



МИНОБРНАУКИ РОССИИ

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

**Институт информационных технологий (ИИТ)
Кафедра математического обеспечения и стандартизации
информационных технологий (МОСИТ)**

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №4
по дисциплине «Тестирование и верификация программного обеспечения»

Тема: Тестирование программного продукта методом «черного ящика»

Название команды: ИКБО-66-23-ForestBeavers

Состав команды: Маркелов М.Д.
Волков А.В.
Ян Х.
Цибизов М.А.

Дата выполнения: 25 ноября 2025 г.

Москва 2025

1. Введение

1.1. Цель работы

Ознакомиться с основными принципами и методами использования статических и динамических анализаторов кода для раннего выявления ошибок и потенциальных уязвимостей, что позволит повысить качество, безопасность и надёжность программного обеспечения.

1.2. Описание проектов

В рамках данной практической работы были созданы два простых проекта:

- **Проект на C++:** Простое консольное приложение, которое выводит приветствие.
- **Проект на Python:** Аналогичное консольное приложение на Python.

2. Статический анализ

2.1. Инструменты

- **Cppcheck:** для проекта на C++.
- **Pylint:** для проекта на Python.

2.2. Анализ исходного кода (до внесения ошибок)

C++ (Cppcheck)

Анализ исходного кода не выявил никаких ошибок.

Python (Pylint)

Pylint выявил следующие проблемы, связанные со стилем кода:

- C0114: Missing module docstring
- C0116: Missing function or method docstring

2.3. Анализ кода (после внесения ошибок)

Было внесено по 5 ошибок в каждый проект.

C++ (Cppcheck)

Cppcheck обнаружил 4 из 5 ошибок:

Ошибка	Обнаружена	Комментарий
Разыменование null	Да	[nullPointer], [ctunullpointer]
Выход за пределы массива	Да	[arrayIndexOutOfBounds]
Неинициализированная переменная	Да	[uninitvar]
Деление на ноль	Да	[zerodiv]
Синтаксическая ошибка	Нет	Ошибка парсинга, анализ не был завершен.

Python (Pylint)

Pylint обнаружил 3 из 5 ошибок:

Ошибка	Обнаружена	Комментарий
NameError	Да	E0602: Undefined variable 'undefined_function'
TypeError	Частично	W0104: Statement seems to have no effect
IndexError	Частично	W0104: Statement seems to have no effect

Ошибка	Обнаружена	Комментарий
AttributeError	Да	E1101: Instance of 'str' has no 'non_existent_attribute' member
SyntaxError	Да	E0001: Parsing failed: 'expected ':'

3. Динамический анализ

3.1. Инструменты

- **AddressSanitizer (g++ -fsanitize=address):** для проекта на C++.
- **Стандартный интерпретатор Python:** для проекта на Python.

3.2. Анализ кода (с ошибками)

C++ (*AddressSanitizer*)

AddressSanitizer обнаружил 2 из 5 ошибок:

Ошибка	Обнаружена	Комментарий
Разыменование null	Да	SEGV on unknown address 0x000000000000
Выход за пределы массива	Да	stack-buffer-overflow
Неинициализированная переменная	Нет	Не было обнаружено.
Деление на ноль	Частично	Обнаружено после явного добавления exit(1).

Ошибка	Обнаружена	Комментарий
Синтаксическая ошибка	Нет	Код не компилируется.

Python (Интерпретатор)

Интерпретатор Python обнаружил все 4 ошибки времени выполнения:

Ошибка	Обнаружена	Комментарий
NameError	Да	NameError: name 'undefined_function' is not defined
TypeError	Да	TypeError: can only concatenate str (not "int") to str
IndexError	Да	IndexError: list index out of range
AttributeError	Да	AttributeError: 'str' object has no attribute 'non_existent_attribute'
SyntaxError	Да	SyntaxError: invalid syntax

4. Сравнение и выводы

4.1. Статический и Динамический анализ

- **Статический анализ** эффективен для обнаружения ошибок до выполнения программы. Он может выявлять проблемы, связанные со стилем кода, потенциальные ошибки и некоторые виды уязвимостей. Однако, статические анализаторы могут пропускать ошибки, которые проявляются только во время выполнения, а также могут давать ложные срабатывания. Синтаксические ошибки могут полностью блокировать работу статического анализатора.

- **Динамический анализ** эффективен для обнаружения ошибок, которые проявляются во время выполнения программы. Он может выявлять утечки памяти, ошибки доступа к памяти и другие проблемы, которые сложно обнаружить статически. Однако, динамический анализ требует запуска программы, что не всегда возможно, и может не покрыть все ветви кода.

4.2. Рекомендации

- **Используйте оба подхода:** Статический и динамический анализ дополняют друг друга. Рекомендуется использовать оба подхода для обеспечения максимального качества кода.
- **Интегрируйте в CI/CD:** Анализаторы кода следует интегрировать в конвейеры непрерывной интеграции и доставки (CI/CD) для автоматического обнаружения ошибок на ранних стадиях разработки.
- **Выбирайте инструменты в зависимости от проекта:** Выбор инструментов зависит от языка программирования, типа проекта и специфических требований.

5. Приложения

5.1. Исходный код C++ (с ошибками)

```
#include <iostream>
```

```
void greet(const char* name) {  
    std::cout << "Привет, " << name << "!" << std::endl;  
}
```

```
int divide(int a, int b) {  
    if (b == 0) {
```

```

        std::cerr << "Ошибка: Деление на ноль!" << std::endl;
        exit(1);
    }
    return a / b;
}

```

```

int main() {
    // Ошибка 1: Null pointer dereference
    // greet(nullptr);

    // Ошибка 2: Out-of-bounds access
    int arr[5] = {1, 2, 3, 4, 5};
    // std::cout << arr[5] << std::endl;

    // Ошибка 3: Uninitialized variable
    int uninitialized;
    std::cout << uninitialized << std::endl;

    // Ошибка 4: Division by zero
    divide(1, 0);

    greet("Мир");
    return 0;
}

```

5.2. Исходный код Python (с ошибками)

```
def greet(name):  
    print(f"Привет, {name}!")  
  
if __name__ == "__main__":  
    # Ошибка 1: NameError  
    # undefined_function()  
  
    # Ошибка 2: TypeError  
    # "hello" + 5  
  
    # Ошибка 3: IndexError  
    my_list = [1, 2, 3]  
    # my_list[3]  
  
    # Ошибка 4: AttributeError  
    my_string = "Привет"  
    my_string.non_existent_attribute  
  
    # Ошибка 5: SyntaxError  
    for i in range(5):  
        print(i)  
  
    greet("Мир")
```