# Отчет по лабораторной работе №3 Применение статических анализаторов программного кода

1. Цель работы
   1. Изучить процесс использования статических анализаторов кода для повышения качества программного обеспечения.
2. Задание
   1. Установка PVS-studio:
      1. Открыть Visual Studio;

Изображение выглядит как текст, снимок экрана, Мультимедийное программное обеспечение, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

* + 1. Перейти в меню "Extensions" (Расширения) и выбрать "Manage Extensions" (Управление расширениями);

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

* + 1. В окне расширений выбрать "Online" (Онлайн) и ввести "PVS-studio" в строке поиска;

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

* + 1. Найти в списке PVS-studio и нажать "Download" (Загрузить); Запустится скачивание установщика PVS-studio.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

* + 1. Установить скачанный пакет со стандартными настройками;

Изображение выглядит как текст, электроника, снимок экрана, Операционная система

Содержимое, созданное искусственным интеллектом, может быть неверным.

* + 1. После установки запустить Visual Studio для завершения установки;
  1. Создать новый проект C# или открыть существующий;
     1. При необходимости внести несколько намеренных ошибок, таких как закомментированный код, неиспользуемые переменные, потенциальные NullReferenceException, SQL-injection и т.д.;

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

* 1. Анализ кода
     1. Открыть вкладку «Расширения» - «PVS-studio» - «Options» и в разделе Detectable Errors (c#) во всех пунктах установить Show All.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

* + 1. 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

Изображение выглядит как текст, дисплей, снимок экрана, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

* + 1. Выбрать «Расширения» - «PVS-studio» - «Check current file» и дождаться завершения анализа.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

* + 1. Изучить результаты анализа программного кода.

1. Ответы на контрольные вопросы
   1. Что такое статический анализ кода и в чем его основные преимущества?

Ответ: Статический анализ кода – это анализ программы без её выполнения. Проверяет исходный код/байт-код на:

1. Синтаксические ошибки
2. Нарушения стандартов кодирования
3. Потенциальные уязвимости
4. "Запахи кода"

Преимущества: раннее обнаружение ошибок, автоматизация проверки, повышение качества кода.

* 1. Какие основные отличия между статическим и динамическим анализом кода?

Ответ:

1. Статический: до выполнения, проверяет код "на бумаге", находит потенциальные проблемы
2. Динамический: во время выполнения, обнаруживает реальные ошибки (утечки памяти, исключения)
   1. Какие типы проблем может обнаружить SonarLint в коде C#?

Ответ:SonarLint для C# обнаруживает:

1. Уязвимости безопасности (SQL-инъекции, XSS)
2. Баги (NullReferenceException, утечки памяти)
3. "Запахи кода" - сложные методы, дублирование
4. Нарушения стандартов кодирования
   1. Какие типы ошибок могут быть пропущены статическими анализаторами?

Ответ:

1. Ошибки в сторонних библиотеках
2. Проблемы, зависящие от данных времени выполнения
3. Ошибки в конфигурационных файлах
4. Проблемы производительности под нагрузкой
5. Логические ошибки в бизнес-логике
6. Вывод
   1. Изучил процесс использования статических анализаторов кода для повышения качества программного обеспечения.