

# Taming a Professional IDE for the Classroom

Charles Reis, Corky Cartwright  
[creis@alumni.rice.edu](mailto:creis@alumni.rice.edu), [cork@cs.rice.edu](mailto:cork@cs.rice.edu)

# Java at Intro Level

- Widespread, great for OO
- But, lots of complexity...
  - Syntax
  - Command Line, Classpath
  - Console I/O
  - `public static void main(String[] args)`

# IDEs Can Help

- Help with language syntax
- Easy compile / run buttons

# Pedagogic IDEs

- Great for Intro:

- Simple and interactive

- Limited features

- No plug-ins

The screenshot shows the DrJava IDE interface. The top menu bar includes File, Edit, Tools, Debugger, and Help. The toolbar below has buttons for New, Open, Save, Close, Cut, Copy, Paste, Undo, Redo, Find, Compile All, Reset, Test, and Javadoc. The main window displays Java code for a linked list implementation. The code defines an abstract class `List` with an abstract method `length()` and a helper method `toStringHelp()`. It also defines a concrete class `Empty` that extends `List` and a class `Cons` that extends `List`. The `Cons` constructor takes a `first` parameter and a `rest` parameter. The code also includes methods `getFirst()`, `getRest()`, `length()`, and `toStringHelp()`. The bottom panel shows the `Interactions` tab with a welcome message and a session where a `Cons` list is created and its length is printed. The status bar at the bottom right shows the path `/home/creis/testsrc/listdemo/ListDemo.java` and the time `23:3`.

```
abstract class List {
    abstract public int length();
    abstract String toStringHelp();
    public String toString() { return "[" + toStringHelp() + "]"; }
}

class Empty extends List {
    public static final Empty ONLY = new Empty();

    private Empty() { }

    public int length() { return 0; }
    String toStringHelp() { return ""; }
}

class Cons extends List {
    protected int first;
    protected List rest;

    public Cons(int f, List r) {
        first = f;
        rest = r;
    }

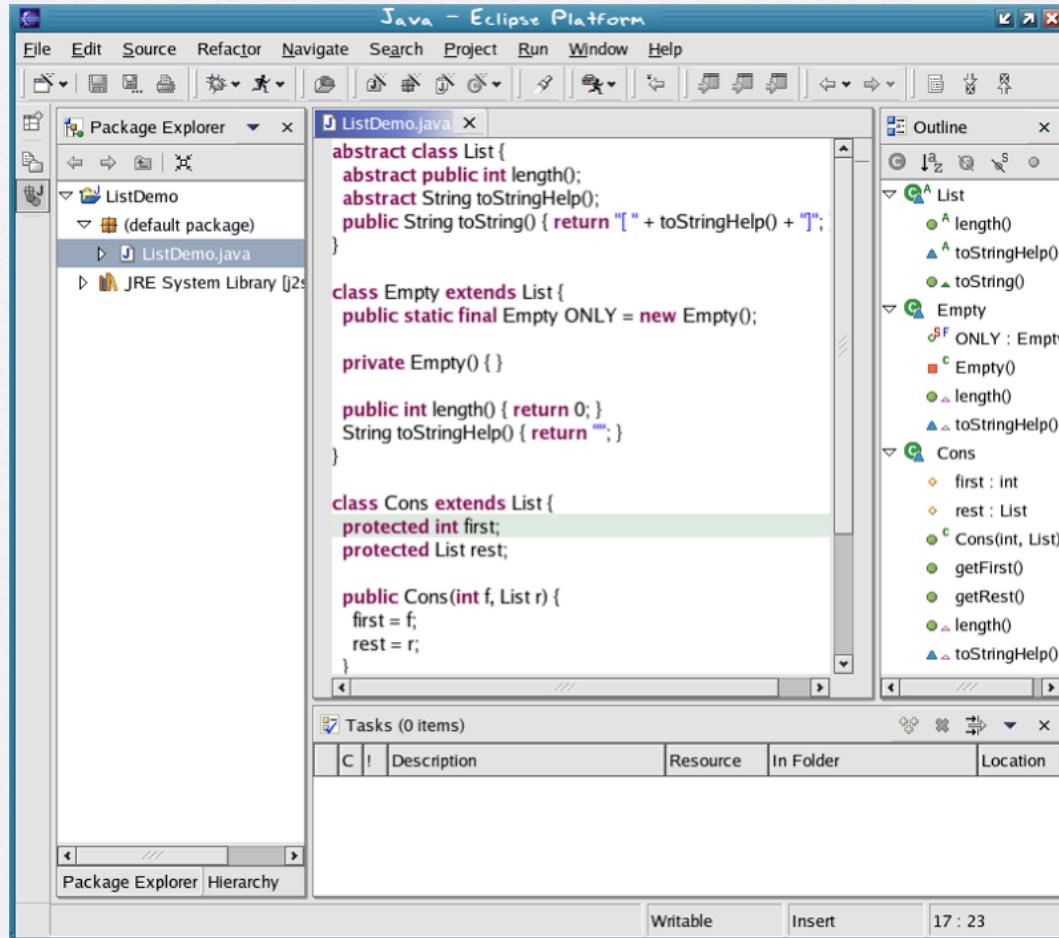
    public int getFirst() { return first; }
    public List getRest() { return rest; }

    public int length() { return 1 + rest.length(); }
    String toStringHelp() { return first + " " + rest.toStringHelp(); }
}

Welcome to DrJava.
> List myList = new Cons(3, new Cons(5, Empty.ONLY));
> myList
[ 3 5 ]
> myList.length()
2
>
```

DrJava

# Professional IDEs



- Many advanced features
- Plug-ins (Eclipse)
- Bad for Intro:
  - Complex UI
  - Console I/O, main

Eclipse

# Modify Eclipse

- Improve for intro level
- Why not use a Pedagogic IDE?
  - Other educational plug-ins
  - Consistent IDE across curriculum
  - Easy transition to advanced tools

# DrJava Plug-in for Eclipse

- Simplify Eclipse's interface
- Add Interactions Pane

# Interactions Pane

- On-the-fly evaluation
- No need for `main`, console I/O
- Lecture demos, student tinkering

# Vs. Eclipse Scrapbook

- Can evaluate code outside `main`, but...
- All-at-once evaluation
- Less convenient, powerful, intuitive

# Demo

# Related Plug-ins

- Gild, Penumbra
  - Further simplify Eclipse's interface
  - Courseware integration

# Result

- Eclipse + DrJava: ready for Intro Level
  - Effective on-the-fly evaluation
  - Integrate with other plug-ins
  - Easy path for growth