

# **Лабораторная работа №3**

## **Применение статических анализаторов программного кода**

### **1 Цель работы**

1.1 Изучить процесс использования статических анализаторов кода для повышения качества программного обеспечения.

### **2 Литература**

2.1 Зверева В. П., Сопровождение и обслуживание программного обеспечения компьютерных систем : учебник для студ. учреждений сред. проф. Образования / В. П. Зверева, А. В. Назаров. – М. : Издательский центр «Академия», 2021. – 256 с.

### **3 Подготовка к работе**

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

### **4 Основное оборудование**

4.1 Персональный компьютер.

### **5 Задание**

5.1 Установка PVS-studio:

5.1.1 Открыть Visual Studio;

5.1.2 Перейти в меню "Extensions" (Расширения) и выбрать "Manage Extensions" (Управление расширениями);

5.1.3 В окне расширений выбрать "Online" (Онлайн) и ввести "PVS-studio" в строке поиска;

5.1.4 Найти в списке PVS-studio и нажать "Download" (Загрузить); Запустится скачивание установщика PVS-studio.

5.1.5 Установить скачанный пакет со стандартными настройками;

5.1.6 После установки запустить Visual Studio для завершения установки;

5.2 Создать новый проект C# или открыть существующий;

5.2.1 При необходимости внести несколько намеренных ошибок, таких как закомментированный код, неиспользуемые переменные, потенциальные NullReferenceException, SQL-injection и т.д.;

5.3 Анализ кода

5.3.1 Открыть вкладку «Расширения» - «PVS-studio» - «Options» и в разделе Detectable Errors (с#) во всех пунктах установить Show All.

5.3.2 В начало проверяемых файлов проекта добавить комментарий

```
// This is a personal academic project. Dear PVS-Studio,  
please check it.  
// PVS-Studio Static Code Analyzer for C, C++, C#, and  
Java: https://pvs-studio.com
```

5.3.3 Выбрать «Расширения» - «PVS-studio» - «Check current file» и дождаться завершения анализа.

5.3.4 Изучить результаты анализа программного кода.

## **6 Порядок выполнения работы**

- 6.1 Повторить теоретический материал п. 3.1;
- 6.2 Исследовать возможности SonarLint п. 5.1-5.5;
- 6.3 Ответить на контрольные вопросы п. 8;
- 6.4 Заполнить отчет п. 7.

## **7 Содержание отчета**

- 7.1 Титульный лист;
- 7.2 Цель работы;
- 7.3 Ответы на контрольные вопросы п. 6.3;
- 7.4 Вывод по проделанной работе.

## **8 Контрольные вопросы**

- 8.1 Что такое статический анализ кода и в чем его основные преимущества?
- 8.2 Какие основные отличия между статическим и динамическим анализом кода?
- 8.3 Какие типы проблем может обнаружить SonarLint в коде C#?
- 8.4 Какие типы ошибок могут быть пропущены статическими анализаторами?

## **9 Приложение**

SonarLint — это расширение для популярных IDE, которое предоставляет функции статического анализа кода непосредственно в среде разработки. SonarLint поддерживает множество языков программирования, включая C#, и позволяет обнаруживать различные типы проблем в коде.