Programming – Lecture 15

Going Beyond ACM

- Lectures Beyond Inf-ProgOO
- IDEs Beyond Eclipse
- Projects Beyond Homework
- Java Beyond ACM
- Programming Beyond Inf-ProgOO

Programming – Lecture 15

Going Beyond ACM

- Lectures Beyond Inf-ProgOO
- IDEs Beyond Eclipse
- Projects Beyond Homework
- Java Beyond ACM
- Programming Beyond Inf-ProgOO

Implementing Languages

Context-free grammars

Parsing source code

Building compilers

Prof. Dr. Hanus

Übersetzerbau

Wahlpflicht

Efficiency and Data Structures

Defining “performance”

Efficiency of algorithms

Efficiency of data structures

Prof. Dr. Jansen

ADS

2. Semester

Prof. Dr. Jansen

Effiziente Algorithmen

Wahlpflicht

Graphics Programming

Image data streams

Feature extraction

Modeling 3D objects

Rendering

Prof. Dr. Koch

Einführung in die Bildverarbeitung

Wahlpflicht

Prof. Dr. Koch

Computer Graphik

Wahlpflicht

Memory Management

Stack and Heap

Virtual memory

Paging and memory hierarchy

Prof. Dr. Landsiedel

Betriebs- und Kommunikationssysteme

2. Semester

Concurrency

Race conditions and synchronization

Distributed systems

Patterns for parallel and distributed systems

PD Dr. Huch

Concurrent and Distributed Progr.

Wahlpflicht

Prof. Dr. Hasselbring

[…] parallele und verteilte Systeme

Wahlpflicht

Real-Time Programming

Control flow and data flow languages

Model-based design

Concurrency and scheduling

Worst-case execution times

Prof. Dr. von. Hanxleden

Synchrone Sprachen

Wahlpflicht

Prof. Dr. von Hanxleden

Embedded real-time systems

Wahlpflicht

Programming Patterns

Requirements analysis

Specification tools

Design patterns

Prof. Dr. Hasselbring

Softwaretechnik

4. Semester

Prof. Dr. Hasselbring

Softwareprojekt

4. Semester

Programming Languages

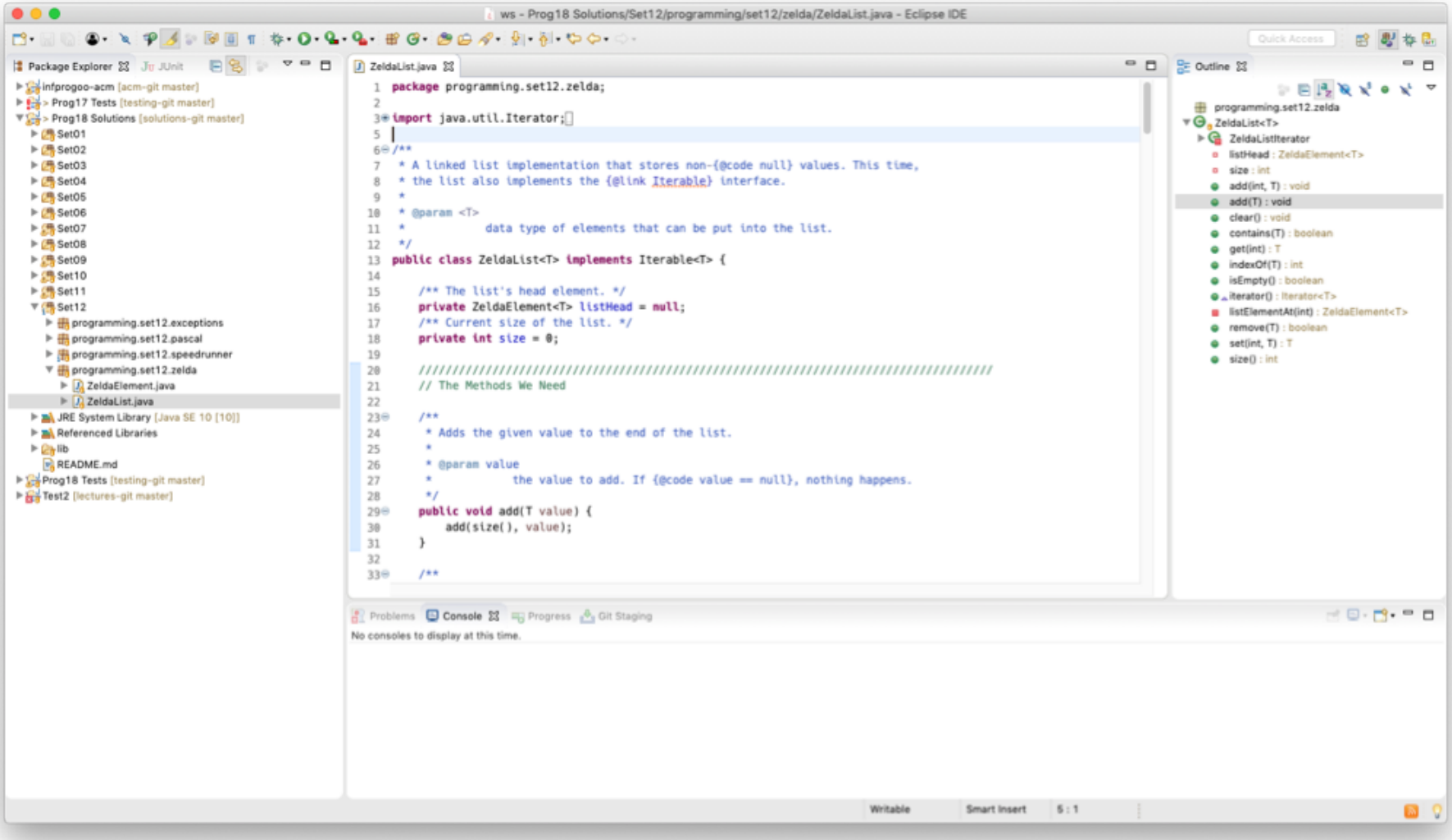
Well…

Programming – Lecture 15

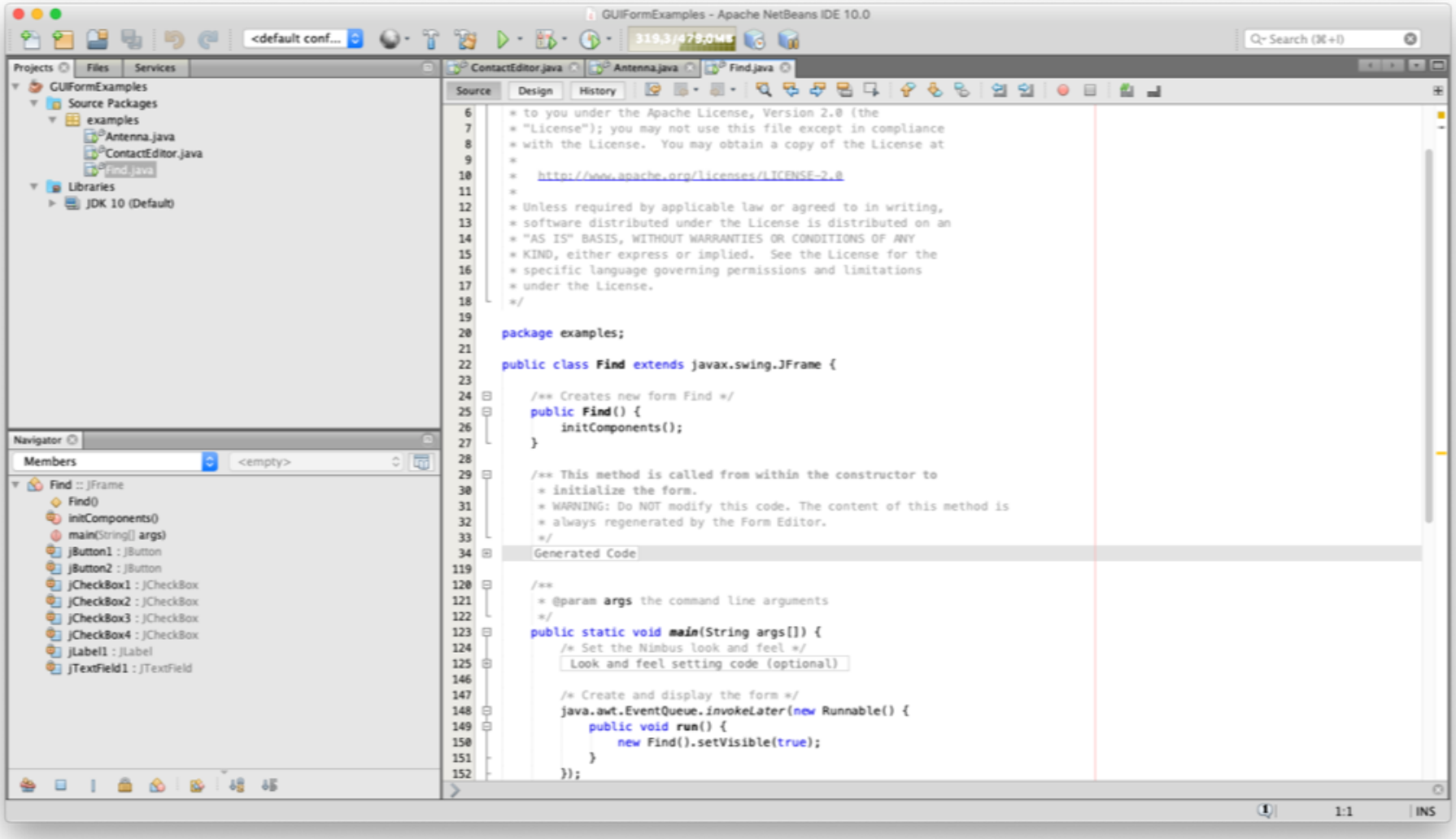
Going Beyond ACM

- Lectures Beyond Inf-ProgOO
- IDEs Beyond Eclipse
- Projects Beyond Homework
- Java Beyond ACM
- Programming Beyond Inf-ProgOO

