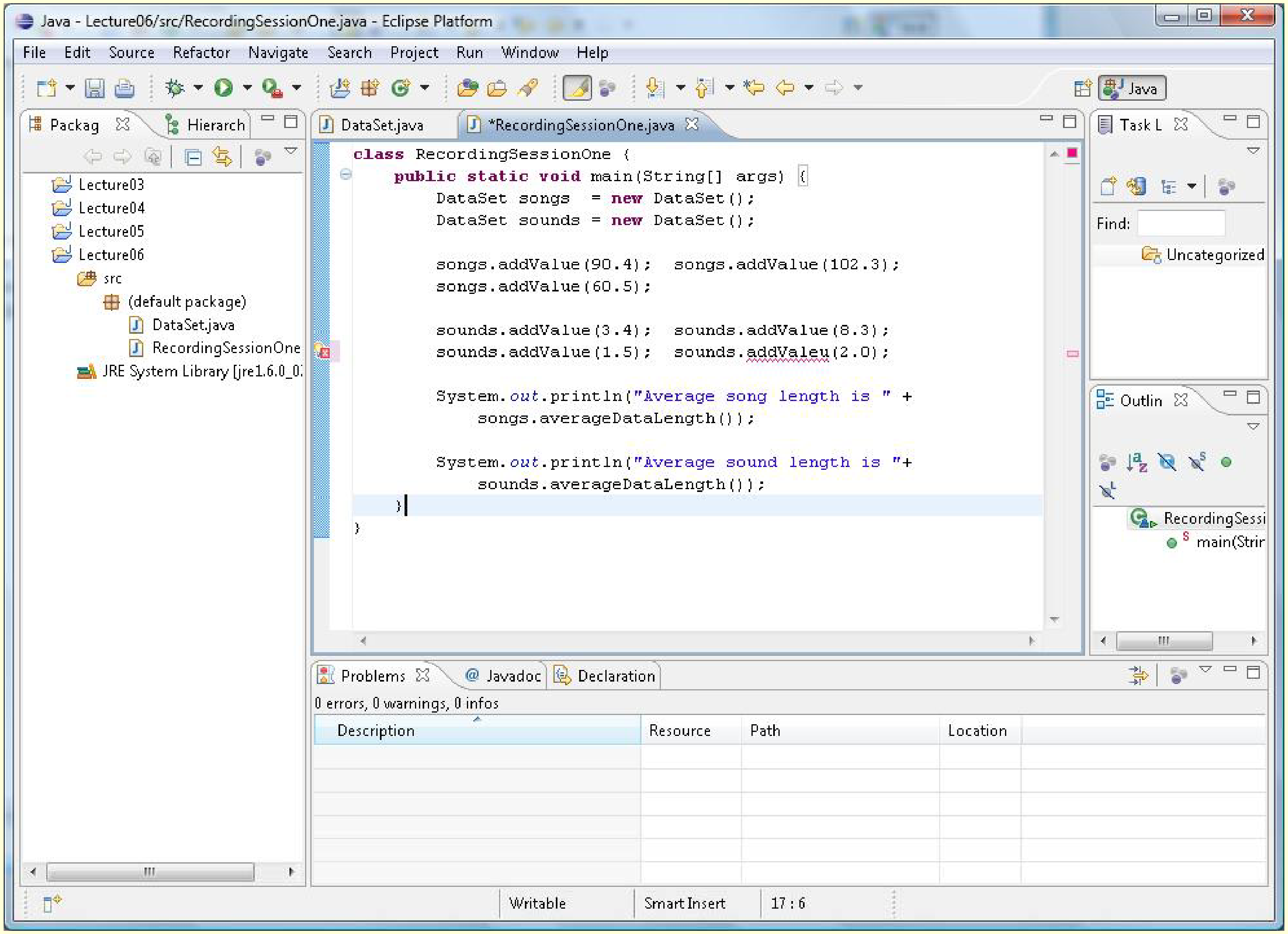
☐ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments as configured in the properties of the current project?

