Обнаружение аномалий в программах на языке Kotlin методами статического анализа кода

Станислав Витальевич Приходько, 646 группа
Научный руководитель: к. т. н., доц. Т.А. Брыксин
Консультант: аналитик ООО ”Интеллиджей Лабс” Н.И. Поваров
Рецензент: инженер по тестированию ПО ООО ”Интеллиджей Лабс” В.А. Петухов

Санкт-Петербургский государственный университет
Кафедра системного программирования
2019

Введение

- Кодовая аномалия — нестандартный пример программного кода
- Обнаружение аномалий позволит:
  - Выявить нестандартное применение языковых конструкций
  - Провести тестирование компилятора
  - Выявить недостатки языка программирования
- Разработчики языка программирования Kotlin заинтересованы в поиске кодовых аномалий с целью улучшения экосистемы языка

Система обнаружения кодовых аномалий на языке программирования Kotlin

- Исследования лаборатории JetBrains Research
- Найдено 46 классов кодовых аномалий
- Часть найденных аномалий включена в тесты для компилятора языка Kotlin

Цель и постановка задачи

Цель:
Расширить существующую систему обнаружения кодовых аномалий в программах на языке Kotlin с целью обнаружения новых классов примеров кода, выделяющихся своим нестандартным содержанием

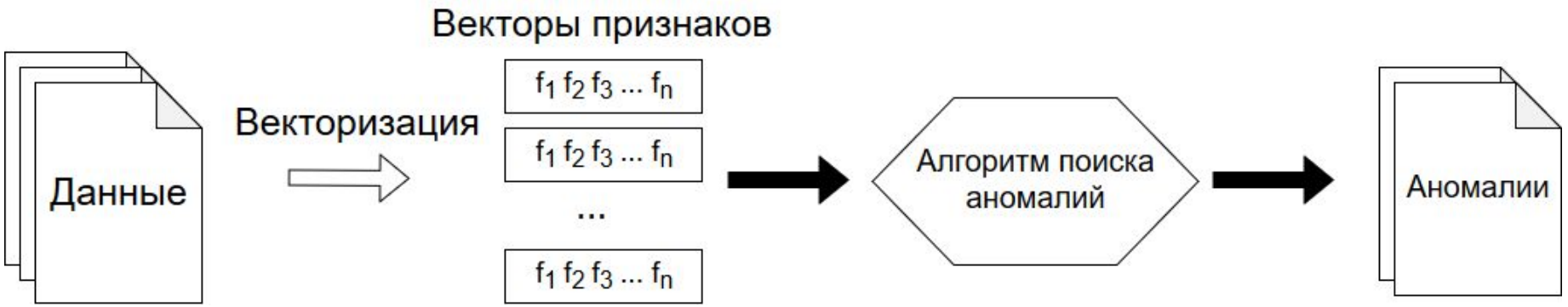
Задачи:

- Разработать и реализовать систему обнаружения кодовых аномалий на основе токенов
- Провести апробацию разработанной системы и получить набор аномалий
- Получить экспертные оценки полезности найденных аномалий

