

Практическая работа: Тестирование «белым ящиком»

Теоретическая часть

Что такое тестирование «белым ящиком»?

Тестирование «белым ящиком» (white-box testing) — метод тестирования программного обеспечения, при котором тестировщик имеет доступ к внутренней структуре и реализации кода. Тестировщик видит код "изнутри" (как через прозрачный белый ящик) и создает тесты на основе анализа логики программы.

Основные принципы:

- **Знание кода:** Тестировщик анализирует исходный код
- **Покрытие путей:** Тесты должны покрывать все возможные пути выполнения
- **Ветвления и условия:** Проверяются все условия `if/else`, циклы
- **Граничные значения:** Тестируются значения на границах условий

Методы тестирования «белым ящиком»:

1. **Покрытие операторов** – каждая строка кода выполняется
2. **Покрытие ветвей** – все ветви условий проверяются
3. **Покрытие условий** – все логические условия тестируются

Практическое задание

Задача: Найти и исправить ошибки в коде с помощью тестирования «белым ящиком»

Код в файле `buggy_code.py`

Задание для студентов

Часть 1: Напишите тесты (`test_buggy_code.py`)

Часть 2: Анализ кода и поиск ошибок



Что конкретно нужно сделать:

✓ Задание 1: Анализ кода

1. Изучите код в `buggy_code.py`
2. Запустите тесты из `test_buggy_code.py`
3. Запишите, какие тесты не проходят

Функция `calculate_discount`:

- Ошибка в условиях возраста: _____
- Проблема с комбинациями скидок: _____

Функция `validate_password`:

- Ошибка в проверке условий: _____
- Проблема с граничными значениями: _____

Функция `get_temperature_status`:

- Ошибка в границах диапазонов: _____

Задание 3: Исправление кода

Исправьте ошибки в `buggy_code.py` и убедитесь, что все тесты проходят

Задание 4: Анализ покрытия

Напишите, какие пути выполнения кода были протестированы и какие были добавлены после исправлений

Критерии оценки:

- **3 балла** - Найдены и описаны ошибки
- **4 балла** - Исправлены некоторые ошибки, тесты частично проходят
- **5 баллов** - Все ошибки исправлены, все тесты проходят, написан анализ покрытия



Подсказки для студентов:

1. Внимательно изучите условия `if/elif/else`
2. Проверяйте граничные значения (равно, больше, меньше)
3. Анализируйте все возможные комбинации условий
4. Используйте отладку для пошагового выполнения кода