**Отчет**

Студента 315 группы

По ПиТПМодуле й

Техникума Информатики и Права

Иванченко Даниил Богданович

Проверила: Подсекина Татьяна Сергеевна

# Отчет по тестированию программы

## 1. Критерии качества тестирования

• Полнота тестирования — насколько полно проверены все требования и функции.

• Корректность тестов — тесты должны выявлять реальные ошибки, а не ложные срабатывания.

• Повторяемость — тесты должны давать одинаковые результаты при одинаковых условиях.

• Прослеживаемость — каждая проверка должна быть связана с конкретным требованием.

• Автоматизируемость — тесты пригодны для автоматического запуска.

• Репрезентативность — тестовые данные отражают реальные сценарии работы пользователя.

• Эффективность — минимальные затраты времени при максимальном охвате проверок.

• Надёжность результатов — устойчивость тестов при изменениях программы.

## 2. Методы инспекции кода

• Ревью кода (Code Review) — групповая проверка исходного кода коллегами.

• Пошаговая проверка (Walkthrough) — разработчик объясняет логику программы.

• Формальная инспекция (Formal Inspection) — строгая процедура с модератором и рецензентами.

• Парное программирование (Pair Programming) — один пишет код, другой проверяет.

• Статический анализ — автоматическая проверка кода инструментами (SonarQube, PVS-Studio).

## 3. Классификация ошибок

### По видам деятельности

|  |  |  |
| --- | --- | --- |
| Вид деятельности | Пример ошибки | Описание |
| Ошибки анализа | Неправильная трактовка требований | Программа решает не ту задачу |
| Ошибки проектирования | Неверная структура данных | Сбои или низкая производительность |
| Ошибки документации | Несоответствие описания коду | Пользователь вводится в заблуждение |
| Ошибки реализации | Ошибки в коде или формулах | Неправильная работа программы |

### По серьёзности

|  |  |  |
| --- | --- | --- |
| Степень | Описание | Примеры |
| Высокая | Ошибка приводит к сбою системы | Деление на ноль, падение программы |
| Средняя | Ошибка влияет на отдельные функции | Неверный расчёт в одной формуле |
| Низкая | Ошибка не мешает работе, но ухудшает удобство | Опечатки, некорректный интерфейс |

## 4. Оценка ошибок программы

По видам деятельности:

|  |  |  |
| --- | --- | --- |
| Категория | Количество | Примеры |
| Ошибки анализа | 1 | Неправильное понимание условия задачи |
| Ошибки проектирования | 2 | Неверное использование переменной, неудачная структура |
| Ошибки документации | 1 | Несовпадение описания функции с кодом |
| Ошибки реализации | 3 | Деление на ноль, неверная формула, неправильное условие |

По серьёзности:

|  |  |  |
| --- | --- | --- |
| Серьёзность | Количество | Примеры |
| Высокая | 1 | Деление на ноль |
| Средняя | 3 | Неверные вычисления |
| Низкая | 3 | Опечатки, некорректный вывод сообщений |

## 5. Анализ количества тестов

Для проверки программы, реализующей кусочную функцию, требуется провести не менее 6 тестов:

|  |  |  |
| --- | --- | --- |
| № | Условия | Ожидаемый результат |
| 1 | x < 1, c ≠ 0 | Вычисляется a\*x² + b/c |
| 2 | x = 1 | Переход между ветвями |
| 3 | x > 1.5, c = 0 | Вычисляется (x-a)/(x-c)² |
| 4 | 1 < x ≤ 1.5 | Вычисляется x²/c² |
| 5 | c = 0, x < 1 | Проверка деления на ноль |
| 6 | Некорректный ввод | Вывод сообщения об ошибке |

Итого: **6 тестов** обеспечивают полное покрытие всех ветвей условий.

Ответы на вопросы:

**1. Что является объектами тестирования?**

Объектами тестирования являются элементы программного обеспечения, подлежащие проверке: модули, интерфейсы, базы данных, документация, интеграционные связи между системами.

**2. Тестовый сценарий — это...**

Тестовый сценарий (Test Case) — это последовательность шагов с входными данными, условиями и ожидаемыми результатами для проверки конкретной функции программы.

**3. В чем заключается метод эквивалентных разбиений?**

Метод эквивалентных разбиений заключается в разделении множества входных данных на классы эквивалентности, внутри которых программа должна работать одинаково. Из каждой группы выбирается одно типичное значение для проверки.

**4. Поясните, что представляет собой мера Холстера?**

Мера Холстера (Halstead metrics) — это набор показателей, характеризующих сложность кода. Основана на подсчете операторов и операндов. Позволяет оценить объем, трудоемкость и сложность сопровождения программы.

**5. В чем заключается метод эквивалентных разбиений?**

Метод заключается в разделении входных данных на классы эквивалентности и выборе представительных значений для каждого класса, чтобы уменьшить количество тестов при сохранении полноты проверки.