Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж им. Б.С. Галущака»

Самостоятельная работа №5

Тема: Определение количества ошибок в программном продукте и числа необходимых тестов.

Учебная дисциплина: МДК.01.02 Поддержка и тестирование программных модулей

Выполнил: студент группы ПР-21.101

Портнов М.А.

Проверила: Гербер М.Р.

2024

Цель: Изучить процесс и организацию тестирования; критерии качества тестирования; методы, используемые при тестировании; классификацию и оценку ошибок  
  
Вопросы:  
**1. Что является объектами тестирования?**

Объектами тестирования могут быть различные компоненты программного обеспечения, включая:

Модули или компоненты: Отдельные части приложения, такие как функции, методы, классы и библиотеки.

Интерфейсы: Пользовательские интерфейсы (UI), API (Application Programming Interface).

Системы и подсистемы: Взаимодействие между различными частями системы.

Процессы: Бизнес-процессы и логика приложения.

Данные: Базы данных, файлы конфигурации, потоки данных и их обработка.

Производительность: Время отклика системы, использование ресурсов.

**2. Тестовый сценарий - это...**

Тестовый сценарий — это конкретный набор условий, включающих начальные состояния, входные данные и ожидаемые результаты, используемый для проверки правильности работы программы или ее части. Тестовый сценарий описывает шаги, которые необходимо выполнить, и результаты, которые нужно сравнить с ожидаемыми, чтобы определить, правильно ли работает тестируемый объект.

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

Метод эквивалентных разбиений (Equivalence Partitioning) заключается в разделении множества возможных входных значений функции на подмножества (или классы эквивалентности), которые предполагается, что обрабатываются одинаково. Для каждого класса выбирается один или несколько представителей, которые используются для тестирования. Предполагается, что если тестирование одного значения из класса эквивалентности успешно, то и другие значения из этого класса будут обрабатываться правильно.

Например, если функция принимает целые числа от 1 до 100, то можно создать следующие классы эквивалентности:

Значения меньше 1 (недопустимые).

Значения от 1 до 100 (допустимые).

Значения больше 100 (недопустимые).

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

Мера Холстеда (Halstead Metrics) — это набор метрик программного обеспечения, предложенных Морисом Холстедом для измерения сложности программного кода. Она основана на анализе операторов и операндов в программе. Основные метрики включают:

Длина программы (N): Общее количество операторов и операндов.

Объем программы (V): Логический размер программы, измеряемый в словах.

Словарь программы (n): Общее количество уникальных операторов и операндов.

Трудоемкость (E): Оценка усилий, необходимых для написания и понимания программы.

Потенциальная ошибка (B): Оценка количества ошибок, которые могут быть допущены.

Формулы для вычисления этих метрик:

N = N1 + N2 (где N1 — количество операторов, N2 — количество операндов).

V = N \* log2(n).

Трудоемкость (E) = V / (2 \* n2) (где n2 — количество уникальных операндов).