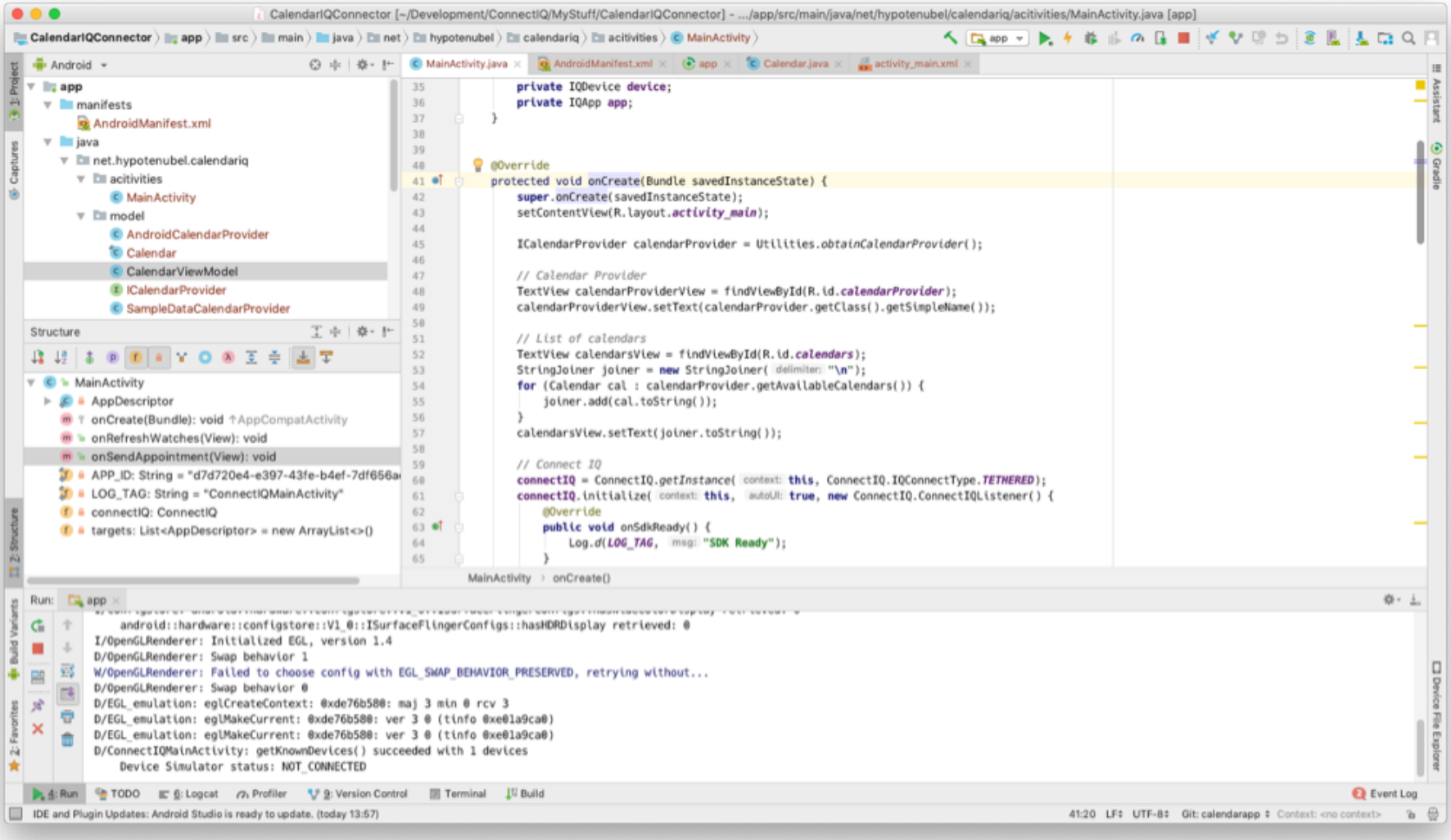
More Eclipse



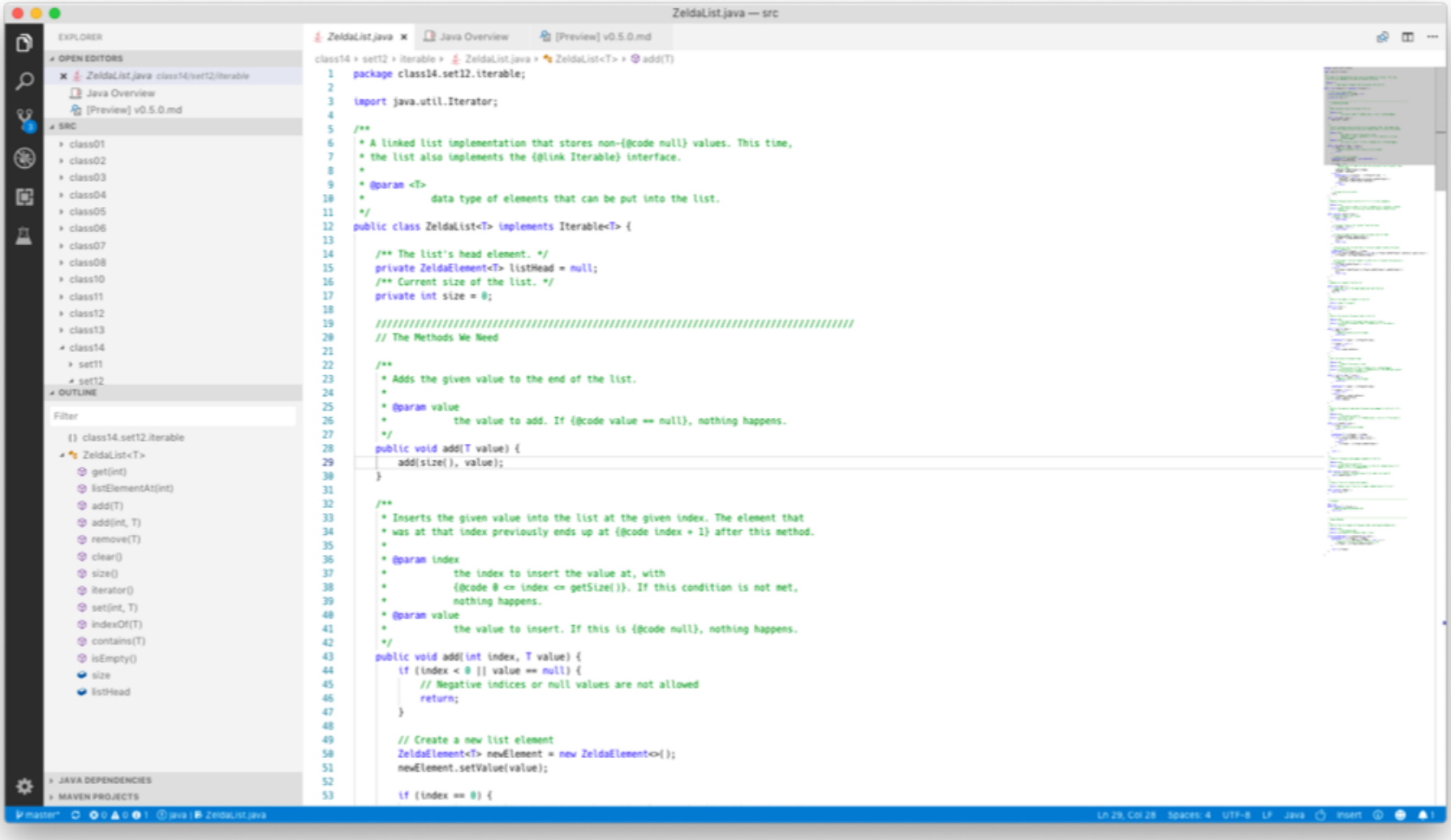
NetBeans


GUIFormExamples - Apache NetBeans IDE 10.0
Projects
Files
Services
GUIFormExamples
Source Packages
examples
Antenna.java
ContactEditor.java
Find.java
Libraries
JDK 10 (Default)
Navigator
Members
Find :: JFrame
Find()
initComponents()
main(String[] args)
jButton1 : JButton
jButton2 : JButton
jCheckBox1 : JCheckBox
jCheckBox2 : JCheckBox
jCheckBox3 : JCheckBox
jCheckBox4 : JCheckBox
jLabel1 : JLabel
jTextField1 : JTextField
ContactEditor.java
Antenna.java
Find.java
Source
Design
History
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
package examples;
public class Find extends javax.swing.JFrame {
/** Creates new form Find */
public Find() {
initComponents();
}
/** This method is called from within the constructor to
* initialize the form.
* WARNING: Do NOT modify this code. The content of this method is
* always regenerated by the Form Editor.
Generated Code
/**
* @param args the command line arguments
public static void main(String args[]) {
/* Set the Nimbus look and feel */
Look and feel setting code (optional)
/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
new Find().setVisible(true);
}
});
1:1
INS

IntelliJ IDEA



Visual Studio Code



Remember

IDEs are monsters
and each works differently,
so give them time
before forming an opinion!

Programming – Lecture 15

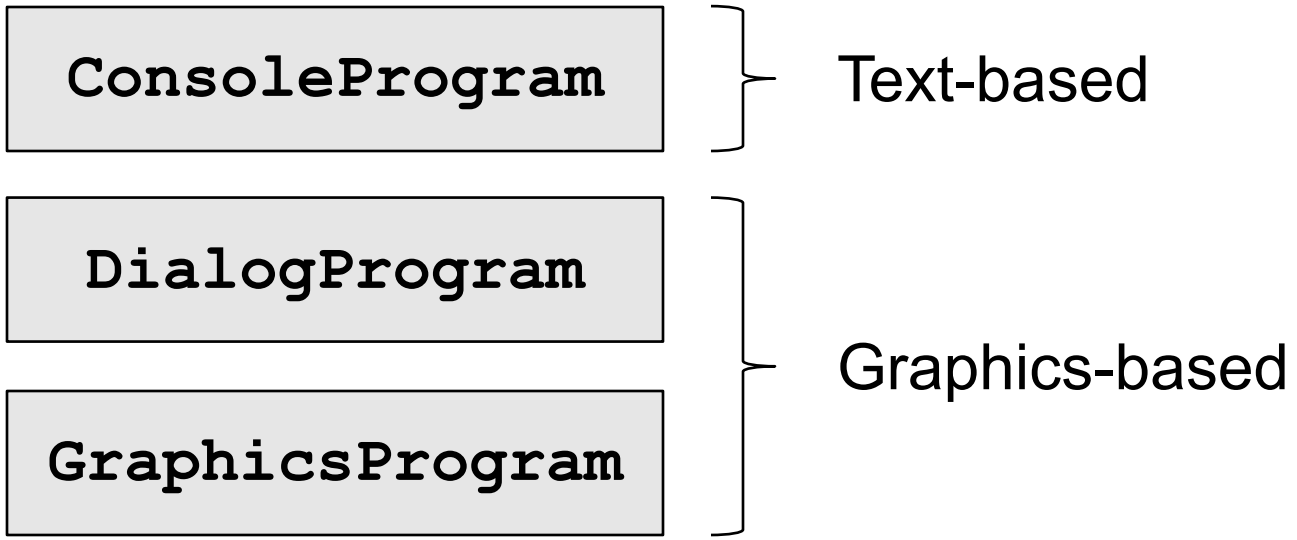
Going Beyond ACM

- Lectures Beyond Inf-ProgOO
- IDEs Beyond Eclipse
- Projects Beyond Homework
- Java Beyond ACM
- Programming Beyond Inf-ProgOO

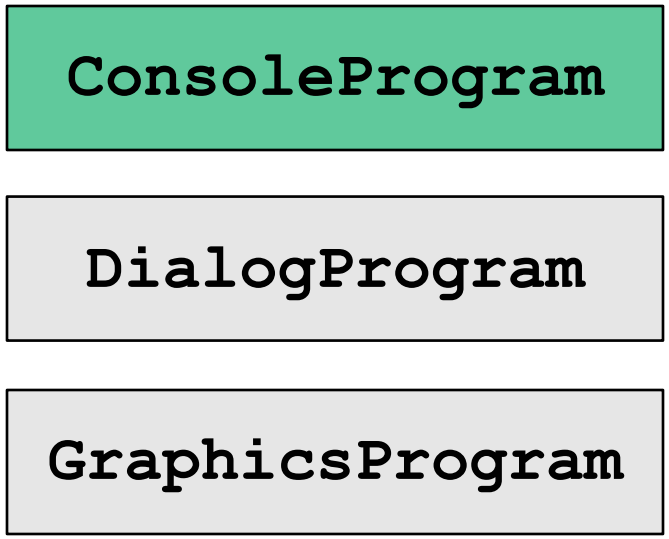
The ACM Library

So far, we have used the ACM library to write our programs.

