Практическое занятие № 2.

Сравнение результатов тестирования с требованиями технического задания и/или спецификацией.

Цели: ознакомиться документацией на разработку программного обеспечения; сравнить результаты тестирования с требованиями к программным продуктам.

Теоретические сведения:

Техническое задание – это основной исходный документ для разработки программного обеспечения, в котором формулируются основные цели разработки, список принципиальных требований к продукту, определяются сроки и этапы разработки и регламентируется процесс приемно-сдаточных испытаний.

Основой технического задания являются:

1. Исходные данные и требуемые результаты

2. Среда (программная и аппаратная)

3. Возможное взаимодействие с другим программным обеспечением и/или конкретными техническими средствами

Практическая часть:

1. Составьте техническое задание, которое будет содержать необходимые требования к программному продукту

2. Составьте спецификацию на разработку программного продукта

3. Сравните результаты тестирования из предыдущей работы с требованиями технического задания

4. Сравните результаты тестирования из предыдущей работы с требованиями спецификации

Техническое задание

Название проекта: Стандартный калькулятор

Цель проекта: Разработка калькулятора, выполняющего основные арифметические операции над двумя числами.

Функциональные требования:

1. Программа должна запрашивать у пользователя два числа с плавающей точкой.

2. Программа должна запрашивать у пользователя выбор операции (сложение, вычитание, умножение, деление).

3. Программа должна обрабатывать как положительные, так и отрицательные и дробные числа.

4. Программа должна выводить сообщение об ошибке при некорректном вводе данных

Нефункциональные требования:

1. Программа должна быть написана на языке программирования Python.

2. Программа должна иметь простой и интуитивно понятный интерфейс.

3. Программа должна завершать выполнение с выводом результата не более чем за 1 секунду.

4. Программа должна быть устойчивой к ошибкам и не завершаться аварийно при вводе некорректных данных.

Ограничения:

1. Программа не должна использовать сторонние библиотеки для выполнения основных функций.

2. Программа должна работать на всех основных операционных системах (Windows, macOS, Linux).

Спецификация на разработку программного продукта:

Название проекта: Стандартный калькулятор

Описание продукта:

Программное обеспечение представляет собой программу, которая принимает два числа и арифметическую операцию в качестве входных данных, выводит результат операции в консоль.

Функциональные компоненты:

1. Модуль ввода данных:

– Принимает два числа.

– Принимает строковое значение, обозначающее арифметическую операцию.

2. Модуль проверки корректности данных;

– Проверяет корректность ввода (числовые значения).

– Проверяет корректность ввода (строковое значение).

3. Модуль вычислений:

– Вызывает соответствующую функцию.

– Обрабатывает исключения при некорректном вводе данных.

– Корректно обрабатывает пустой массив.

– Возвращает результат вычисления.

4. Модуль вывода данных:

– Отображает результат вычислений.

Технические требования:

1. Язык программирования: Python.

2. Платформы: Windows, macOS, Linux.

3. Интерфейс: Командная строка.

4. Время отклика: Не более 1 секунды для выполнения операций.

Критерии приемки:

1. Программа должна корректно производить арифметическую операцию.

2. Программа должна обрабатывать ввод некорректных данных, выводя предупреждение о некорректных данных.

3. Программа должна завершаться без ошибок и корректно выводить результат.

Ожидаемые результаты тестирования представлены в таблице 1.