☐ Generate comments

Step 3. Create the Main Program (contd)

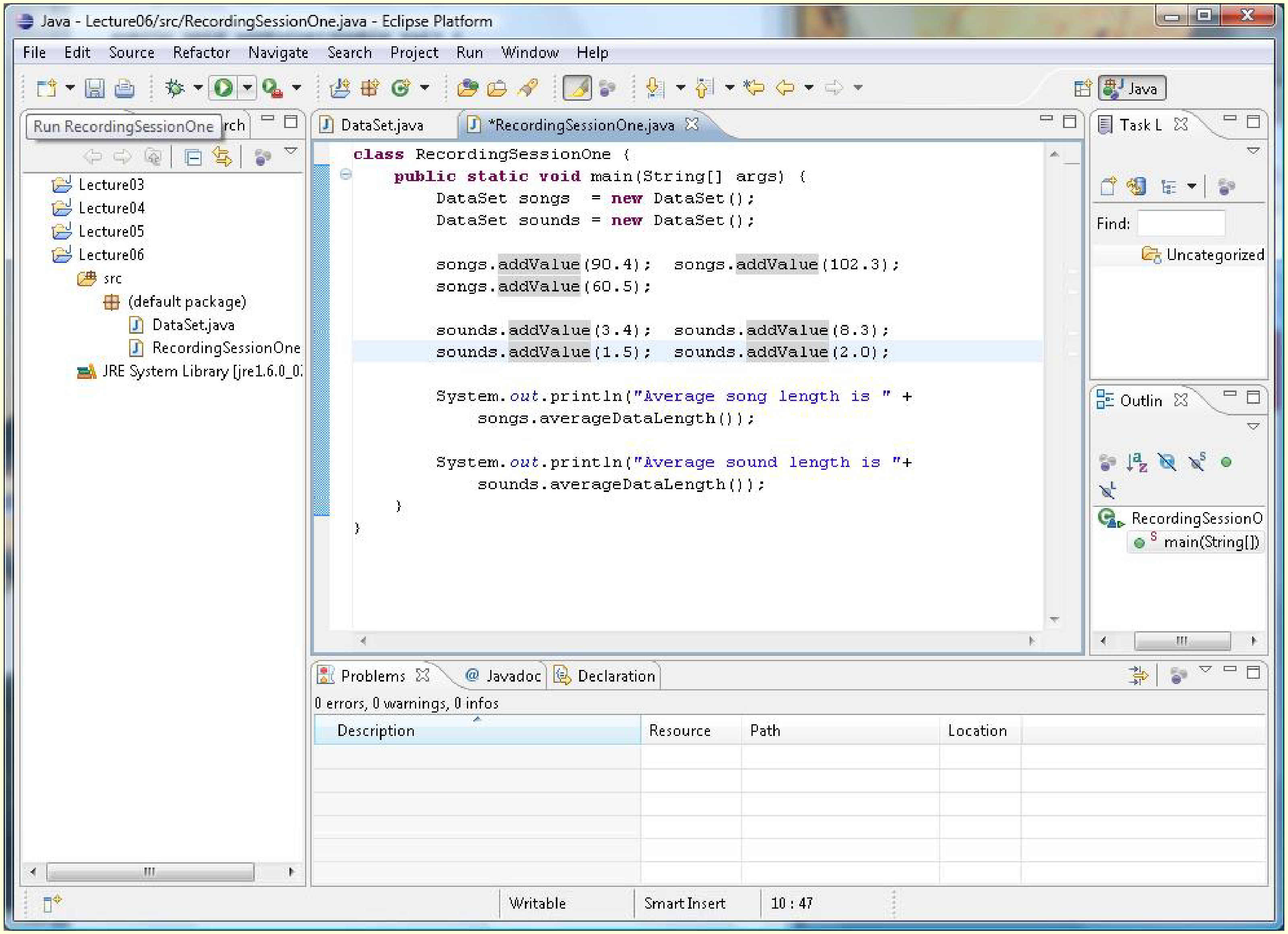


```java
class RecordingSessionOne {
    public static void main(String[] args) {
        DataSet songs  = new DataSet();
        DataSet sounds = new DataSet();

        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.addValeu(2.0);

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

        System.out.println("Average sound length is "+
            sounds.averageDataLength());
    }
}
```