Most of our programs were instances of one of these classes:



Alternatives



Alternatives: Text-Based

Simple Java programs are text-based by default. Remember Lecture 2:

```
public class HelloWorld {
    public static void main (String[] args) {
        System.out.println("Hello World!");
    }
}
```

Class Definition

ACM:

```
public class Wubbel
    extends ConsoleProgram {
   //...
}
```

Java:

No restrictions. Every class can have a **`main`** method.

Program Entry Point

ACM:

```
public void run() {
   //...
}
```

Java:

```
public static void main(String[] args) {
   //...
}
```

Console Output

ACM:

```
println(...);
```

Java:

```
System.out.println(...);
```

In Eclipse, just type "syso<ctrl><space>"

Console Input

ACM:

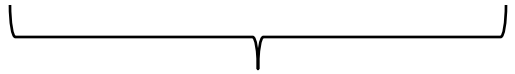
```
String s = readLine();
int i = readInt();
```

Java:

```
Scanner scanner = new Scanner(System.in);
String s = scanner.next();
int i = scanner.nextInt();
```

Command-Line Arguments

```
public static void main(String[] args) {
   //...
}
```



Array of Arguments

Command-Line Arguments

Entry Point:

```
public static void main(String[] args) {
    for (String arg : args)
        System.out.println(arg);
}
```

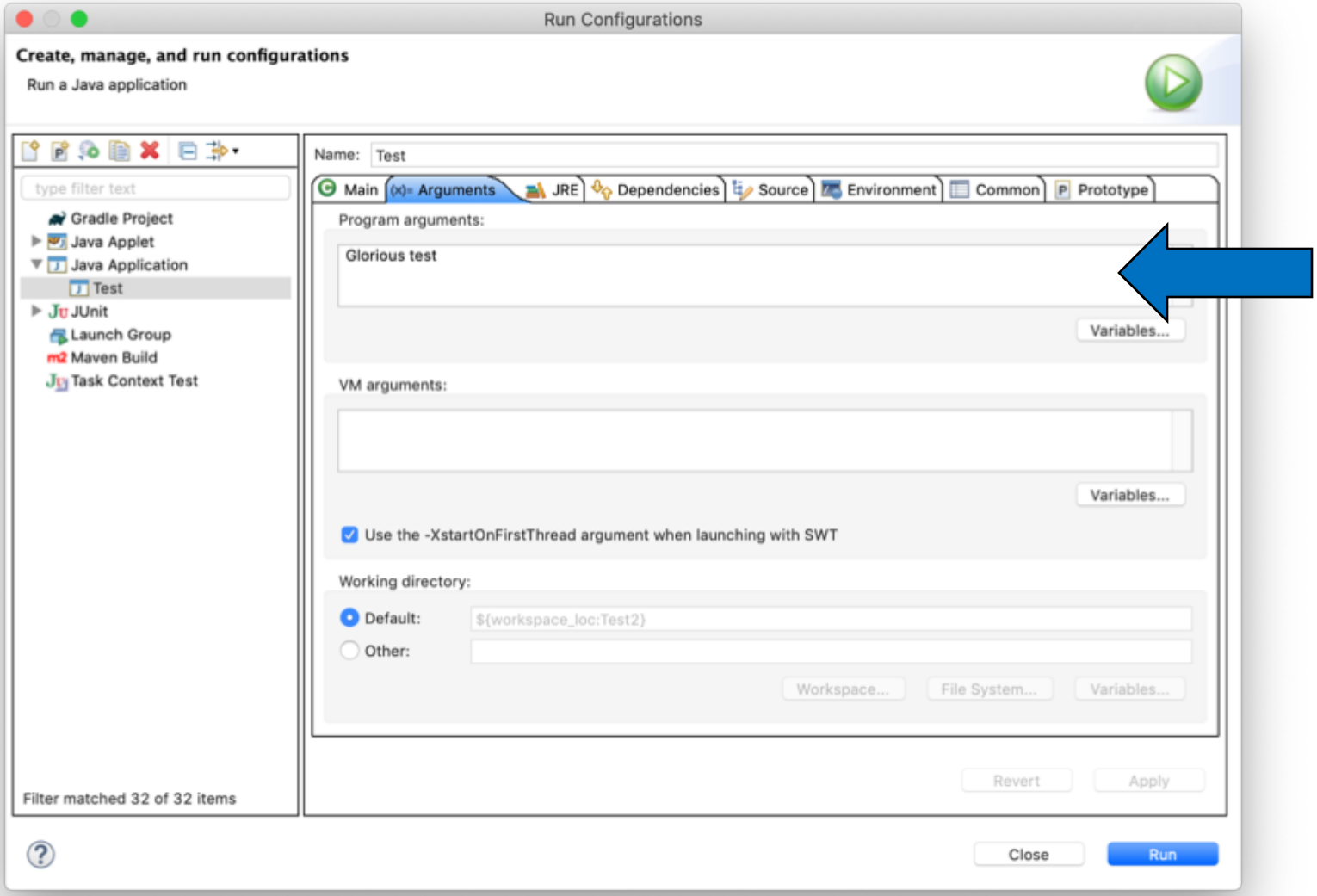
Invocation:

```
java MyClass Glorious test
```

Output:

```
Glorious
test
```

Command-Line Arguments



Common Code Patterns

Static Code:

```
public static void main(String[] args) {
    // Call only other static methods
}
```

Only use for small, single-class programs.

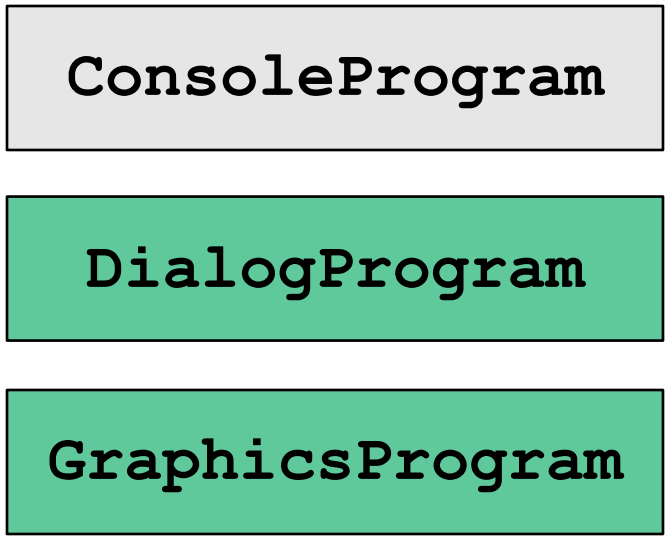
Common Code Patterns

Slim Main Method:

```
public static void main(String[] args) {
    MyClass mc = new MyClass(args);
    mc.startDoingStuff();
}
```

Standard pattern. Only uses main to instantiate your main class and have it start doing stuff.

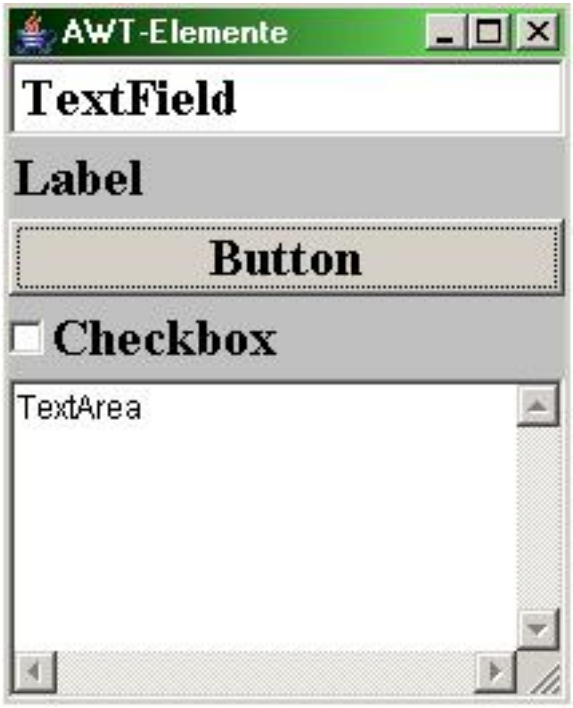
Alternatives



Alternatives: Graphics-Based

Java GUI land
has become a sad place…

Java AWT



Wikipedia Example

Java AWT

The original Java GUI library.

