Surviving Eclipse Collections

Twenty Years of Software, Content, and Community Development

Donald Raab, JCP EC Meeting, August 13, 2024

About me

Programmed in Java since 1997

- Creator, Project Lead, Committer, Contributor for Eclipse Collections for over 20 years
  - Soon to be Eclipse Collections book author! Stay tuned!
- Java Champion (since 2018)
- Previously a JCP EC Primary Representative
- Member of the JSR 335 Expert Group (Lambdas, Streams)
- Contributing Author, “97 Things Every Java Programmer Should Know" - O'Reilly
- Twitter/X: @TheDonRaab
- Mastodon: @TheDonRaab@mastodon.social
- Medium: https://donraab.medium.com
- GitHub: https://github.com/donraab

Evolution of Eclipse Collections

Twenty Years (2004 - 2024)

- Eclipse Collections
  - 2015 - 2024 - Community-driven OSS at Eclipse Foundation
    - Top 5 Project on Eclipse Foundation GitHub Account
  - 6 Committers, 100+ Contributors
  - 800,000+ downloads from Maven Central in July 2024
- GS Collections
  - 2012 - 2016 - Goldman Sachs OSS on GitHub
- Caramel
  - 2004 - 2012 - Goldman Sachs Proprietary
- Who is currently involved with Eclipse Collections?

eclipse collections
GS Collections
Caramel

Planning for Survival

is not the same as planning for success

- Success is goal oriented
    - Keys to success -- Passion, Patience, Persistence
- Survival is journey oriented
    - Keys to survival -- People, Purpose, Progress, Perseverance
- Planning a long journey
    - Think in decades
    - Reflect on goals
    - Grow the Community
    - Celebrate wins
    - Prioritize self care
    - Look for forks in the road
    - Let it grow
- From Surviving Open Source Keynote

A Five Year "Plan"

Wishlist for community contributions

Technical Debt	Java Upgrades	New Containers / API	IDE Refactoring Support
• Combine Unit Tests and Unit Tests Java 8 modules • Replace Scala Unit Tests with Java Equivalent Tests • Remove JMH Scala Tests • Refresh and update the Eclipse Collections Reference Guide • Improve JavaDoc • Move slow running unit tests to acceptance tests • Replace legacy anonymous inner classes with lambdas and method references • ~~Clean up performance tests~~	• ~~Upgrade library to Java 11~~ • ~~Leverage Java Module System fully~~ • Leverage Local Variable Type Inference sparingly to improve readability • Test and integrate with Project Valhalla features • *Upgrade to JUnit 5 (In Progress)*	• ~~DataFrames~~ • Trees • More primitive collection types • Lazy Collections (specific types like LazyList) • Off-heap Collections • Persistent Collections (Functional) • Improve symmetry between object and primitive APIs • More converter methods between types • Implement more Parallel APIs • *Optimize APIs inherited from JDK (In Progress)*	• Refactor from Java Streams to Eclipse Collections • Refactor from for-loops to Eclipse Collections • Refactor from Object collections to Primitive Collections

Challenges

Thoughts from a long time open source maintainer

- People
  - Have different needs and lose motivation when their interests are not aligned.
  - Challenging to grow the number of committers. Meritocracy requires investment.
- Purpose
  - Goal posts move. You need to move with them. Sponsorship and dedication helps.
  - Tech upgrades require massive investment and provide “invisible” value (e.g. JUnit 4 -> JUnit 5, JPMS, OSGi, Java ##)
- Progress
  - Development is easy and can be done by anyone.
  - Releases are hard and require key people.
- Perseverance
  - Two other active committers have contributed for over a decade.
  - I’m writing a book after twenty years of coding on the library. Hoping one day to find a replacement for my role on the library.

Foundations are not Fundations

Corporate Philanthropy funds Foundations not Developers

- Community-driven projects can benefit from corporate sponsorship
  - Funding internal corporate developers (time/dollars)
  - Funding community developers (dollars)
- How to fund community developers is challenging
  - Commonhaus Foundation is exploring funding solutions

Open Discussion: Why invest in OSS Communities?

- Benefits of the continued evolution of Eclipse Collections
    - Active contributor to OpenJDK Quality Outreach Program (QOP)
        - ~170,000 unit tests running against JDK 11, 17, 21, 22, 23EA on each PR and multiple issues have been discovered and reported over the years
        - Blog posts encouraging open source community participation in OpenJDK QOP
    - Used by other open source and commercial software
        - e.g. Neo4J, BigQuery, AtlasDB, HiveMQ, HugeGraph, SquashQL, IoTDB, RSocket Broker Spring Boot Starter, GraphDB, Legend, DataFrame-EC, and more
    - Helps test, drive, and validate innovation in the OpenJDK
        - e.g. Lambdas/Streams, RandomAccessSpliterator, Empty List lazy array initialization optimization