`ioutil`: A Lightweight Toolkit for GUIs and Text Stream I/O CS1 and CS2

Contact Information
Kenneth A. Lambert, Washington and Lee University, klambert@wlu.edu
Martin Osborne, Western Washington University, martin@cs.wwu.edu

Problem Statement
Beginning students find it difficult to write realistic, event-driven Java programs with GUIs. One challenge is to find a solution that avoids distortions of GUI-based programming, such as the use of a series of simple modal dialogs for inputs. Another challenge is to provide an API that is lightweight enough that it both resembles "real Java" and can be easily trimmed away when the students reach maturity.

Beginning students also find it difficult to include terminal input and text file I/O in their Java programs. There are three basic challenges:

1. They must use a decorator pattern to set up a text stream object.
2. They must embed their text stream code in a try/catch statement.
3. They must convert string representations from input to data of the appropriate types, such as `int` or `double`.

Each of these challenges requires beginning students to master some fairly advanced concepts before they can do something as simple as input an integer.

Solution Overview
`ioutil`, like its predecessors, `BreezyGUI` [Lambert99, Lambert00] and `BreezySwing`, is a package of classes that allows students to construct easy, realistic GUIs. `ioutil` is somewhat lighter in weight than its predecessors, however. The package includes an `EasyGridLayout` class that encapsulates the details of a gridbag layout, allowing the student to place in containers many types of standard components by specifying simple row and column positions, a width in columns, and a height in rows. The classes `DoubleField` and `IntegerField` are extensions of `JTextField` that hide the details of numeric I/O. Otherwise, students are responsible for setting up their own containers (frames, applets, or dialogs), setting up their own components, and writing the listener classes to respond to events in these controls.

The use of `ioutil` focuses students' attention immediately on the idea that a graphical user interface is built from a collection of objects that cooperate by sending messages. The rest of

their time can then be focused on designing and implementing the algorithms that respond to events in the user interface and the code that manages the data in the application's model.

`ioutil` also contains classes that support easy text stream I/O. Students create instances of `KeyboardReader` and `FileReader` and send them the same set of messages for terminal input or text file input of characters, integers, doubles, words, or lines of text. A `FileWriter` class is used for easy text file output. The details of setting up the text streams and handling the required exceptions are encapsulated in these classes.

Experience with the Solution

We developed `ioutil` for the second edition of a CS2 book. Several users of the first edition of this book complained that they had trouble weaning their students away from `BreezySwing` by the end of the CS2 course. We are using `ioutil` in CS2 a class for the first time this winter term, and students have enjoyed using it.

API Documentation

Here is an example of an interactive program that uses `ioutil`'s text stream capabilities:

```
import ioutil.*;

public class PersonApp{

   public static void main(String[] args){
      KeyboardReader kbd = new KeyboardReader();
      FileWriter = new FileWriter("test.txt");
      int age;
      String name;
      for (int i = 1; i <= 3; i++){
         name = kbd.readLine("Name: ");
         age = kbd.readLine("Age: ");
         writer.println(name);
         writer.println(age);
      }
      writer.close();
      FileReader flrdr = new FileReader("test.txt");
      while (!flrdr.eof()){
         name = flrdr.readLine();
         age = flrdr.readInt();
         System.out.println(name + " " + age);
      }
   }
}
```

Complete documentation and downloads of `ioutil` are available at www.cs.wwu.edu/martin, scroll down to "Open source software packages."