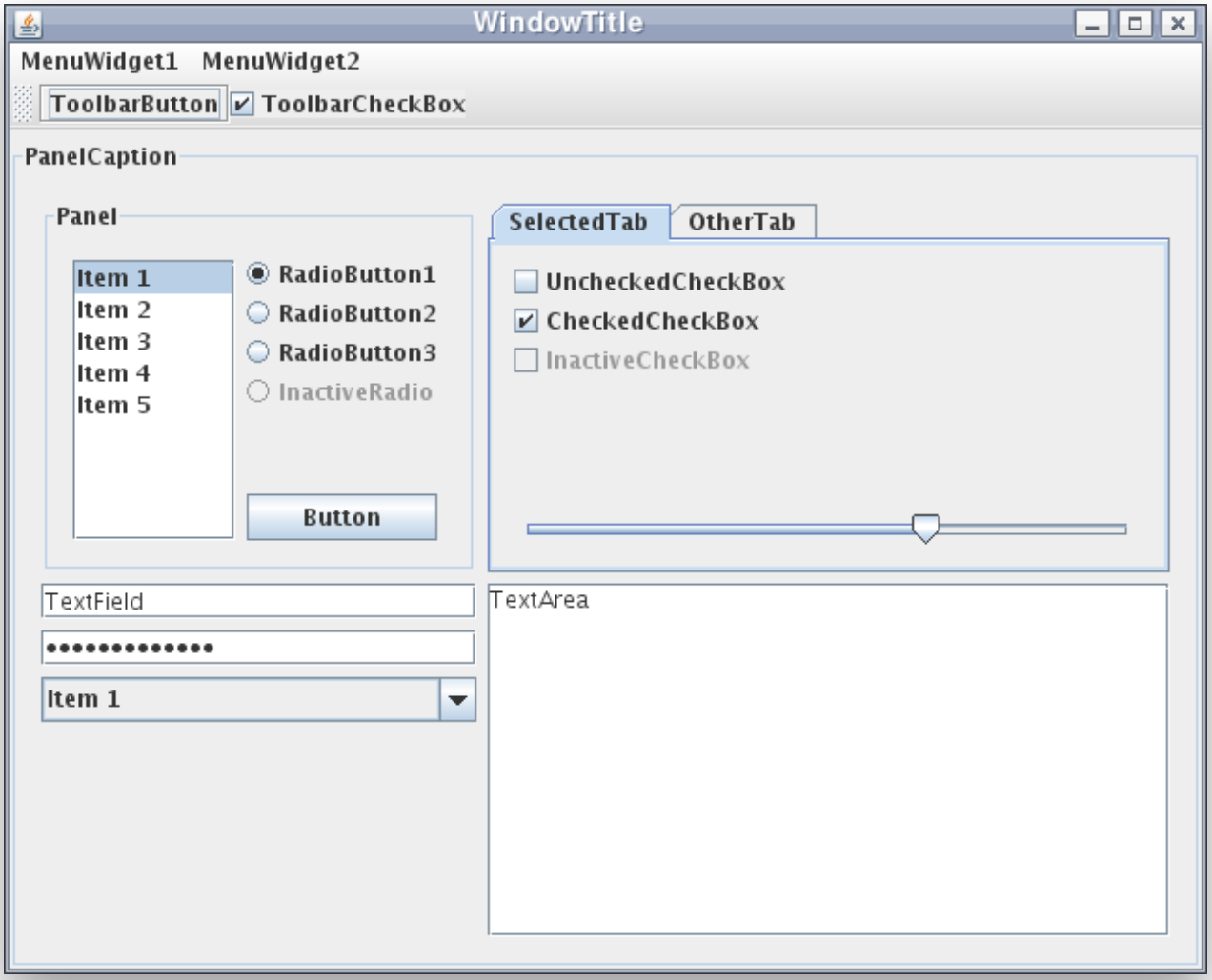
Can look plain and outdated.

Not rich in features.

Verdict

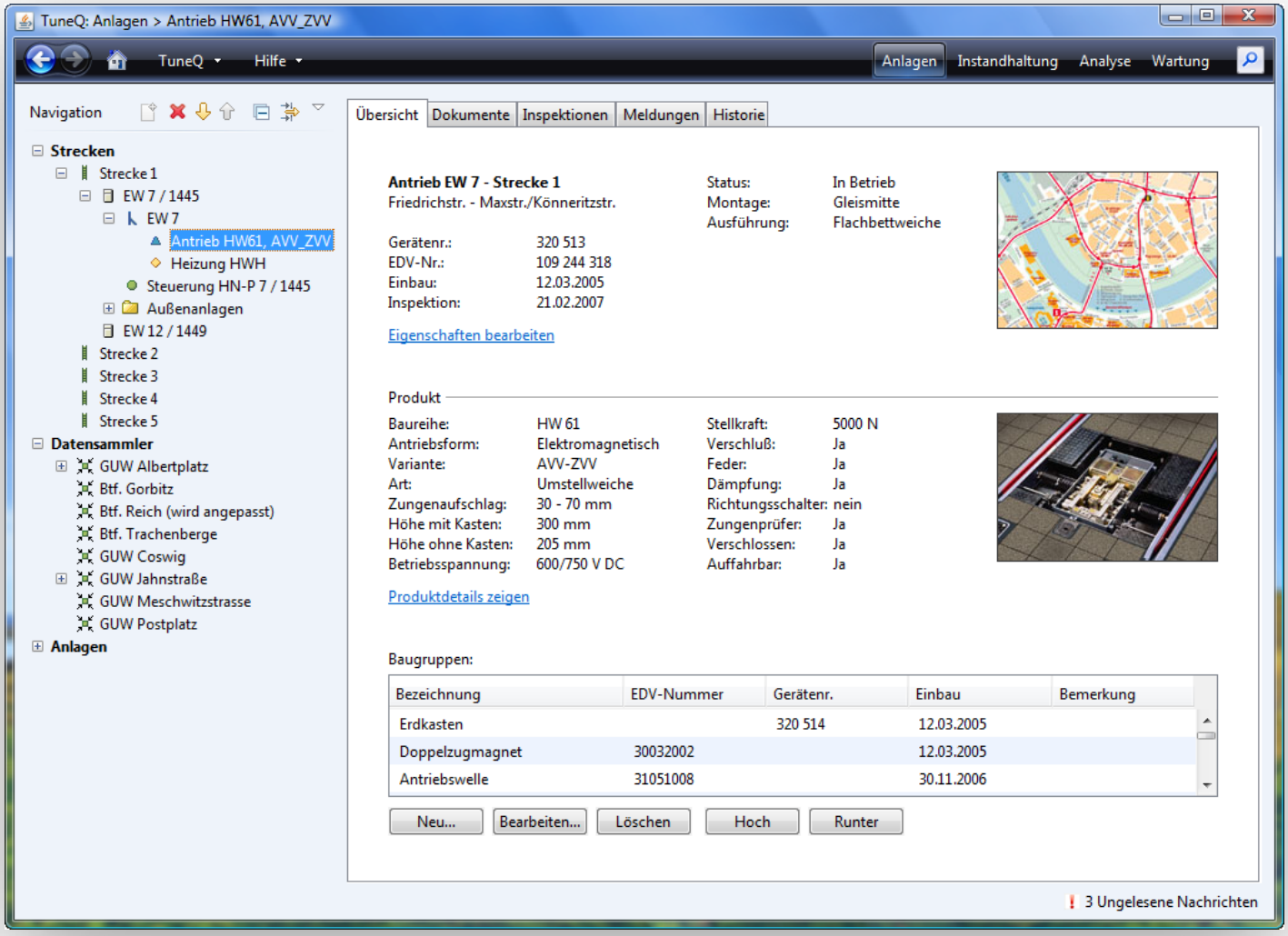
Don’t use directly.

Java Swing



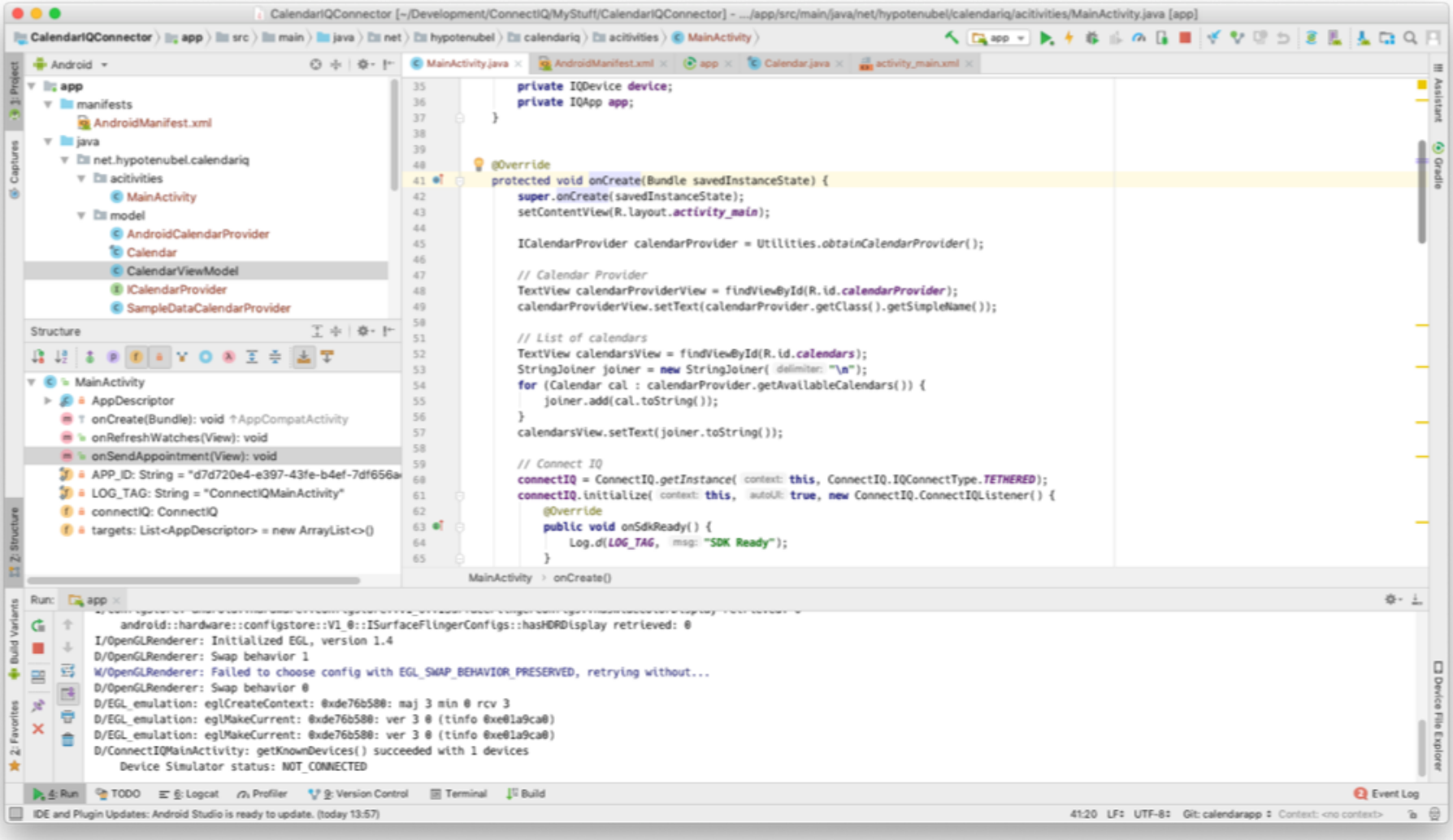
Wikipedia Example

Java Swing



JGoodies

Java Swing



IntelliJ IDEA

Java Swing

Successor to AWT.

