3134 Data Structures in Java

Lecture 16
Mar 26 2007
Shlomo Hershkop
Announcements

- More example code today
- Some hw clarifications
Homework questions:

- Will upload virus other data today...needed to work it out with courseworks system
- Window size
- code issues
- NB
Today

- Java serializable
- Java comparable
- generics
Serializable
Problem outline

- Imagine you are writing a program using java, would like to personally address all error to the user.

- Would like to get users name, and say “Hey SHLOMO, error on drive E, file X!”

- Problem is that everytime we start we need to ask the user their name
Solution 1

- Ask once
- Save to file

- Works with some limitations
  - Not always can we save the information in a way that make sense into a file
  - Example: complex information DS
Serial Solution

- Grab an output stream
- Write the java object to the output stream
- Let other classes worry about physical storage

- Then we can grab the object back the other way
Idea:

- Allow the programmer to take a snapshot of live memory, and save it in a binary form.....
- No need to recreate data in the objects

1. We need to tell java we want to save a certain class
2. Get object output stream
3. Save the objects
   1. Can even be a bunch of them in an array or in a ds for example (tree, hash, etc)
Simple Example 1

```java
public class student {
    private String name;
    private int age;

    ...

    public String getName() {
        return name;
    }
}
```
public class Student implements Serializable {
    private String name;
    private int age;

    ... 
    public String getName() {
        return name;
    }
}
Save routine

```java
public static void main(String[] args) {
    Student one = new Student...

    try{
        FileOutputStream fos = new FileOutputStream("saved.data");
        ObjectOutputStream out = new ObjectOutputStream(fos);
        out.writeObject(one);
        out.close;
    }catch(IOException ioe){ .. }
```
Load Routine

```java
try{
    FileInputStream fis = new FileInputStream("saved.data");
    ObjectInputStream in = new ObjectInputStream(fis);
    Student oldone = (Student)in.readObject();
}
```
Important note

- Only objects which extend serializable can be saved
- basically:
  - Java will go through each field member of the class and try to serialize each one
  - If your class has field variables which don’t implement this....
Options

Two choices:

1. Mark those non serializable as ‘transient’ this tells the JVM not to save those variables

2. Implement a custom `writeObject` and `readObject`

   can then choose which fields to save and load, and initialize any others
How to add read/write

```java
private void writeObject(java.io.ObjectOutputStream out)
    throws IOException{
    out.writeObject(name);
    out.writeInt(age);
}

private void readObject(java.io.ObjectInputStream in)
    throws IOException, ClassNotFoundException{
    name = (String)in.readObject();
    age = in.readInt();
}
```
Uses

- So where is serializable useful ??
Collection

- We covered arrays and queues and other basic DS

- Many times will create specialized class to hold specific data
  - Example student class
  - Might want to order a bunch of item of that type
Useful interface: Comparable

- Allows you to define a compareTo function which returns -1, 0, 1 for comparing same objects
- Throws classCastException is not correct class to compare
Next

- Bingo example, but first

- Generics
generics

- Idea: loosen type constraints

- When coding/creating object need to specify type....allows you to push off decision to later on
class Example2<T, V> {
    T ob1;
    V ob2;

    Example2(T o1, V o2) {
        ob1 = o1;
        ob2 = o2;
    }

    T getob1() {
        return ob1;
    }

    V getob2() {
        return ob2;
    }
}
Bingo

- Is everyone familiar with the game??

- Lets look over a good example and discuss the code
Generics with constraints...
Example 3

class Example3<Q extends Number> {

    Q[] nums; // array of Number or subclass

    Stats(Q[] o) {
        nums = o;
    }

    double sum() {
        double sum = 0.0;
        for(int i=0; i < nums.length; i++)
            sum += nums[i].doubleValue();
        return sum;
    }
}
Usage

Double dnums[] = { 1.1, 2.2, 3.3, 4.4, 5.5 };  

Example3<Double> dob = new Example3<Double>(dnums);  

double w = dob.sum();  

System.out.println("dob sum is " + w);
Example 4

class TwoD {
    int x, y;
    TwoD(int a, int b) {
        x = a;
        y = b;
    }
}

class ThreeD extends TwoD {
    int z;
    ThreeD(int a, int b, int c) {
        super(a, b);
        z = c;
    }
}

class FourD extends ThreeD {
    int t;
    FourD(int a, int b, int c, int d) {
        super(a, b, c);
        t = d;
    }
}
class Coords<T extends TwoD> {
    T[] coords;
    Coords(T[] o) { coords = o; }
}
static void showXY(Coords<?> c) {
    System.out.println("X Y Coordinates:");
    for(int i=0; i < c.coords.length; i++)
        System.out.println(c.coords[i].x + " " + c.coords[i].y);
    System.out.println();
}
static void showXYZ(Coords<? extends ThreeD> c) {
    System.out.println("X Y Z Coordinates:");
    for(int i=0; i < c.coords.length; i++)
        System.out.println(c.coords[i].x + " " + c.coords[i].y + " " + c.coords[i].z);
    System.out.println();
}
static void showAll(Coords<? extends FourD> c) {
    System.out.println("X Y Z T Coordinates:");
    for(int i=0; i < c.coords.length; i++)
        System.out.println(c.coords[i].x + " " +
                             c.coords[i].y + " " +
                             c.coords[i].z + " " +
                             c.coords[i].t);
    System.out.println();
}
Using in methods

- Just learn how to read the following:

```java
static <T, V extends T> boolean isIn(T x, V[] y) {
    for(int i=0; i < y.length; i++) {
        if(x.equals(y[i])) return true;
    }
    return false;
}
```