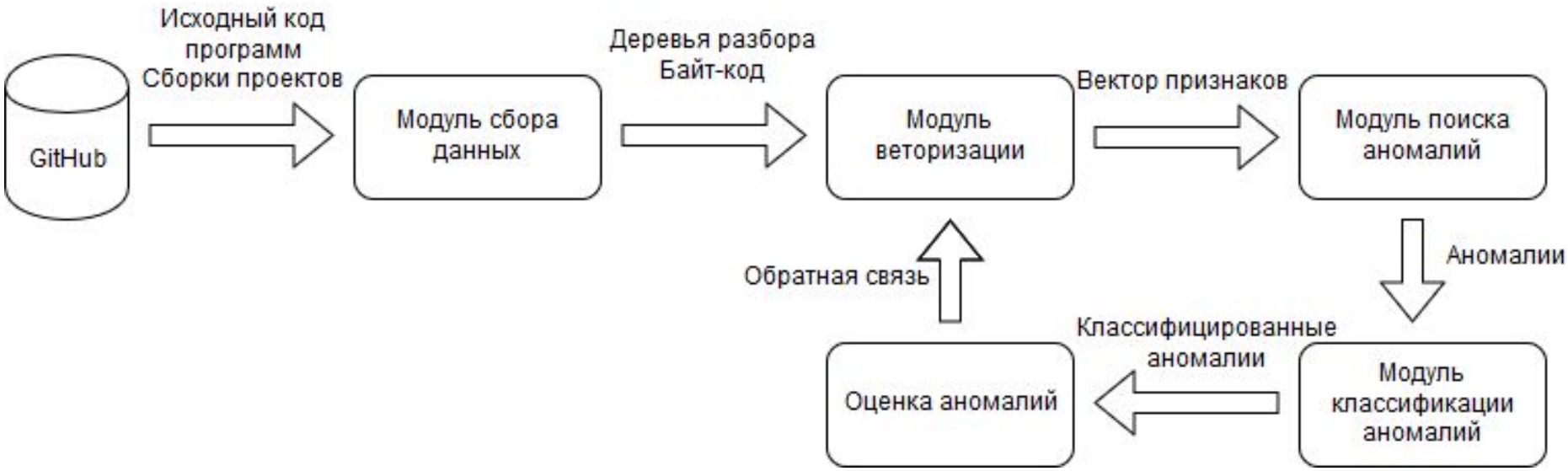
Обнаружение аномалий

Задача обнаружения аномалий включает подзадачи:

- Векторизация данных
- Поиск аномалий в пространстве векторов



Система обнаружения кодовых аномалий на языке программирования Kotlin



Смиренко К.П., Петухов В.А.

2018 г.

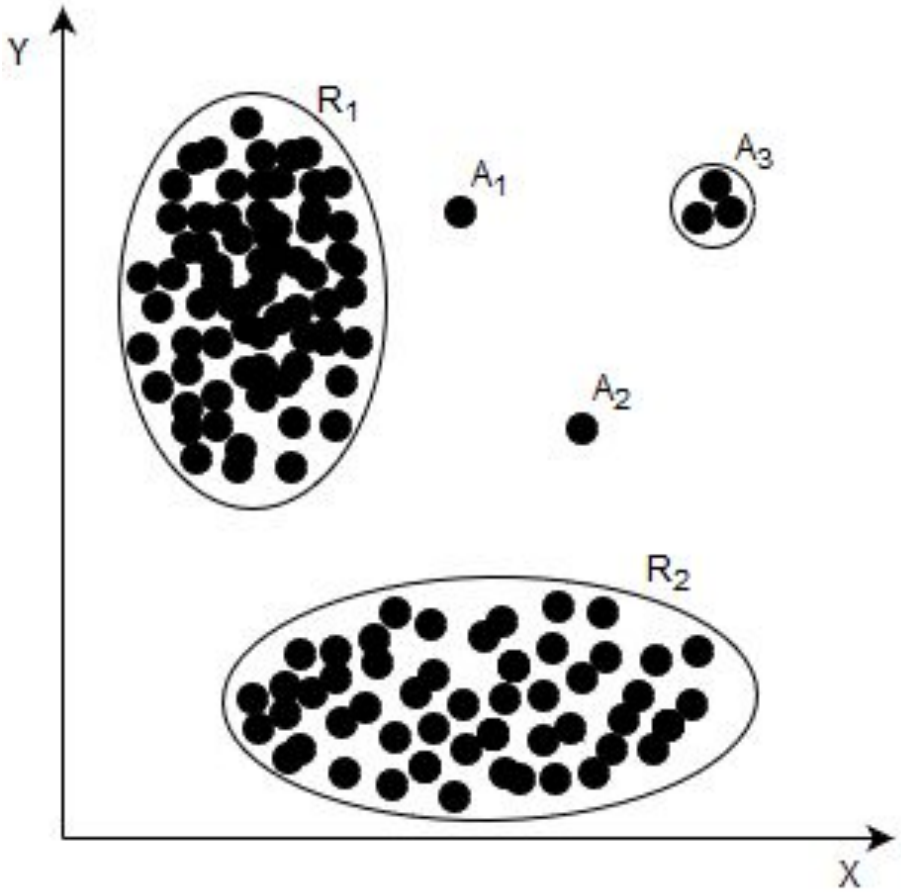
Векторизация данных

- Единица векторизуемых данных: токен
- Токен: (тип токена, лексема)
  - (IDENTIFIER, “value”)
  - (REGULAR_STRING_PART, “amount_currency”)
- Методы обработки естественного языка (NLP):
  - Bag-of-words
  - TF-IDF