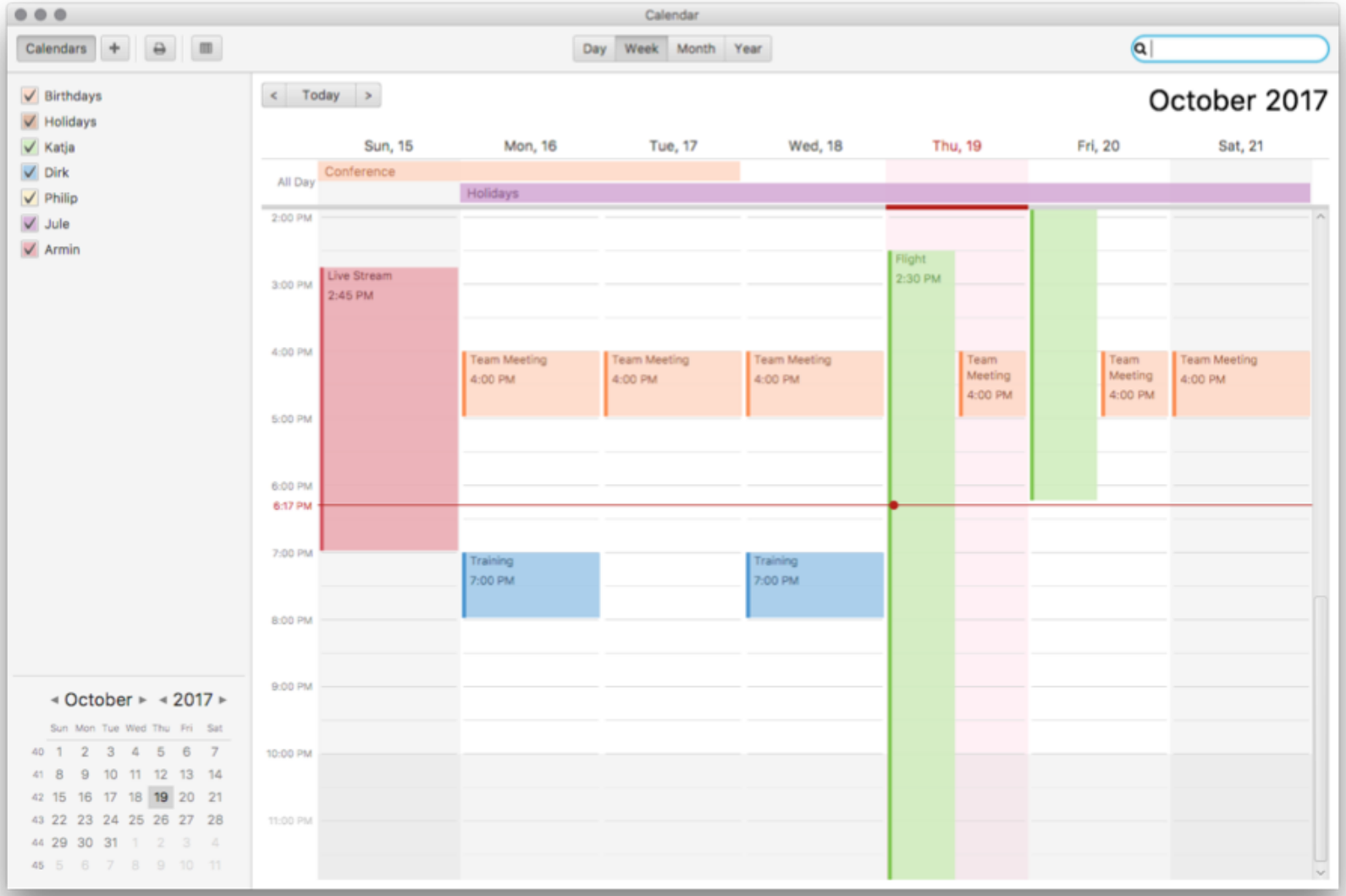
Rich in features and 3rd party support.

Work required to make it look really good.

Verdict

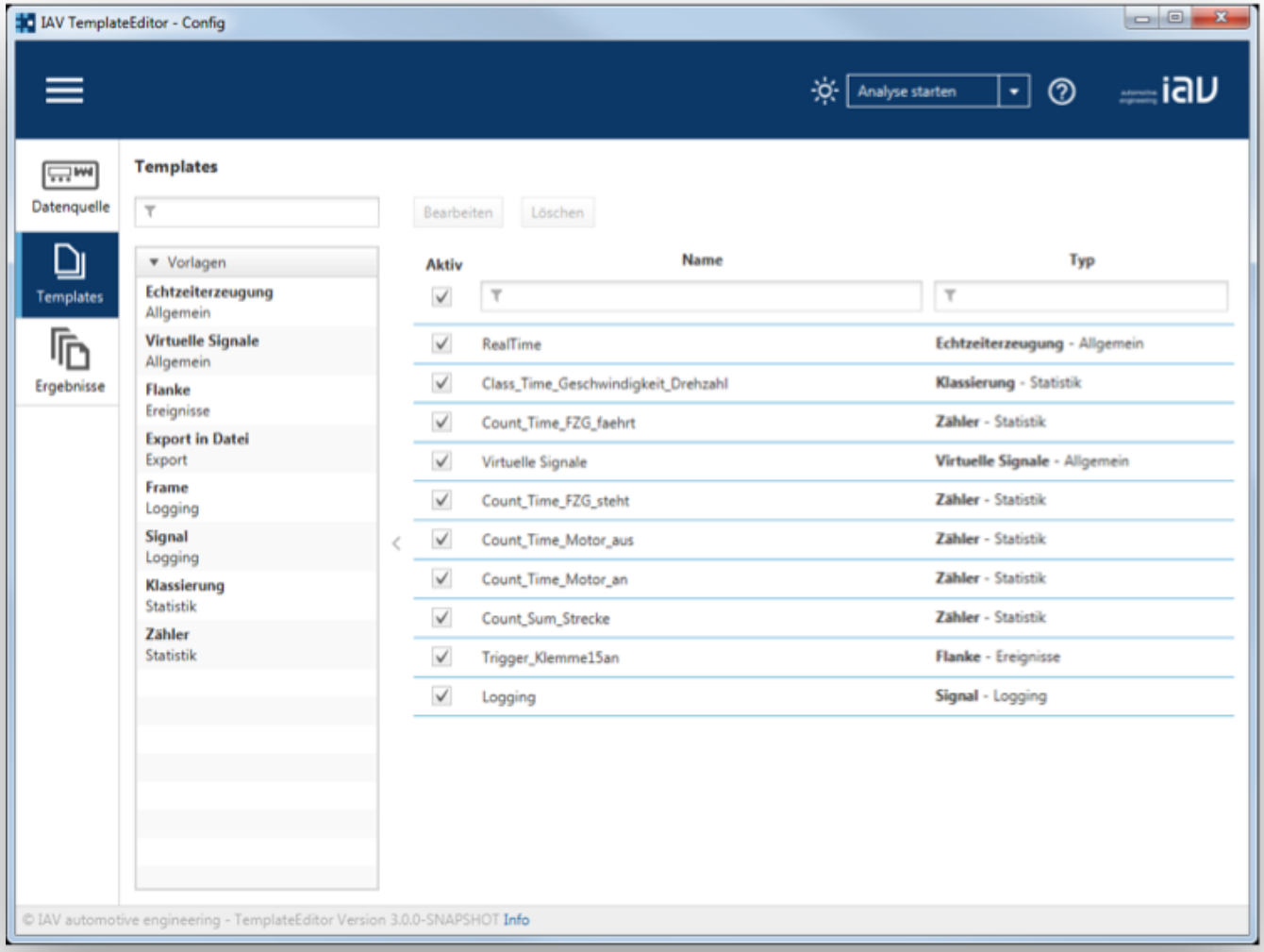
The simplest choice for Java GUIs.

JavaFX



CalendarFX

JavaFX



IAV TemplateEditor

JavaFX

Modern, beautiful GUIs.

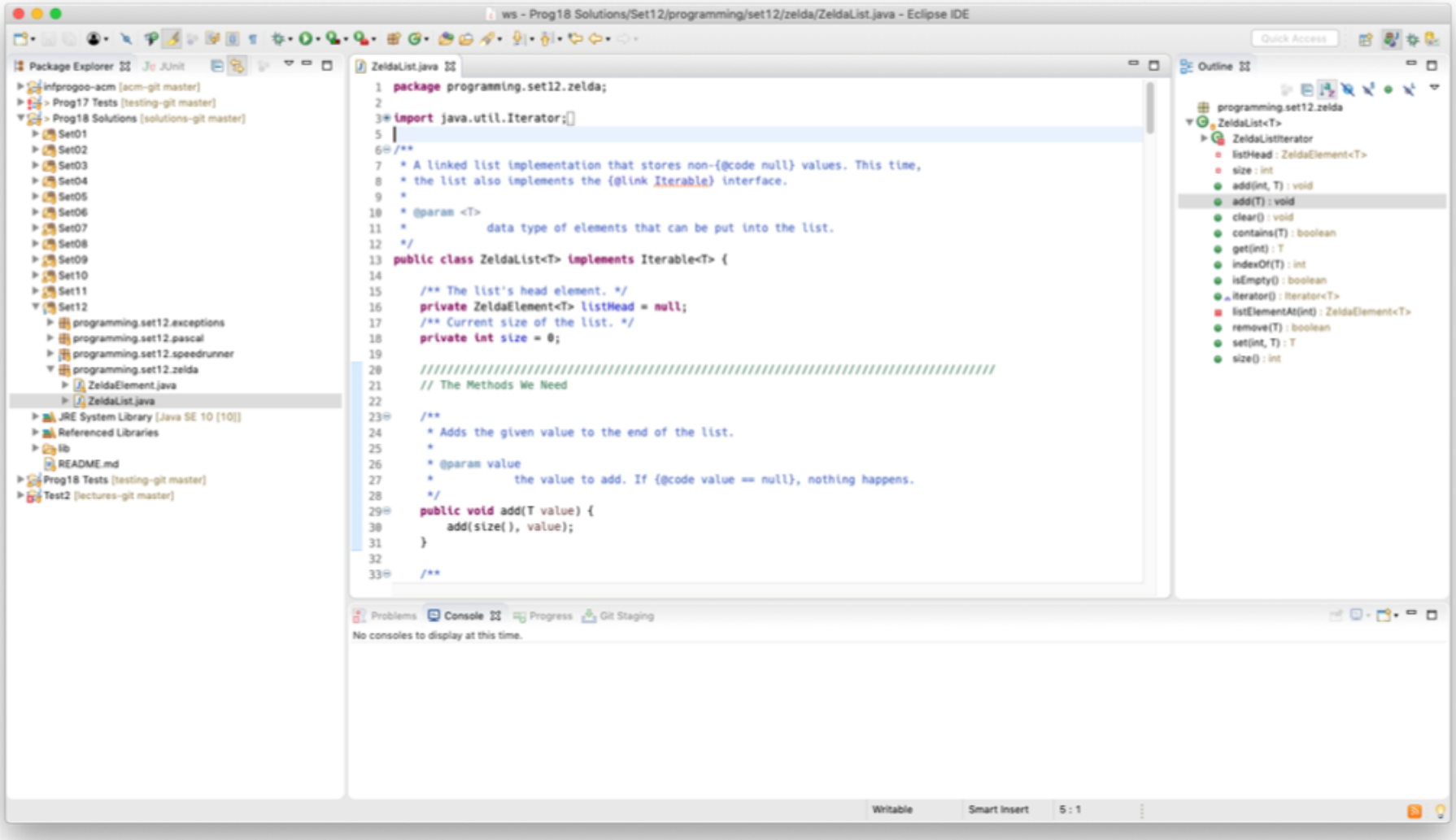
Easier to work with than Swing.

