§14.6: Object Serialization

21 March 2007
CMPT167
Dr. Sean Ho
Trinity Western University

Review last time

- Classes in java.io
  - Byte-based vs. character-based vs. object-based
- File object and methods
- JFileChooser
- Formatted text I/O
  - Formatter, Scanner

Quiz 4: 10 minutes

- Contrast Swing with AWT. What is meant by lightweight widgets? [5]
- Discuss the delegation model of event handling [6]
- Name three sub-interfaces of EventListener. [3]
- Write a Java function [6]
  - void printMsg(String filename, int id, String message)

  that prints a given ID code and message to a file
  - printMsg(“msg.txt”, 15, “Hello, World!”);

  should put the text “15: Hello, World!” in the file.

Quiz 4: answers #1-2

- Contrast Swing with AWT. What is meant by lightweight widgets?
  - Swing: lightweight widgets written in Java, independent of local platform
  - AWT: heavyweight widgets tied to the local platform
- Discuss the delegation model of event handling
  - Events are handled by separate objects which implement the event-handling interface (e.g., ActionListener). The object which generates events is not the same object which handles.

Quiz 4: answers #3–4

- Name three sub-interfaces of EventListener.
  - ActionListener, AdjustmentListener, ComponentListener, ContainerListener, FocusListener, ItemListener, KeyListener, MouseListener, MouseMotionListener, TextListener, WindowListener
- Java function that prints ID code and message

```
void printMsg( String filename, int id, String message ) {
    Formatter out = new Formatter( filename );
    out.format( “%d: %s\n”, id, message );
    out.close();
}
```

  - (Exception handling optional)

What's on for today

- Exceptions in I/O
- Serializable objects
- Object-based I/O: ObjectInputStream
- Random-access files

Exceptions in file I/O

- An instance of the class SecurityException is raised if file permissions fail:

```
try {
    out = new Formatter( “out.txt” );
} catch ( SecurityException e ) {
    System.err.println( “No write permissions!” );
}
```

- FileNotFoundException, IllegalStateException
- Scanner raises NoSuchElementException if the data is in the wrong format
- EOFException, IOException

Serializable objects

- Serialization is converting an object to a representation that can be written to a stream
- The Serializable interface is a tag:
  - Interface with no methods
  - Used to identify what objects are serializable
- Primitive types are serializable
- Arrays of serializable objects are serializable
- A class can be tagged as serializable if all its instance variables are serializable
  - Non-serializable vars can be declared transient (skipped during serialization)

Object-based I/O

- Use FileInputStream / FileOutputStream to open a file for binary I/O
    - fos = new FileOutputStream(“output.db”)
- Wrap the stream in an ObjectInputStream / ObjectOutputStream to use object serialization
    - oos = new ObjectOutputStream(fos);
- Use readObject/writeObject to do the I/O:
    - oos.writeObject(myobj);
  - readObject() returns a generic Object:
    - Cast it back to the original type
    - myobj = (MyObj) ios.readObject();

Random-access files

- Sequential files are hard to modify in-place
  - Must erase and rewrite entire file
- Random-access files:
  - `file = new RandomAccessFile( “user.db”, “rw” );`
- Can be used in place of FileInputStream / FileOutputStream, e.g., to do object-based I/O
- File position pointer:
  - `file.seek( num_bytes );`
  - Seek to position relative to start

TODO

- Lab5 due Wed 11Apr:
  - File I/O
  - Store inventory and point-of-sale system
  - This last lab is longer:
    - Due in 3 weeks
    - Worth 60 points
- Last day for submitting late labs is Fri 13Apr
- Last day of classes is Mon 16Apr
- Final exam is Fri 20Apr 2-4pm