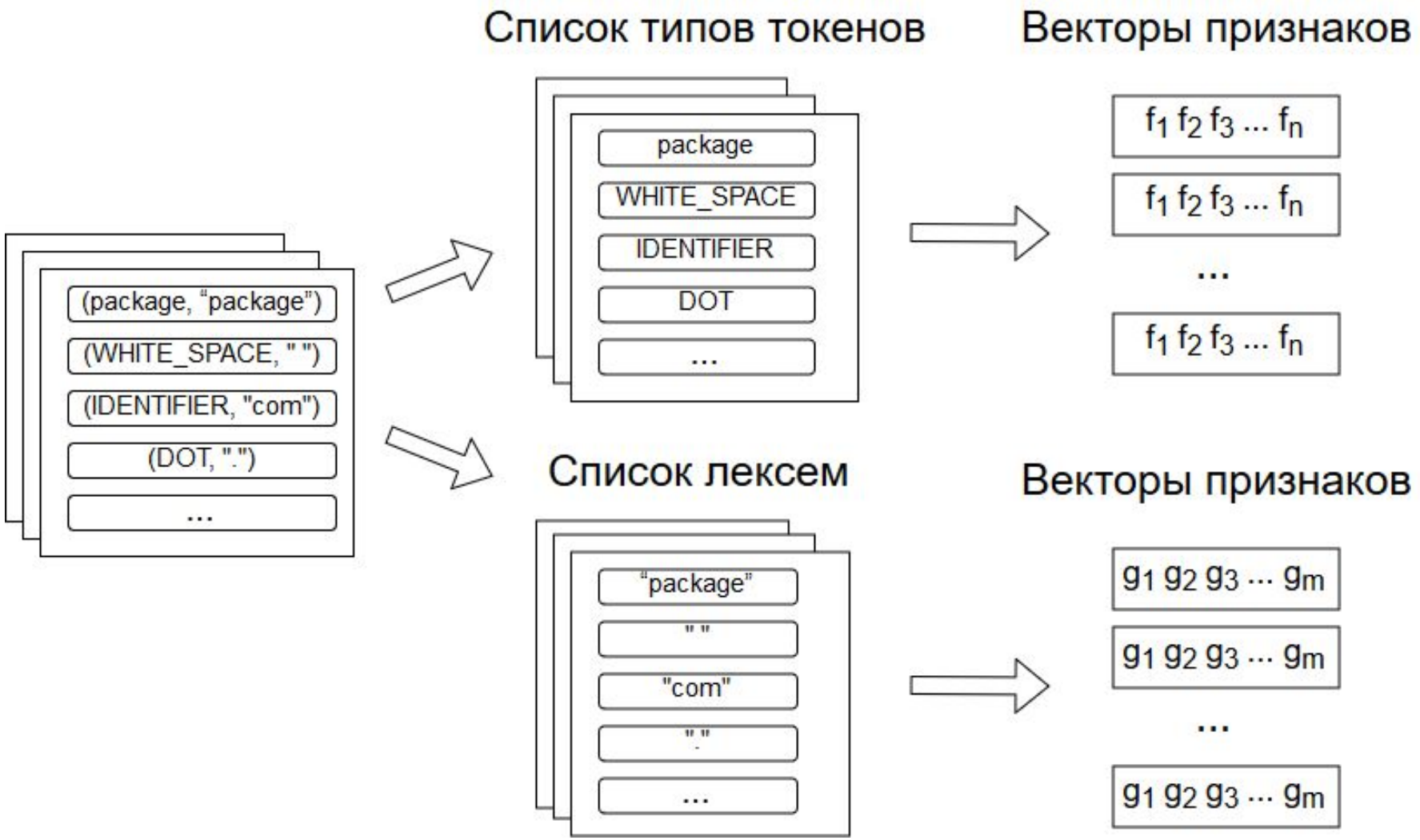
Поиск аномалий

Методы поиска аномалий в неразмеченных данных:

- One-Class SVM
- Local Outlier Factor
- DBSCAN
- HDBSCAN



Векторизация токенов



Векторизация типов токенов

- Bag-of-words
  - Подсчет вхождений
  - Однократное кодирование (one-hot encoding)
- TF-IDF
  - TF — отношение числа вхождений токена к общему числу токенов в файле

$$tf(t,d) = \frac{n_t}{\sum_k n_k}$$

  - IDF — инверсия частоты, с которой токен встречается в файле

$$idf(t,D) = \log \frac{|D|}{\left|\{d_i \in D | t \in d_i\}\right|}$$

$$tf - idf(t,d,D) = tf(t,d) \times idf(t,D)$$

Векторизация лексем

- Для каждого типа токена происходит подсчет усредненных статистик длины и хэш-кода соответствующих лексем

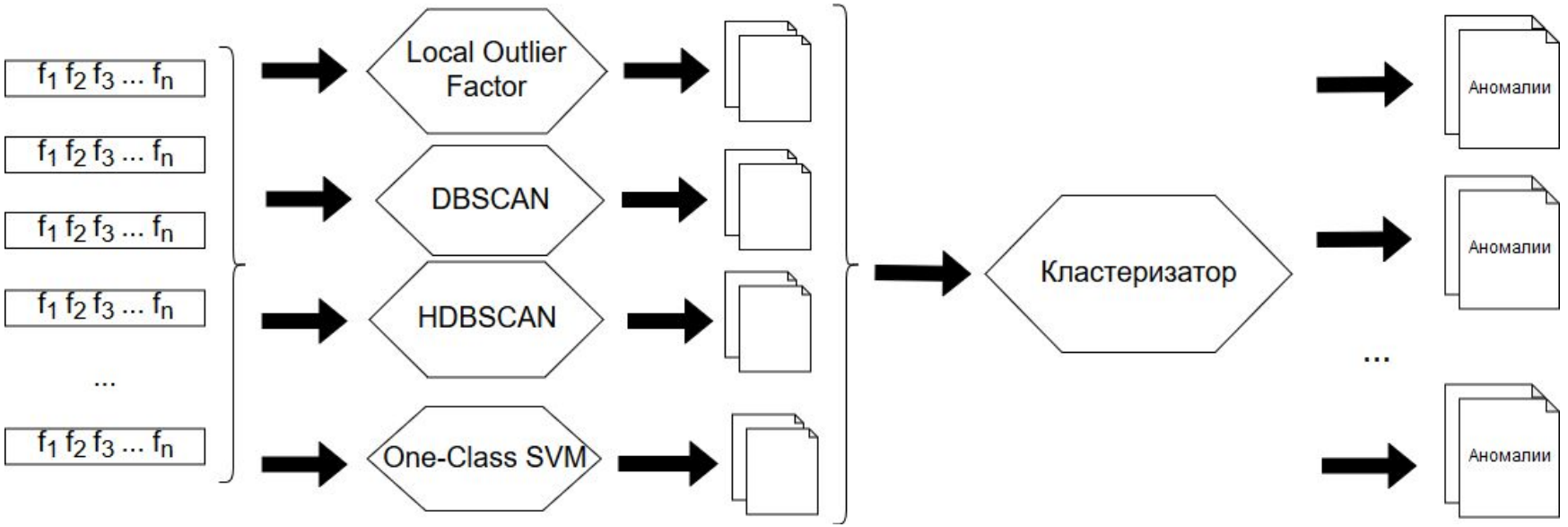
$$SumLen(t,d) = \sum_k len(l_k)$$

$$AvgLen(t,d) = \frac{\sum_k len(l_k)}{\sum_k n_{l_k}}$$

$$LogLen(t,d) = \log \frac{\sum_k len(l_k)}{\sum_k n_{l_k}}$$

$$LogHash(t,d) = \log \frac{\sum_k hash(l_k)}{\sum_k n_{l_k}}$$

Поиск аномалий



Апробация

- Собрано 1,8 млн файлов с исходным кодом на языке Kotlin с ресурса GitHub
- Получено 1237 уникальных аномалий
- Отобрано 180 наиболее интересных примеров аномалий
- Отобранные аномалии разделены на 23 класса

Экспертные оценки

- Получены экспертные оценки полезности найденных аномалий от разработчиков языка Kotlin