Since removed from JDK,
but still exists as OpenJFX.

Verdict

Nicer GUIs, harder to setup.

SWT


ws - Prog18 Solutions/Set12/programming/set12/zelda/ZeldaList.java - Eclipse IDE
Package Explorer
ZeldaList.java
Outline
Problems
Console
Progress
Git Staging
No consoles to display at this time.
Writable
Smart Insert
5 : 1

Eclipse

SWT

Platform-specific controls.

Feels fast and integrates well with the OS.

Perhaps a bit more difficult to use.

Verdict

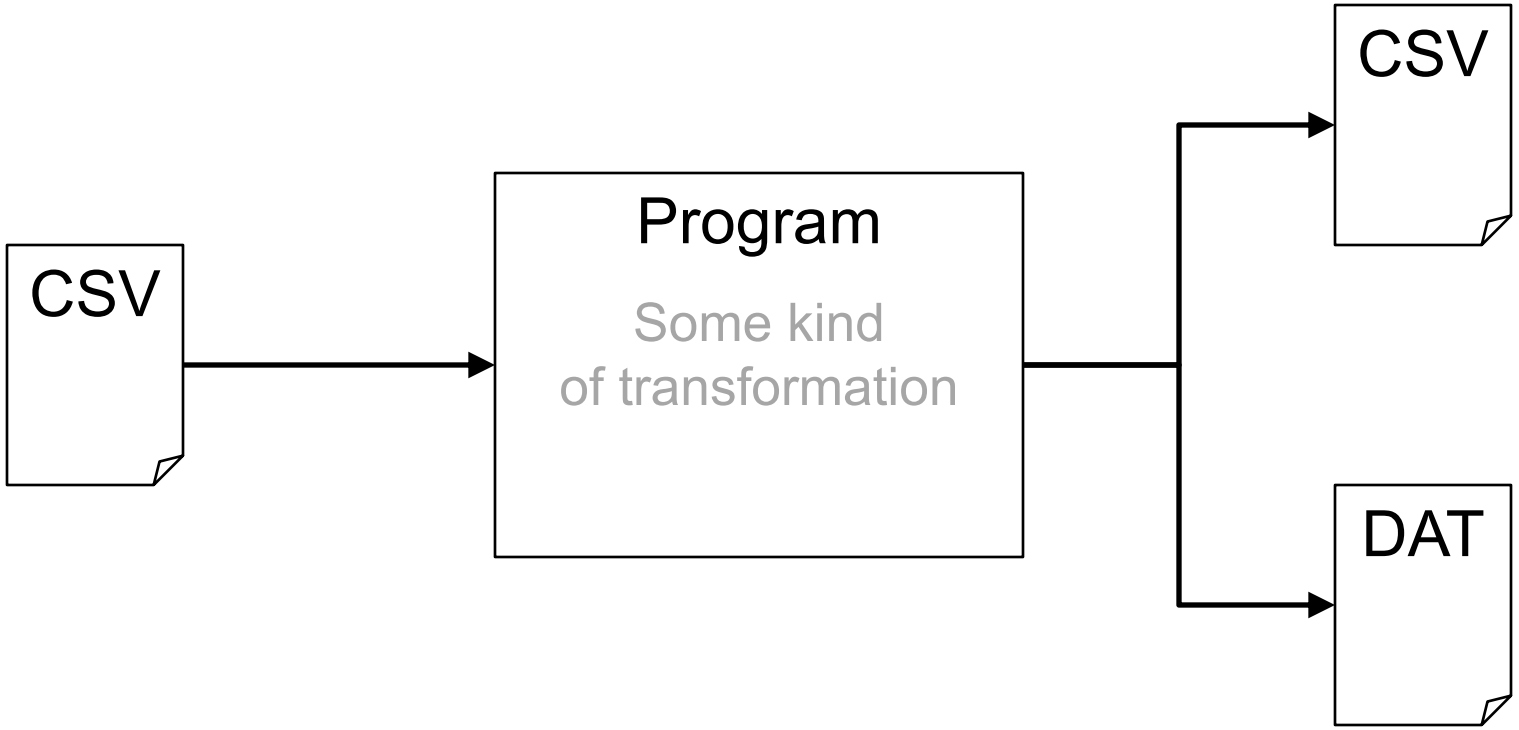
Native feeling, harder to setup.

Programming – Lecture 15

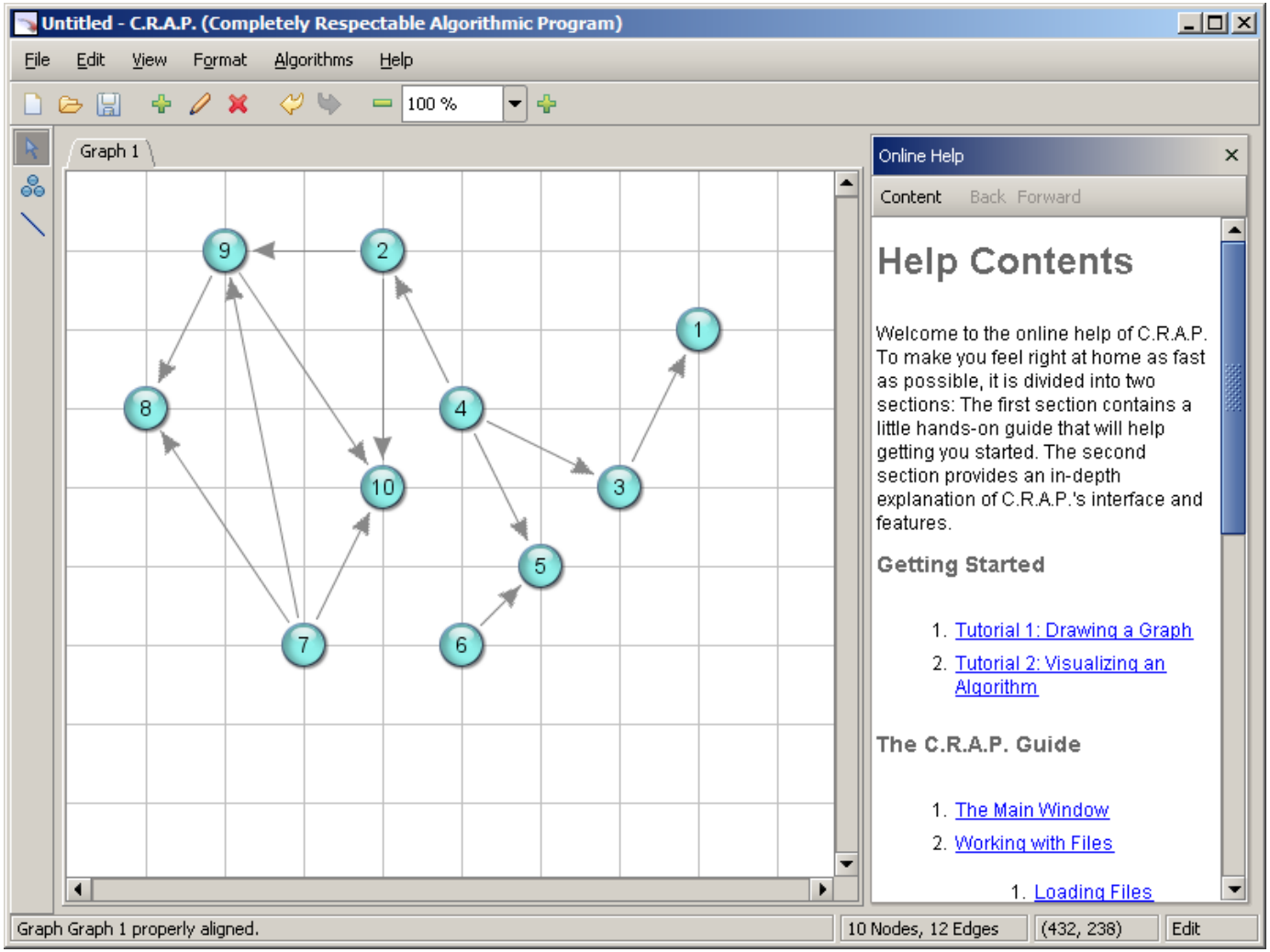
Going Beyond ACM

- Lectures Beyond Inf-ProgOO
- IDEs Beyond Eclipse
- Projects Beyond Homework
- Java Beyond ACM
- Programming Beyond Inf-ProgOO