Таблица 1 – Ожидаемые результаты соответствия ТЗ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тестируемая функция | Тест (значения для входных данных) | Ожидаемый результат (значения для выходных данных) | Фактический результат (полученные значения выходных данных) | Результат тестирования | Соответствие спецификации |
| Main | Запуск программы (нормальные входные данные) | Вывод на экран числа, полученного после проведения арифметической операции. | 11.0 | Успешно | да |
| Запуск программы (некорректный ввод чисел) | Ошибка: Пожалуйста, введите числа. | Ошибка: Пожалуйста, введите числа. | Успешно | да |
| Запуск программы (некорректный ввод операции) | Ошибка: Недопустимая операция или деление на ноль. | Ошибка: Недопустимая операция или деление на ноль. | Успешно | да |
| Calculate | (6.5, -5, 'add') | 1.5 | 1.5 | Успешно | да |
| (6.5, -5, 'substract') | 11.5 | 11.5 | Успешно | да |
| (6.5, -5, 'multiply') | -32.5 | -32.5 | Успешно | да |
| (6.5, -5, 'divide') | -1.3 | -1.3 | Успешно | да |
| (6.5, 0, 'divide') | Ошибка: Недопустимая операция или деление на ноль. | ZeroDivisionError: float division by zero | Неуспешно | нет |
| Add | (6, 5) | 11.0 | 11.0 | Успешно | да |
| (6.5, 5) | 11.5 | 11.5 | Успешно | да |
| (6, 0) | 6.0 | 6.0 | Успешно | да |
| (6, -5) | 1.0 | 1.0 | Успешно | да |
| Subtract | (6, 5) | 1.0 | 1.0 | Успешно | да |
| (6.5, 5) | 1.5 | 1.5 | Успешно | да |
| (6, 0) | 6.0 | 6.0 | Успешно | да |
| (6, -5) | 11.0 | 11.0 | Успешно | да |
| Multiply | (6, 5) | 30.0 | 30.0 | Успешно | да |
| (6.5, 5) | 32.5 | 32.5 | Успешно | да |
| (6, 0) | 0.0 | 0.0 | Успешно | да |
| (6, -5) | -30.0 | -30.0 | Успешно | да |
| Divide | (6, 5) | 1.2 | 1.2 | Успешно | да |
| (6.5, 5) | 1.3 | 1.3 | Успешно | да |
| (6, 0) | Ошибка: Недопустимая операция или деление на ноль. | ZeroDivisionError: float division by zero | Неуспешно | нет |
| (6, -5) | -1.2 | -1.2 | Успешно | да |

Ожидаемые результаты соответствия спецификации представлены в таблице 2.

Таблица 2 – Ожидаемые результаты соответствия спецификации

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тестируемая функция | Тест (значения для входных данных) | Ожидаемый результат (значения для выходных данных) | Фактический результат (полученные значения выходных данных) | Результат тестирования | Соответствие ТЗ |
| Main | Запуск программы (нормальные входные данные) | Вывод на экран числа, полученного после проведения арифметической операции. | 11.0 | Успешно | да |
| Запуск программы (некорректный ввод чисел) | Ошибка: Пожалуйста, введите числа. | Ошибка: Пожалуйста, введите числа. | Успешно | да |
| Запуск программы (некорректный ввод операции) | Ошибка: Недопустимая операция или деление на ноль. | Ошибка: Недопустимая операция или деление на ноль. | Успешно | да |
| Calculate | (6.5, -5, 'add') | 1.5 | 1.5 | Успешно | да |
| (6.5, -5, 'substract') | 11.5 | 11.5 | Успешно | да |
| (6.5, -5, 'multiply') | -32.5 | -32.5 | Успешно | да |
| (6.5, -5, 'divide') | -1.3 | -1.3 | Успешно | да |
| (6.5, 0, 'divide') | Ошибка: Недопустимая операция или деление на ноль. | ZeroDivisionError: float division by zero | Неуспешно | нет |
| Add | (6, 5) | 11.0 | 11.0 | Успешно | да |
| (6.5, 5) | 11.5 | 11.5 | Успешно | да |
| (6, 0) | 6.0 | 6.0 | Успешно | да |
| (6, -5) | 1.0 | 1.0 | Успешно | да |
| Subtract | (6, 5) | 1.0 | 1.0 | Успешно | да |
| (6.5, 5) | 1.5 | 1.5 | Успешно | да |
| (6, 0) | 6.0 | 6.0 | Успешно | да |
| (6, -5) | 11.0 | 11.0 | Успешно | да |
| Multiply | (6, 5) | 30.0 | 30.0 | Успешно | да |
| (6.5, 5) | 32.5 | 32.5 | Успешно | да |
| (6, 0) | 0.0 | 0.0 | Успешно | да |
| (6, -5) | -30.0 | -30.0 | Успешно | да |
| Divide | (6, 5) | 1.2 | 1.2 | Успешно | да |
| (6.5, 5) | 1.3 | 1.3 | Успешно | да |
| (6, 0) | Ошибка: Недопустимая операция или деление на ноль. | ZeroDivisionError: float division by zero | Неуспешно | нет |
| (6, -5) | -1.2 | -1.2 | Успешно | да |

