

Java Users Group Annual Review & Planning Meeting

July 11, 2017

Topics for 2017-2018

[Topic to be Determined]

- presented by Venkat Subramaniam
- March 19-20, 2018
- final arrangements with NYJavaSIG to be determined
- see below for titles and abstracts

JavaEE 8

- presented by Reza Rahman
- e-mail sent to Reza on July 11, 2017

Containers 101

- presented by Docker via webinar
- Mike to contact Docker

Microservices

- presented by Lightbend (via webinar) or Mike Redlich

Doug Crockford JavaScript

- presented by Barry Burd (as suggested by Bill Brutzman)
- <http://javascript.crockford.com/>

node.js

- presented by Scot Jenkins (as suggested by Bill Brutzman)

Functional JavaScript

- presented by Barry Burd (as suggested by Bill Brutzman)

Android Beacons

Pure Functional Programming for Android

- presented by Barry Burd
- September 12, 2017 (tentative)

Swift

- presented by Barry Burd
- September 12, 2017 (tentative)

Java 9 Modules

- presented by Mike Redlich

Kotlin

- presented by Mike Redlich and/or Barry Burd

Convention for Java Application Development

- presented by Bruce Spilker
- October 10, 2017

Game Development

- Architecture
- Rendering
- speaker to be determined

Java 8 Streams by Puzzle

Java for AWS Lambda Functions

- presented by Jeanne Boyarsky
- Barry to contact Jeanne

Titles & Abstract Offered by Venkat Subramaniam

Twelve Ways to Make Code Suck Less

We all have seen our share of bad code and some really good code as well. What are some of the common anti patterns that seem to be recurring over and over in code that sucks? By learning about these code smells and avoiding them, we can greatly help make our code better. Come to this talk to learn about some common code smell and how to improve the quality of code.

Designing Functional Programs

Functional Programming promotes immutability and the use of higher order functions. For those of us who have designed and architected applications using imperative style of programming and the object-oriented paradigm, this largely appears like a strange idea. We often ask, how is it practical to apply these ideas, realistically to build practical applications. It turns out it's highly practical but we have to change how we design and how we model our systems. Come to this presentation to think about functional style and how to start viewing design to make better use of this way of programming.

Java 8 Programming Idioms

A number of developers and organizations are beginning to make use of Java 8. With anything that's new, we often learn it the hard way. By stepping back and taking a look at programming style as idioms, we can quickly gravitate towards better coding style and also avoid some common traps that we often get drawn towards.

The Power and Perils of Parallel Streams

"If streams can be parallel, why not make them parallel all the time?" is a common question from developers getting introduced to Java 8 streams. In this talk we'll take on three separate topics. 1. When to consider parallelization and when not to. 2. How to parallelize, how to decide on number of threads, and how to control the threads pool. 3. Learn about some common mistakes people make when using parallel streams. The goal of this talk is for us to learn when and how to make good use of parallel streams.

The Power and Practicality of Immutability

Functional Programming promotes assignment-less programming. When we remove mutability, we reduce errors in code. How practical is it to really program with immutability in Java and if we manage to achieve that, what are the real benefits. Come to this talk to learn about the power, the benefits that immutability brings and how to practically make use of that in Java.

From Functional to Reactive Programming

We're in the midst of renewed interest in functional programming. At the same time we see quite a bit of excitement around reactive programming. Where did reactive programming come from? How is it related to functional programming, if at all? In this presentation we will discuss the merits of reactive programming and how functional programming concepts seamlessly transition into the programming model espoused by reactive programming.

Exploring Java 9

If Java 8 was all about how we code, Java 9 is all about how we will build. Modularization will have the biggest impact of any change that happened in Java since its inception. In this presentation we will learn about the need for modularization, how it impacts development, the rules to follow when creating modules, and the effect it has on legacy code. We will explore creating module, using modules, readability, exports, automatic modules, and unnamed modules.