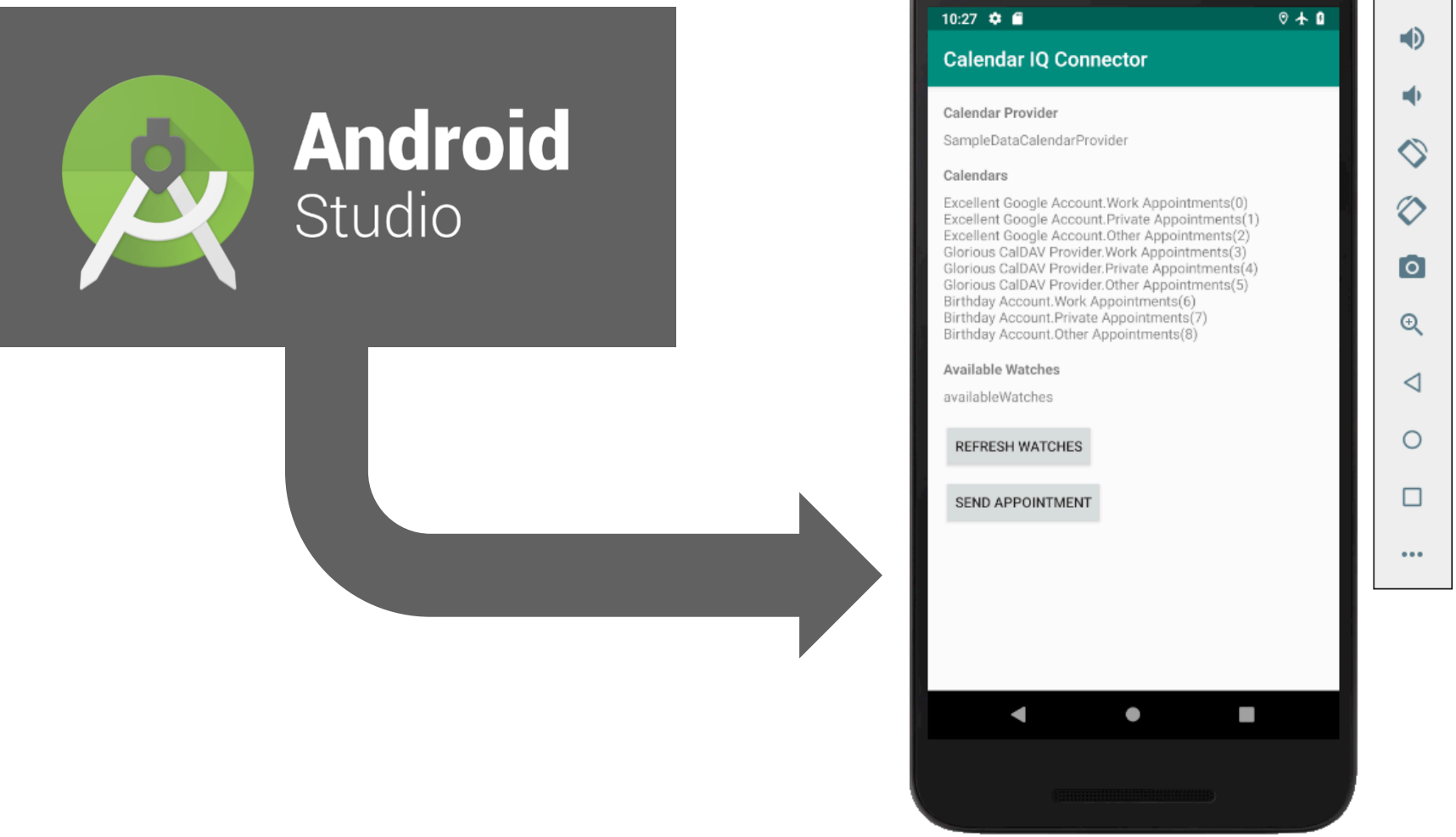
Small Console Programs



Something You're Missing



An Android App



I’m Fresh Out Of Ideas ☹

Thou shalt be helped!

Open Campus

Coworking space

Fablab

Waterkant

Toppoint

Mit Leuten und Werkzeugen
Dinge tun!

Raceyard

Programming – Lecture 15

Going Beyond ACM

- Lectures Beyond Inf-ProgOO
- IDEs Beyond Eclipse
- Projects Beyond Homework
- Java Beyond ACM
- Programming Beyond Inf-ProgOO

Don’t Be Afraid

We have tried to teach you concepts using Java as our example language.

Now go forth
and learn new languages!

Which Language to Learn?

The Two Main Questions

1

What do I want to build?

2

Where should it run?

