

Name : Devinder Singh

Father name : Parbat Singh

RollNO : 2K18/CSME/08

Course title :- Software Engineering

Teacher : Dr. Gulsher Laghari

Department : IMCS

University of Sindh
Jamshoro



Using Large - Scale Anomaly detection on code to Improve ~~Kot~~ Kotlin Compiler

Author :- Timofey Bryksin from
jetbrains Research Saint Petersburg
State University and etc.

Date :- 29-06-2020

Anomaly detection we apply for
source code and bytecode to
facilitate development of a programming
Language and its compiler. We
define anomaly as a code fragment
that is different from typical code
written in a particular programming
Language. Code fragments is
beneficial for both Language
developers and end users.

Anomalies may indicate potential
issues with the compiler or with
runtime performance. Anomalies
could corresponding to problems
in design Language. For this
study, they choose Kotlin as

the target programming language.
The paper presents a method that
aims to detect two types of
Anomalies: Syntax tree anomalies
and so called compiler-induced
anomalies that arise only in the
compiled bytecode.

Summary

An Empirical study of method chaining in java.

Author :- Tomoki Nakamaru from Graduate school of Information Science and Technology, the University of Tokyo and etc.

Date :- 29-06-2020

Some promote method chaining as a good practice for improving code readability, others refer to it as a bad practice that worsens code quality. In this paper, they first investigate whether method chaining is a programmers. For the answer this question, they collected 2,814 java repositories on GitHub and analyzed historical trends in the frequency of method chaining. The results of their analysis revealed the increasing use of method chaining; 23.1% of method invocations were part of

(4)

method chains in 2018, where as only 16.0% were such invocations in 2010. ~~the~~ then they explore language features that are helpful to the method chaining style but have not been supported yet in java. for this aim, they conducted manual inspections of method chains that are randomly sampled from the collected repositories. They also estimated how effective they are to encourage the method-chaining style if they are adopted in java.

On the Prevalence, Impact, and Evolution of SQL code smells in Data Intensive Systems.

Author: Alessandro Garcia, Biruk Asmare Muse and Masud Rahman.

Date:- 29-06-2020

Code smells means software designing problems that is harmful for software quality. In this paper, they conduct an empirical study to investigate the prevalence and evolution of SQL code smells in open source, data-intensive systems. They collected 150 projects and examined both traditional and SQL code smells in these projects. Their investigator delivers several important findings. First, SQL code smells are indeed prevalent in data-intensive software systems. Second, SQL code smells have a weak co-occurrence with traditional code smells. Third SQL code smells have a weaker association with bugs than that of traditional code

(6)

Smells. Fourth, SQL code smells are more likely to be introduced at the beginning of the project life time and likely to be left in the code without a fix, compared to traditional code smells. Overall, these results show that SQL code smells are indeed prevalent and persistent in the studied data-intensive software systems. Developers should be aware of these smells and consider detecting and refactoring SQL code smells and traditional code smells separately, using dedicated tools.