

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий



ВЫПУСКНАЯ РАБОТА БАКАЛАВРА

Тема: **Автоматическое добавление ошибок в
программный код**

Студент гр. 43501/3 А.О. Алексюк

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Работа допущена к защите
зав. кафедрой

_____ В.М. Ицыксон

«_____» _____ 2017 г.

ВЫПУСКНАЯ РАБОТА БАКАЛАВРА

**Тема: Автоматическое добавление ошибок в
программный код**

Направление: 230100 – Информатика и вычислительная техника

Выполнил студент гр. 43501/3

_____ А.О. Алексюк

Научный руководитель,
к. т. н., доц.

_____ В.М. Ицыксон

РЕФЕРАТ

Отчет

ТРАНСФОРМАЦИЯ ПРОГРАММ, СТАТИЧЕСКИЙ АНАЛИЗ,
КАЧЕСТВО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Краткое описание работы

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

1. Анализ подходов

1.1. Подход 1

1.1.1. Инструмент 1

Ссылка на статью [?]

Ссылка на сайт [?]

1.2. Подход 2

Ссылка на книгу [?]

Ссылка на рисунок ??

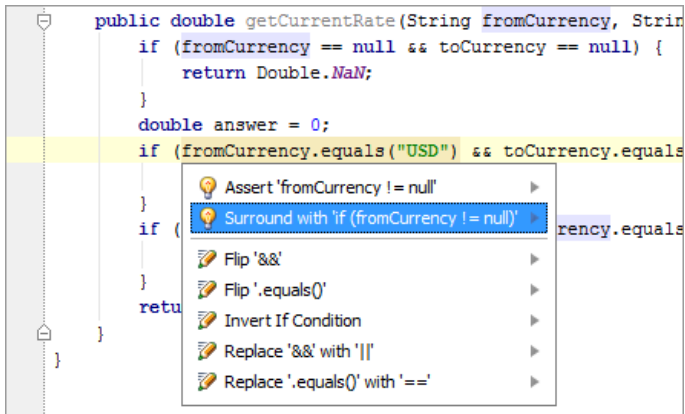


Рисунок 1.1. Предупреждение от статического анализатора в IntelliJ IDEA и всплывающее меню Quick Fix

2. Постановка задачи

3. Проектирование архитектуры системы

4. Разработка системы

Пример кода:

```
1 private int line = -1;
2 private int column = -1;
3
4 public int getLine() {
5     return line;
6 }
7
8 public int getColumn() {
9     return column;
10 }
```

5. Тестирование системы

Пара слов о тестировании системы

ЗАКЛЮЧЕНИЕ

Выводы по работе

ПРИЛОЖЕНИЕ А

ЛИСТИНГИ

Листинг А.1. Код на Java

```
1 private static boolean changesLine(final GenericTreeNode patternTree
2     , int reportLine) {
3     LineNumberFetcher fetcher = new LineNumberFetcher();
4     try {
5         fetcher.visit(patternTree);
6         return fetcher.lines.contains(reportLine);
7     } catch (Exception e) {
8         return false;
9     }
10 }
11 private static class LineNumberFetcher implements TreeVisitor {
12     Set<Integer> lines = new TreeSet<>();
13
14     @Override
15     public void visit(GenericTreeNode genericTreeNode) throws
16         Exception {
17         for (GenericTreeNode matchingNode : genericTreeNode.
18             getMatchNodes()) {
19             lines.add(matchingNode.getLine());
20         }
21         for (GenericTreeNode child : genericTreeNode.getChildren()) {
22             child.acceptVisitor(this);
23         }
24     }
25 }
```