В процессе сравнения результатов тестирования с требованиями технического задания было выявлено, что программа выдает ошибку при вводе нуля в качестве второго числа и деления в виде операции, при остальных значениях программа корректно отображает результат арифметической операции и обрабатывает ситуации неверного ввода данных.

В процессе сравнения результатов тестирования с требованиями спецификации было выявлено, что программа выдает ошибку при вводе нуля в качестве второго числа и деления в виде операции, при остальных значениях программа корректно отображает результат арифметической операции и обрабатывает ситуации неверного ввода данных, выводя предупреждение.

Контрольные вопросы:

1. Подтверждает ли тестирование правильность программы?

Ответ: Нет, тестирование не подтверждает абсолютную правильность программы. Оно лишь помогает выявить ошибки и убедиться, что программа работает корректно на конкретных тестовых данных.

Тестирование уменьшает вероятность наличия ошибок, но не исключает её полностью.

2. Что можно сказать о программе, если она на значительном количестве тестов ведет себя правильно?

Если программа проходит значительное количество тестов, включая граничные случаи и стресс-тесты, это говорит о её высокой надежности и качестве.

Однако это не гарантирует отсутствие ошибок в других, непроверенных сценариях. Программа может быть частично корректной, но не абсолютно безошибочной.

3. Может ли повысить надежность программы процесс тестирования?

Ответ: Да, процесс тестирования может повысить надежность программы. Выявление и исправление ошибок на этапе тестирования уменьшает вероятность сбоев в реальной эксплуатации.

4. Типы ошибок, обнаруживаемые при тестировании.

Ответ*:* Синтаксические ошибки *,*Логические ошибки, Ошибки ввода-вывода, Ошибки производительности, Ошибки взаимодействия, Ошибки безопасности, Граничные ошибки

5. Основные принципы тестирования

Ответ:

* Принцип отсутствия ошибок (Fallacy of exhaustive testing):

Невозможно протестировать программу на всех возможных входных данных. Поэтому тестирование должно быть целенаправленным и основываться на анализе рисков.

* Принцип раннего тестирования:

Тестирование должно начинаться как можно раньше, на этапе проектирования и разработки, чтобы выявлять ошибки до их внедрения в код.

* Принцип независимости:

Тестирование должно проводиться независимыми специалистами (например, тестировщиками, а не разработчиками).

* Принцип покрытия:

Тесты должны покрывать все ключевые функциональные и нефункциональные требования, а также граничные случаи.

* Принцип повторяемости:

Тесты должны быть воспроизводимыми, чтобы можно было проверить исправление ошибок и убедиться в стабильности программы.

* Принцип приоритета:

Тестирование должно быть сосредоточено на наиболее критичных и часто используемых частях программы.

* Принцип «тестирование показывает наличие дефектов»:

Тестирование помогает выявить ошибки, но не доказывает их отсутствие.

* Принцип «группировки дефектов»:

Ошибки часто группируются в определенных модулях или компонентах программы. Если в одном модуле найдено много ошибок, стоит уделить ему больше внимания.

* Принцип «тестирование зависит от контекста»:

Подход к тестированию зависит от типа приложения, его сложности и требований. Например, тестирование веб-приложения отличается от тестирования встроенных систем.

* Принцип «отсутствие ошибок — это иллюзия»:

Даже если программа работает корректно на всех тестах, это не означает, что она абсолютно безошибочна.

Вывод: в ходе работы ознакомился с документацией на разработку программного обеспечения и сравнил результаты тестирования с требованиями к программному продукту.