Android Development

Kotlin

Java

iOS and Mac Development

Swift

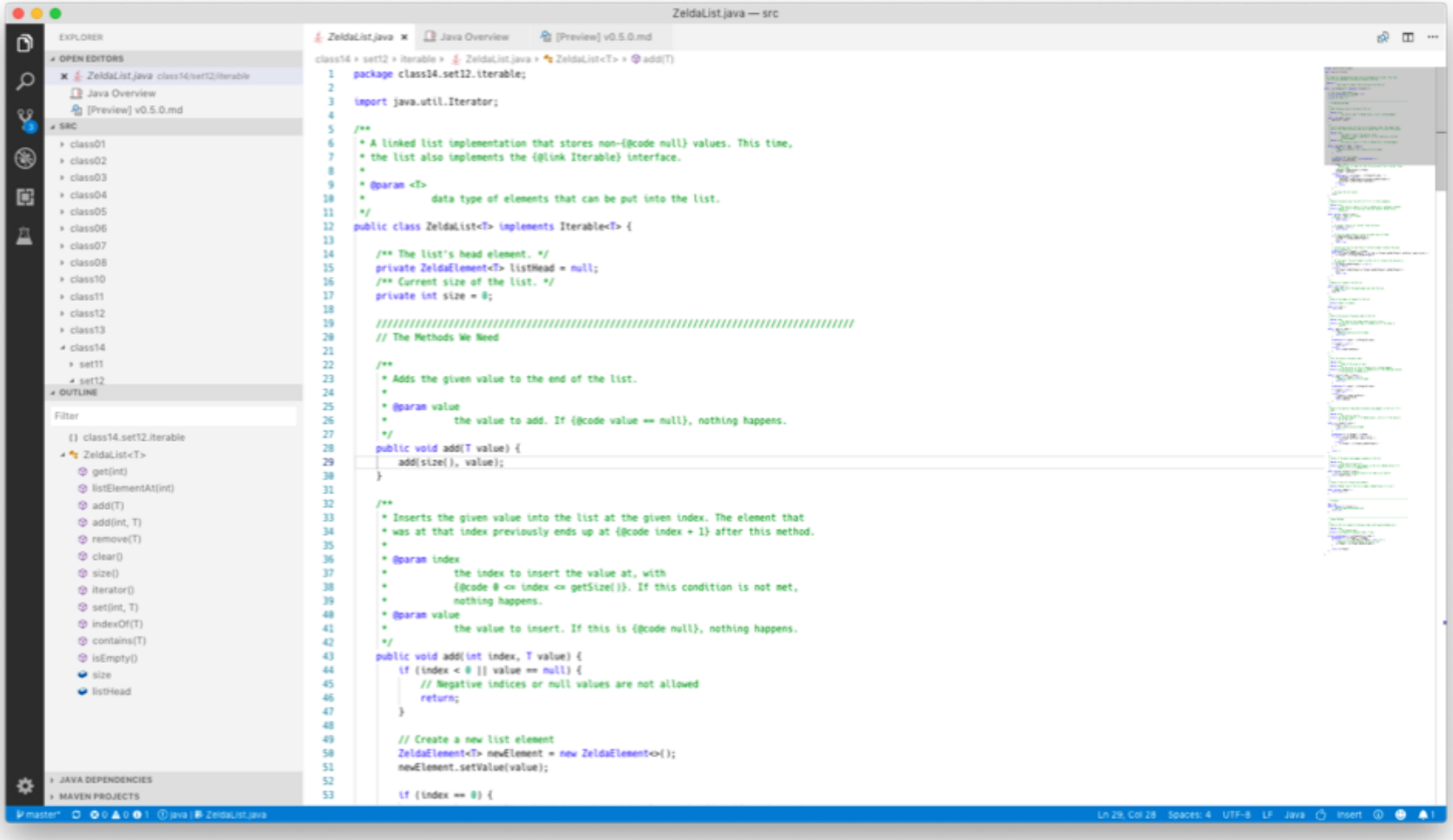
Web Development

JavaScript

Ruby

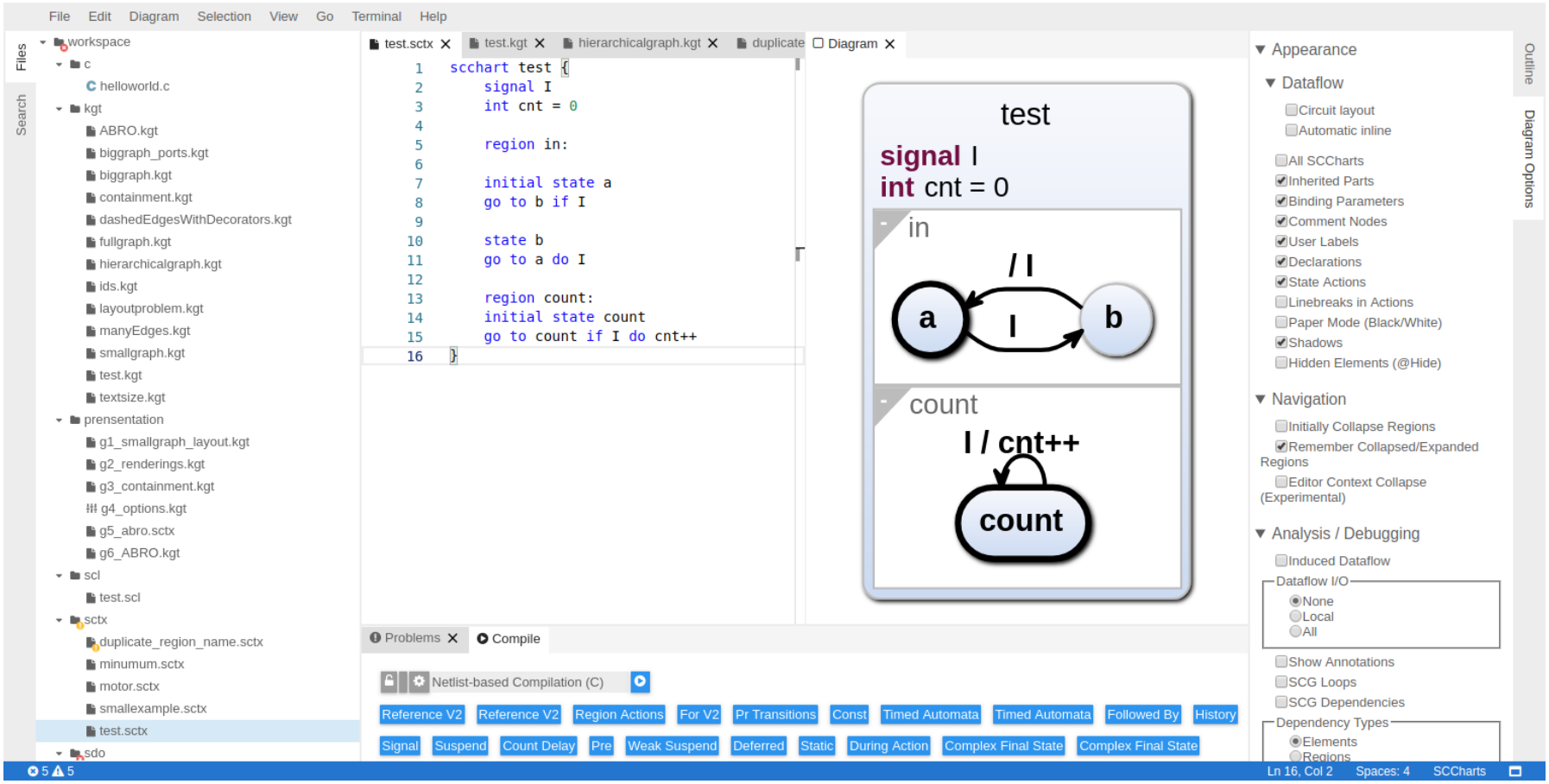
PHP

Web Development



Visual Studio Code

Web Development



Theia

Command-Line Tools

Python

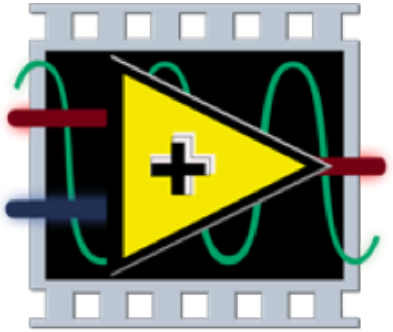
Perl

Ruby

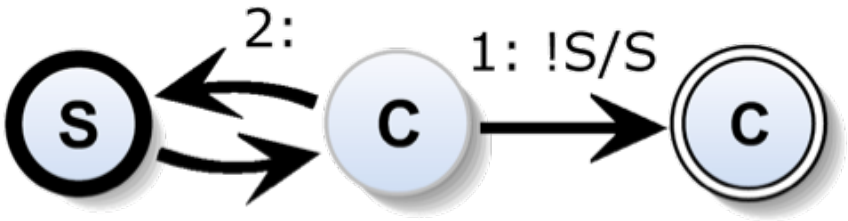
Tinkering With Hardware

C / C++

Visual Languages

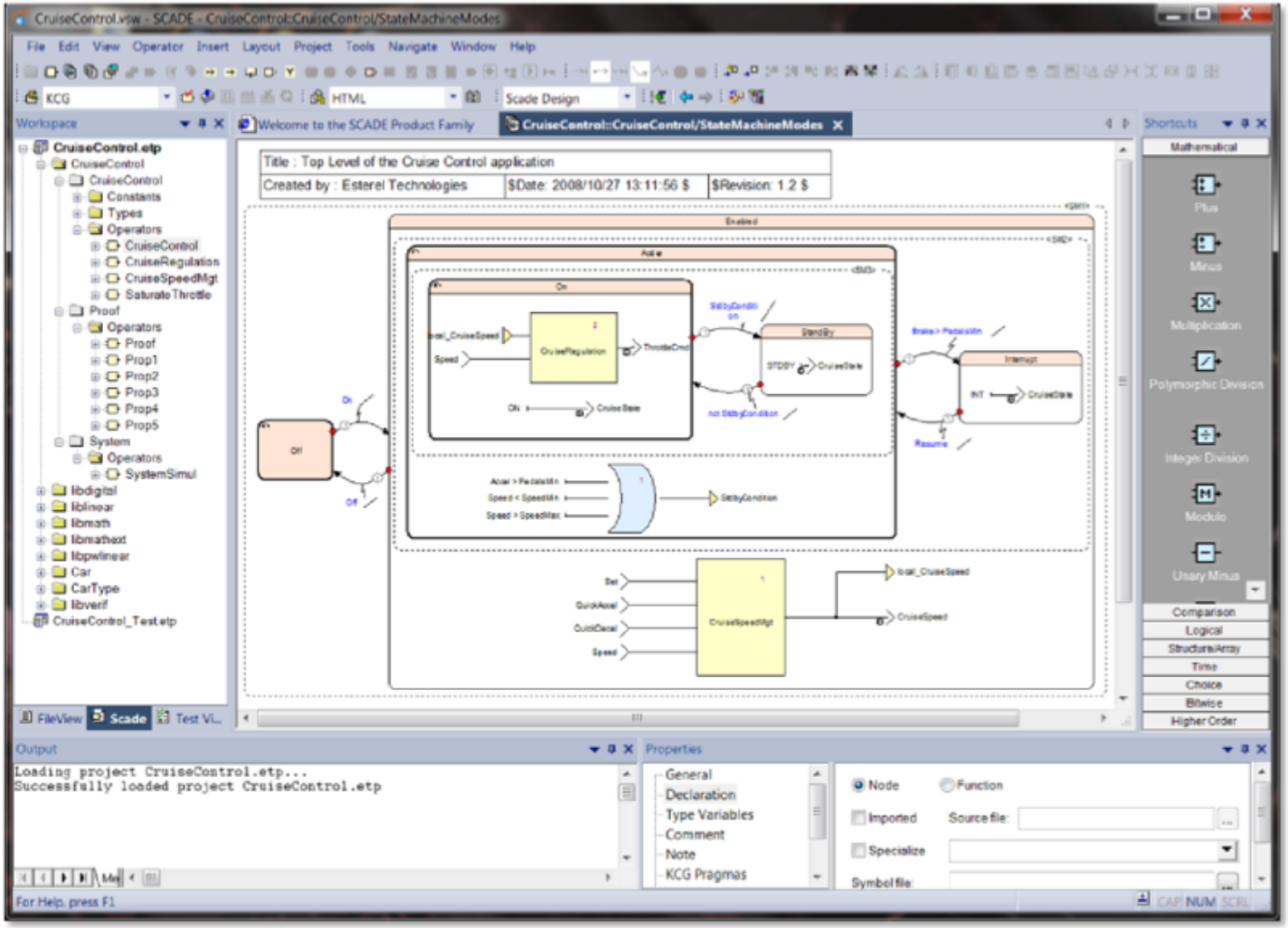


LabVIEW



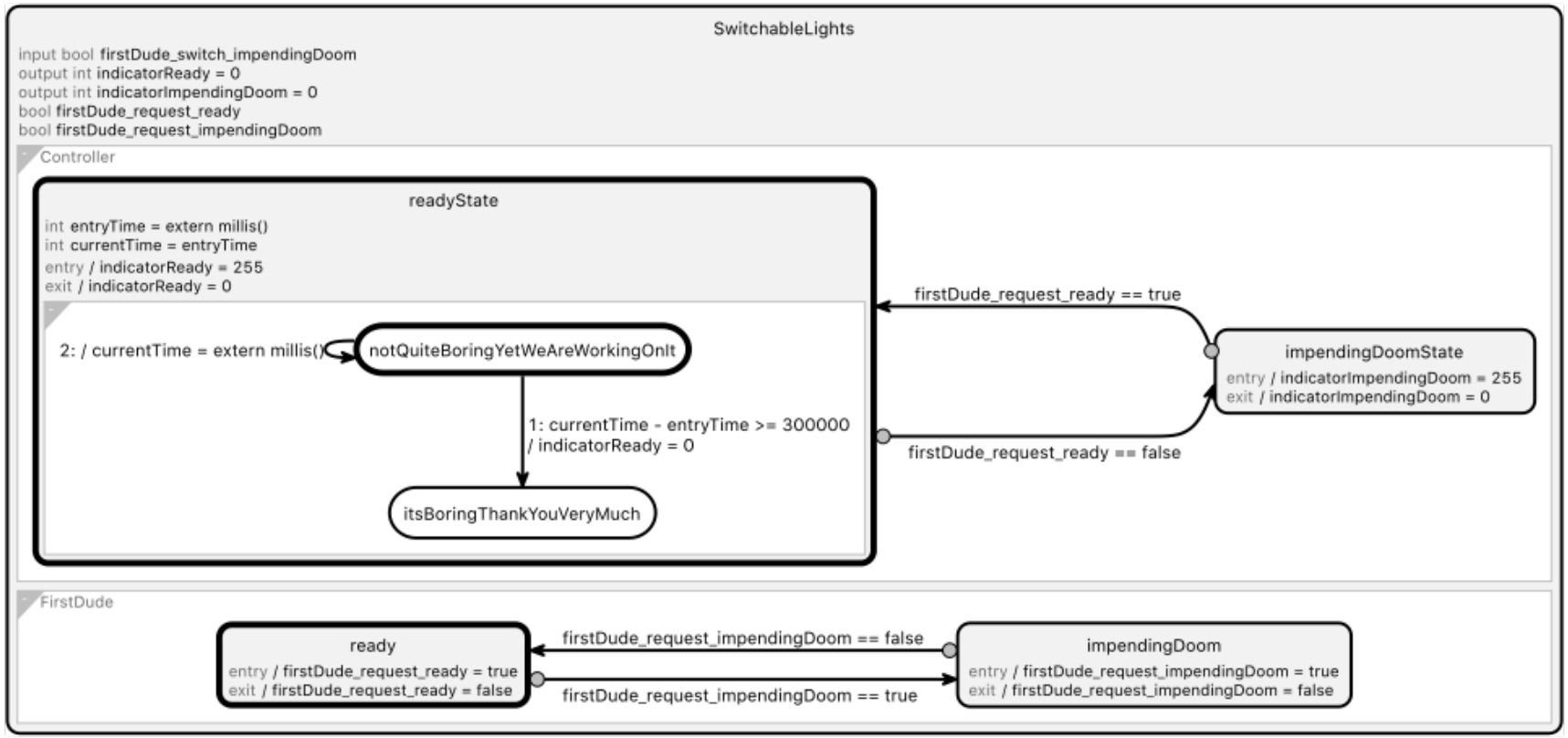
SCCharts

Visual Languages



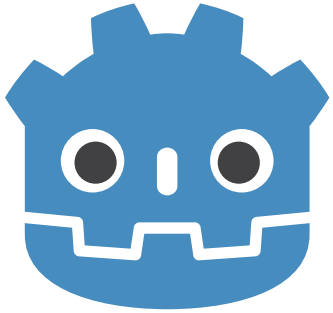
SCADE

Visual Languages

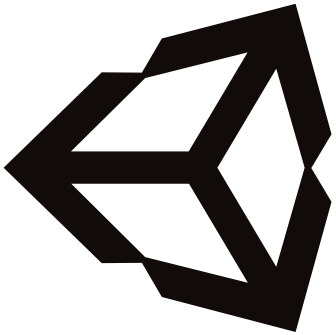


SCCharts

Games



Godot



Unity

You're just at the start
of your journey.

Now go and enjoy
being programmers!