Cornell University
Computing and Information Science

CS 5150 Software Engineering
17. Program Development

William Y. Arms

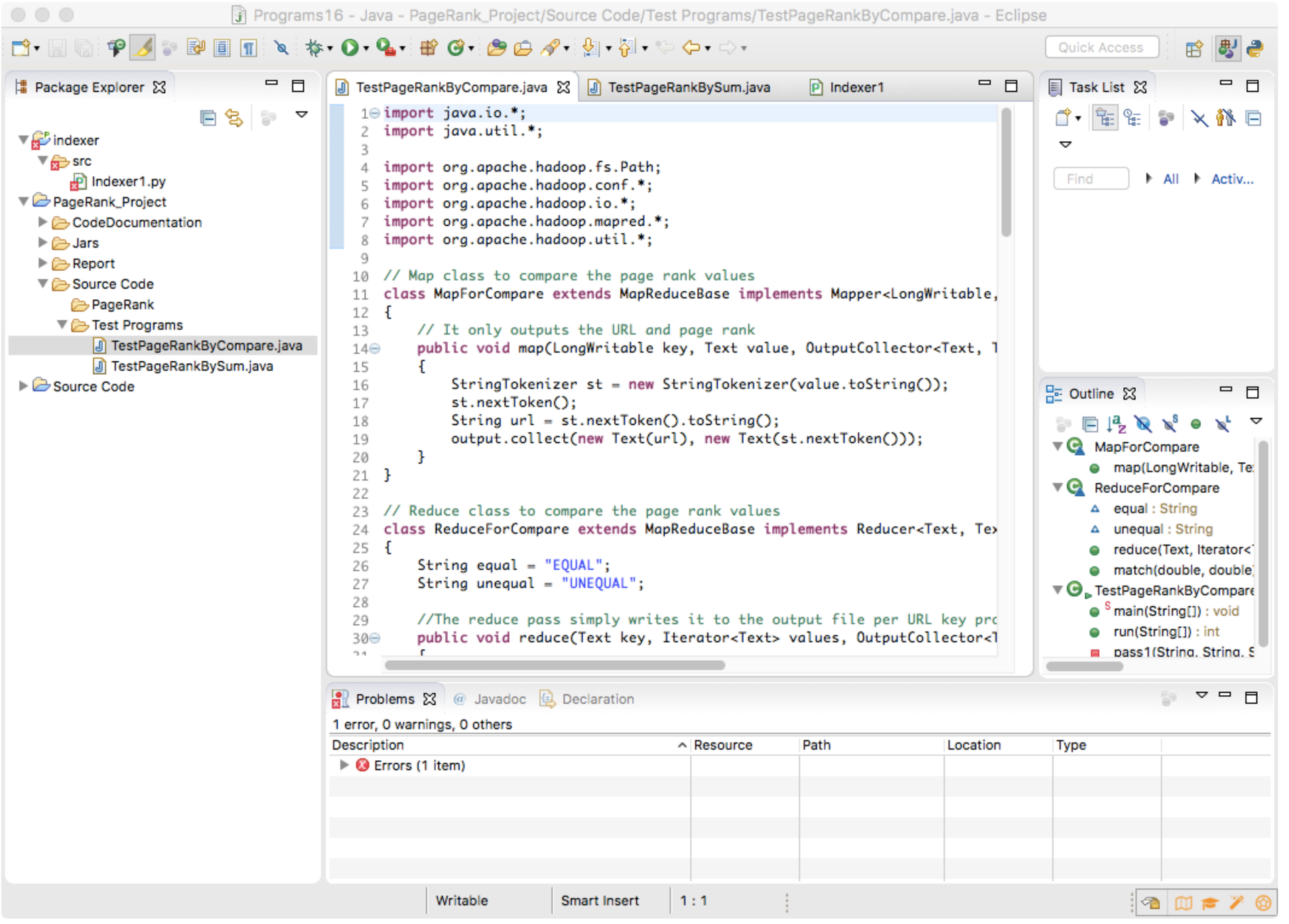
Integrated Development Environments

Basic software development requires:

- text editor (e.g., vi editor for Linux)
- compiler for individual files
- build system (e.g., make for Linux)

Integrated development environments combine:

- source code editor
- incremental compiler
- build automation tools
- a debugger
- and much, much more


Programs16 - Java - PageRank_Project/Source Code/Test Programs/TestPageRankByCompare.java - Eclipse
Quick Access
Package Explorer
indexer
src
Indexer1.py
PageRank_Project
CodeDocumentation
Jars
Report
Source Code
PageRank
Test Programs
TestPageRankByCompare.java
TestPageRankBySum.java
Source Code
TestPageRankByCompare.java
TestPageRankBySum.java
Indexer1
1 import java.io.*;
2 import java.util.*;
3
4 import org.apache.hadoop.fs.Path;
5 import org.apache.hadoop.conf.*;
6 import org.apache.hadoop.io.*;
7 import org.apache.hadoop.mapred.*;
8 import org.apache.hadoop.util.*;
9
10 // Map class to compare the page rank values
11 class MapForCompare extends MapReduceBase implements Mapper<LongWritable,
12 {
13 // It only outputs the URL and page rank
14 public void map(LongWritable key, Text value, OutputCollector<Text, T
15 {
16 StringTokenizer st = new StringTokenizer(value.toString());
17 st.nextToken();
18 String url = st.nextToken().toString();
19 output.collect(new Text(url), new Text(st.nextToken()));
20 }
21 }
22
23 // Reduce class to compare the page rank values
24 class ReduceForCompare extends MapReduceBase implements Reducer<Text, Te
25 {
26 String equal = "EQUAL";
27 String unequal = "UNEQUAL";
28
29 //The reduce pass simply writes it to the output file per URL key prc
30 public void reduce(Text key, Iterator<Text> values, OutputCollector<T
Task List
Find
All
Activ...
Outline
MapForCompare
map(LongWritable, Te
ReduceForCompare
equal : String
unequal : String
reduce(Text, Iterator<
match(double, double
TestPageRankByCompare
main(String[]) : void
run(String[]) : int
pass1(String, String, S
Problems
Javadoc
Declaration
1 error, 0 warnings, 0 others
Description
Resource
Path
Location
Type
Errors (1 item)
Writable
Smart Insert
1 : 1

Integrated Development Environment: Eclipse

Eclipse is a modern integrated development environment. It was originally created by IBM's Rational division. There are versions for many languages including Java, C/C++, Python, etc.

The Java system provides:

- source code editor
- debugger
- incremental compiler
- programming documentation
- build automation tools
- version control
- XML editor and tools
- web development tools

Much more is available via plug-ins.

Program Design: Integrated Development Environment

Integrated development environments provide little help in designing a program.

They assume that you have already have a design:

- classes
- methods
- data structures
- interfaces

Options for program design:

- program design using modeling tools, such as UML
- design while coding: design — code — redesign loop (small programs only)
- existing frameworks
- advanced environments that combine frameworks and development tools

It is often good to combine aspects of these different approaches.

The Design — Code — Redesign Loop

If the class structure is straightforward it may be possible to use the integrated development environment to:

- create an outline of the class structure and interfaces
- write code
- modify the class structure as needed and rework the code as necessary

This is only possible with small teams with close communication.

The maximum size of program depends on experience of programmer(s) and complexity of the program.

It may be possible to complete a a single agile sprint.

Eventually the amount of rework becomes overwhelming.

Class Hierarchies

Since the design of class hierarchies is difficult it is good practice to use existing frameworks.

Often many of the classes will have been written for you, or abstract classes are provided that you can use as a basis for your own subclasses.

Examples:

- class hierarchies that are part of programming languages
- toolkits (e.g., for graphical user interfaces)
- design patterns
- frameworks for web development and mobile apps

Class Hierarchies: Programming Languages

Example: Java

Java is a relatively straightforward language with a very rich set of class hierarchies.

- Java programs derive much of their functionality from standard classes.
- Learning and understanding the classes is difficult.
- Experienced Java programmers can write complex systems quickly.
- Inexperienced Java programmers write inelegant and buggy programs.

Languages such as Java and Python steadily change their class hierarchies over time. Commonly the changes replace special purpose functionality with more general frameworks.

If you design your programs to use the class hierarchies in the style intended by the language developers, it is likely to help with long term maintenance.