№	Описание класса	T	L	Q
1	Много выражений expect	+	+	5
2	Много типов аргументов	+		5
3	Много перечислений (enum)	+		4
4	Много классов	+		4
5	Много длинных циклов	+		4
6	Функции со свойствами	+		4
7	Строковое интерполирование	+	+	3
8	Арифметические выражения	+		3
9	Много выражений when	+		3
10	Большие массивы данных	+		3
11	Динамические выражения		+	3
12	Длинные строки		+	3
13	Строковые подстановки		+	3

№	Описание класса	T	L	Q
14	Много присвоений	+		2
15	Много условных выражений	+		2
16	Замена имен типов		+	2
17	Шестнадцатеричные числа		+	2
18	Выражения assert	+	+	1
19	Интервалы	+	+	1
20	Именованные аргументы	+		1
21	Многострочные строки		+	1
22	Проверки на null		+	1
23	Нестандартные кодировки		+	1

Пример аномалии (1)

```
@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("ArraysKt")
package kotlin.collections
import kotlin.comparisons.*

@kotlin.internal.InlineOnly
public expect inline operator fun <T> Array<out T>.component1(): T

@kotlin.internal.InlineOnly
public expect inline operator fun ByteArray.component1(): Byte

@kotlin.internal.InlineOnly
public expect inline operator fun ShortArray.component1(): Short


/* and 578 similar expressions */
```

Пример аномалии (2)

```
@file:Suppress("UNUSED_PARAMETER")
class Unit internal constructor()
private val u = Unit()
operator fun <P1, P2, P3, /* ... */,  P20, R> ((
  P1, P2, P3, /* ... */, P20) -> R).invoke(
  p1: P1, `2`: Unit = u, `3`: Unit = u, /* ... */, `20`: Unit = u):
        (P2, P3, /* */, P20) -> R =
  { p2: P2, p3: P3, /* ... */, p20: P20 -> this(p1, p2, p3, /* */, p20) }
operator fun <P1, P2, P3, /* ... */, P20, R> ((
  P1, P2, P3, /* ... */, P20) -> R).invoke(
  `1`: Unit = u, p2: P2, `3`: Unit = u, /* ... */, `20`: Unit = u):
        (P1, P3, /* ... */, P20) -> R =
  { p1: P1, p3: P3, /* ... */, p20: P20 -> this(p1, p2, p3, /* ... */, p20) }

/* and 18 similar expressions */
```

Сравнение с результатами других исследований

- Почти все найденные классы обнаружены впервые
- Экспертные оценки по найденным ранее классам удалось улучшить

№	Описание класса	Q_1	Q_2	Q_3	Q_4
1	Много выражений expect	5	–	–	–
2	Много типов аргументов	5	–	–	–
3	Много перечислений (enum)	4	–	4	–
4	Много классов	4	–	–	–
5	Много длинных циклов	4	2	2	2
6	Функции со свойствами	4	–	–	–
7	Строковое интерполирование	3	–	–	–
8	Арифметические выражения	3	1	–	–
9	Много выражений when	3	–	–	–
10	Большие массивы данных	3	–	–	–
11	Динамические выражения	3	–	–	–
12	Длинные строки	3	–	–	–
13	Строковые подстановки	3	–	–	–

№	Описание класса	Q_1	Q_2	Q_3	Q_4
14	Много присвоений	2	–	–	–
15	Замена имен типов	2	–	–	–
16	Много условных выражений	2	–	–	–
17	Шестнадцатеричные числа	2	–	–	–
18	Выражения assert	1	–	–	–
19	Интервалы	1	–	–	–
20	Именованные аргументы	1	–	–	–
21	Многострочные строки	1	–	–	–
22	Проверки на null	1	–	–	–
23	Нестандартные кодировки	1	–	–	–

Результаты

- Разработана и реализована система обнаружения кодовых аномалий на основе токенов, включающая модуль векторизации токенов, модуль поиска аномалий и модуль автоматизированной группировки аномалий
- Проведена апробация разработанной системы на наборе данных, содержащих 1,8 млн файлов с ресурса GitHub с исходным кодом на языке Kotlin, в результате которой получен набор кодовых аномалий
- Получены экспертные оценки полезности наиболее интересных примеров найденных аномалий от разработчиков Kotlin
- Подана статья на конференцию ESEC/FSE 2019
- Сделан доклад на конференции СПИСОК-2019