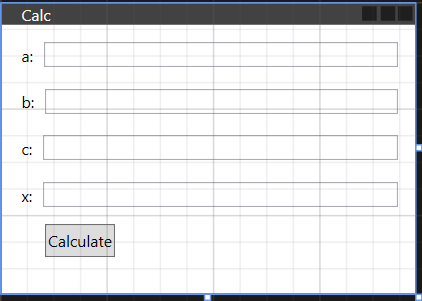
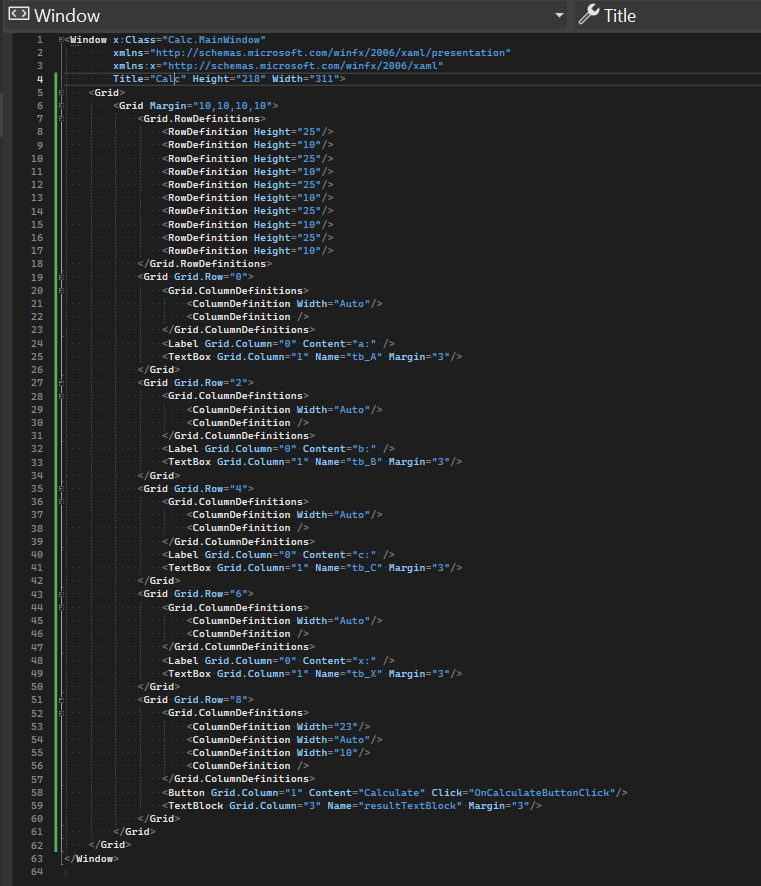
**5. В чем заключается метод эквивалентных разбиений?**

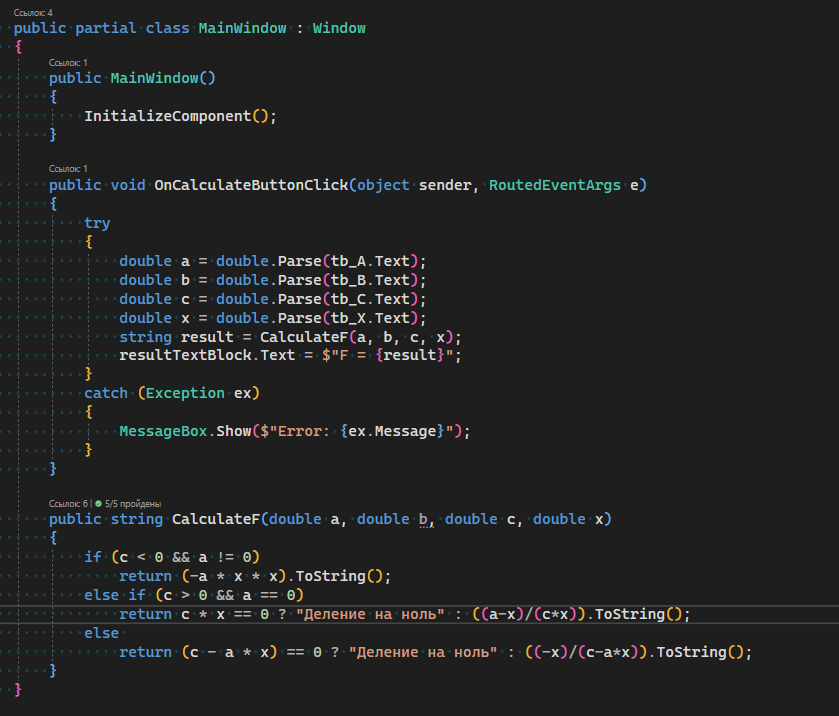
Этот вопрос дублирует третий. Повторю ответ:

Метод эквивалентных разбиений (Equivalence Partitioning) заключается в разделении множества возможных входных значений функции на подмножества (или классы эквивалентности), которые предполагается, что обрабатываются одинаково. Для каждого класса выбирается один или несколько представителей, которые используются для тестирования. Предполагается, что если тестирование одного значения из класса эквивалентности успешно, то и другие значения из этого класса будут обрабатываться правильно.

Пример применения метода эквивалентных разбиений уже был приведен выше.

**Отчет по самостоятельной работе:**

1. **Вертска приложения:**
2. **Код верстки:**
3. **Код основной программы:**



1. **Unit-тесты** 