Web Development Frameworks

A web development framework provides a skeleton for building web applications.

An early example was Cold Fusion, which implements a three tier architecture.

Modern example, such as Ruby on Rails and Django, often use a model-view-controller (MVC) architecture.

For example, Ruby on Rails provides:

- a database
- a web server
- web pages

It is intended to be used with web standards, e.g., XML , HTML, CSS, and JavaScript.

Web Development Frameworks: Django

Django is a Python framework for developing web sites.

- loosely based on MVC architecture
- supports a variety of web and database servers
- web template system
- authentication system
- administrative interface
- mitigation of web attacks

Django is a complex framework. Teams should allow plenty of time for learning.

Frameworks for Responsive Web Design: Bootstrap

William Arms | Career summary | Professional activities | Publications | Contact

Computing and Information Science
Cornell University

William Y. Arms

Professor Emeritus
Computing & Information Science
Cornell University

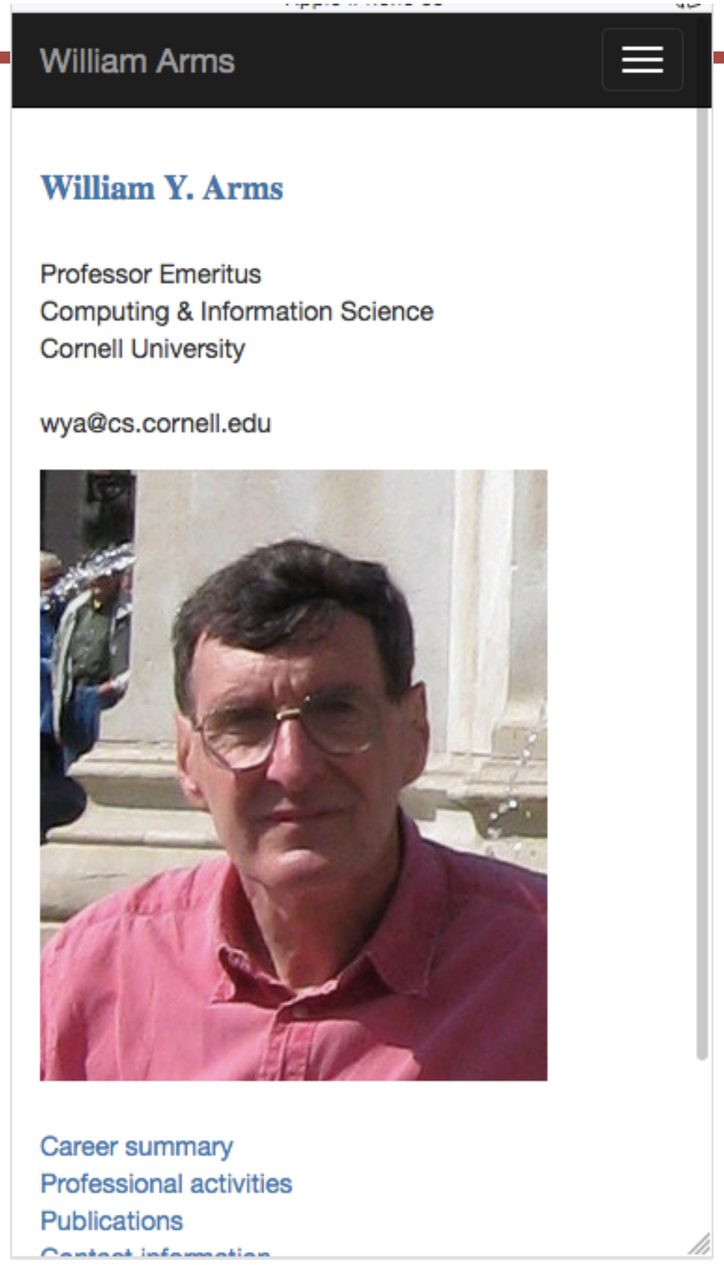
wya@cs.cornell.edu

Career summary
Professional activities
Publications
Contact information

Digital Libraries
The Early Years of Academic Computing

Last changed: September 2017

Frameworks for Responsive Web Design: Bootstrap



CSS media queries are a powerful tool for responsive web design, but complex to use well.