0 errors, 0 warnings, 0 infos

Description	Resource	Path	Location

Step 4. Fix Errors

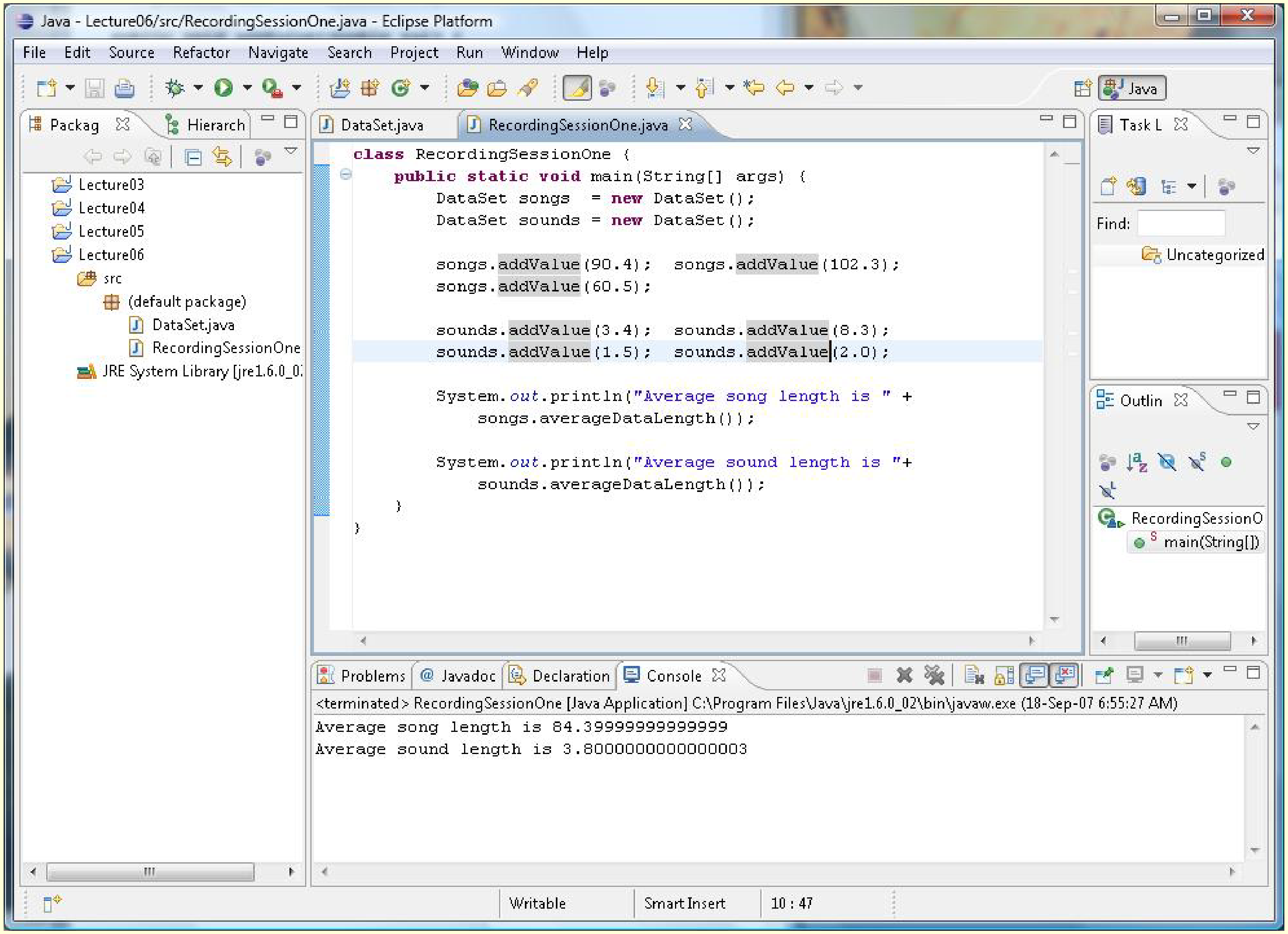


```
class RecordingSessionOne {
    public static void main(String[] args) {
        DataSet songs  = new DataSet();
        DataSet sounds = new DataSet();

        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());
    }
}
```

0 errors, 0 warnings, 0 infos

Description	Resource	Path	Location

Step 5. Run



```
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());
    }
}
```

```
<terminated> RecordingSessionOne [Java Application] C:\Program Files\Java\jre1.6.0_02\bin\javaw.exe (18-Sep-07 6:55:27 AM)
Average song length is 84.39999999999999
Average sound length is 3.8000000000000003
```