Frameworks, such as Bootstrap, provide simple ways to build responsive web sites.

This slide shows a web page displayed on an iPhone. Compare it with the previous slide, which shows the same page in a window on a laptop.

Advanced Development Environments

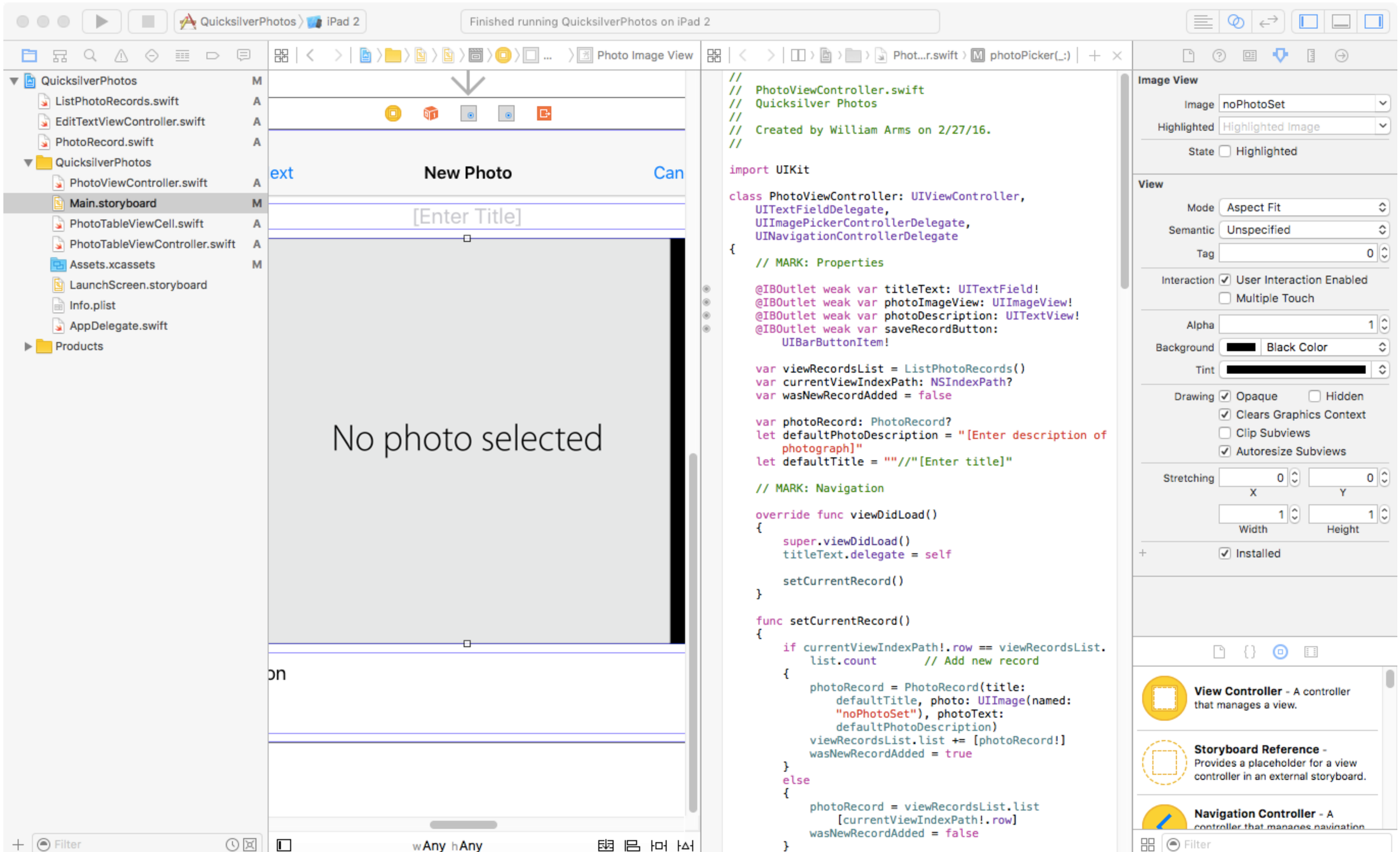
Application frameworks can be used with any program development environment, e.g., Django and Eclipse (Python version)

An advanced development environment combines:

- integrated development environment (IDE)
- application framework
- user interface layout manager
- and more

Example:

Apple's Xcode for iOS


QuicksilverPhotos
iPad 2
Finished running QuicksilverPhotos on iPad 2
QuicksilverPhotos M
ListPhotoRecords.swift A
EditTextViewController.swift A
PhotoRecord.swift A
QuicksilverPhotos
PhotoViewController.swift A
Main.storyboard M
PhotoTableViewCell.swift A
PhotoTableViewController.swift A
Assets.xcassets M
LaunchScreen.storyboard
Info.plist
AppDelegate.swift
Products
Photo Image View
New Photo
[Enter Title]
No photo selected
wAny hAny
Phot...r.swift
photoPicker(_:)
//
// PhotoViewController.swift
// Quicksilver Photos
//
// Created by William Arms on 2/27/16.
//
import UIKit
class PhotoViewController: UIViewController,
UITextFieldDelegate,
UIImagePickerControllerDelegate,
UINavigationControllerDelegate
{
// MARK: Properties
@IBOutlet weak var titleText: UITextField!
@IBOutlet weak var photoImageView: UIImageView!
@IBOutlet weak var photoDescription: UITextView!
@IBOutlet weak var saveRecordButton:
UIBarButtonItem!
var viewRecordsList = ListPhotoRecords()
var currentViewIndexPath: NSIndexPath?
var wasNewRecordAdded = false
var photoRecord: PhotoRecord?
let defaultPhotoDescription = "[Enter description of
photograph]"
let defaultTitle = ""//"[Enter title]"
// MARK: Navigation
override func viewDidLoad()
{
super.viewDidLoad()
titleText.delegate = self
setCurrentRecord()
}
func setCurrentRecord()
{
if currentViewIndexPath!.row == viewRecordsList.
list.count // Add new record
{
photoRecord = PhotoRecord(title:
defaultTitle, photo: UIImage(named:
"noPhotoSet"), photoText:
defaultPhotoDescription)
viewRecordsList.list += [photoRecord!]
wasNewRecordAdded = true
}
else
{
photoRecord = viewRecordsList.list
[currentViewIndexPath!.row]
wasNewRecordAdded = false
}
Image View
Image noPhotoSet
Highlighted Highlighted Image
State Highlighted
View
Mode Aspect Fit
Semantic Unspecified
Tag 0
Interaction User Interaction Enabled
Multiple Touch
Alpha 1
Background Black Color
Tint
Drawing Opaque Hidden
Clears Graphics Context
Clip Subviews
Autoresize Subviews
Stretching 0 X 0 Y
1 Width 1 Height
Installed
View Controller - A controller that manages a view.
Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.
Navigation Controller - A

Advanced Development Environments

An advanced development environment is intended to provide everything that a developer needs.

The developer is expected to follow the program choices that are provided.

For example, when Xcode is used with iOS it has a very specific purpose: mobile apps for Apple devices such as iPhones, iPads.

- Special programming language (Swift or Objective C)
- MVC framework

If you accept the overall program design it is very powerful:

- Auto layout of graphical interfaces
- Comprehensive set of classes for user interfaces and navigation
- Simulators for all Apple devices

Using Development Frameworks

Development frameworks are powerful and flexible.

If your application fits the framework, they do much of the program design and provide high quality code for many of the standard parts of any application.

Some parts of the application may need be designed separately.

But beware:

- You are forced to build your application within the framework that is provided.
- The frameworks are continually modified.
- These frameworks are complex and take a long time to learn.

Production Programming

Murphy's Law:

If anything can go wrong, it will.

Challenges:

- Code has to be maintained over the long term, with different system software.
- Interfaces will be used in new and unexpected ways.
- Every possible error will eventually occur at the worst possible time (bad data, failures of hardware and system software).
- There are likely to be security attacks.

Production Programming

Robust programming:

- Write simple code.
- Avoid risky programming constructs.
- If code is difficult to read, rewrite it.
- Incorporate redundant code to check system state after modifications.
- Test implicit assumptions explicitly, e.g., check all parameters received from other routines.
- Eliminate all warnings from source code.
- Have a thorough set of test cases for all your code.

In a production environment, expect to spend longer on coding and testing than in an academic setting.