**Практическое занятие №41**

**Тема:** Разработка тестового сценария проекта.

**Цель:** получить навыки разработки тестовых сценариев.

Теоретические сведения:

1. Для оценки стоимости программного обеспечения учитываются затраты на разработку, тестирование, поддержку и обновление. Также учитываются возможные затраты на исправление ошибок после выпуска продукта. Ошибки в ПО могут возникнуть из-за неправильного понимания требований, недостаточного тестирования, неудачного проектирования, недостаточного контроля версий и других факторов.
2. Виды тестирования

Функциональное тестирование:

* модульное тестирование: проверка отдельных модулей программы на корректность их работы;
* интеграционное тестирование: тестирование взаимодействия между различными модулями или компонентами программы;
* системное тестирование: проверка работы всей системы в целом, в соответствии с её функциональными требованиями.

Нагрузочное тестирование:

* тестирование производительности: определение производительности системы при различных условиях нагрузки;
* тестирование стабильности: проверка устойчивости системы при длительной нагрузке.

Тестирование безопасности:

* тестирование на проникновение: попытка взлома системы для выявления её уязвимостей;
* аудит безопасности: анализ системы на предмет соответствия стандартам безопасности.

Автоматизированное тестирование:

* модульное тестирование с использованием фреймворков: Автоматизация тестов для отдельных модулей программы;
* интеграционное и системное тестирование с использованием инструментов автоматизации: автоматизация процессов тестирования на уровне системы.

Тестирование совместимости:

* тестирование на различных платформах: проверка работы программы на различных операционных системах:
* тестирование на различных браузерах: обеспечение корректной работы в различных веб-браузерах.

Тестирование на соответствие:

* тестирование по спецификациям: проверка соответствия программы заранее установленным стандартам и требованиям.

Тестирование восстановления после сбоя: проверка способности системы быстро и корректно восстанавливаться после сбоев.

Методы тестирования:

* ручное тестирование;
* автоматизированное тестирование;
* тестирование черного ящика;
* тестирование белого ящика;
* тестирование производительности;
* тестирование безопасности;
* тестирование совместимости.

1. Тест – это процесс проверки программы с целью выявления ошибок или подтверждения ее правильной работы. Тест представляет собой набор входных данных, условий выполнения и ожидаемых результатов.
2. Требования к разработке тестовых сценариев:

* понимание требований;
* ясность и полнота;
* назначение теста;
* повторяемость;
* масштабируемость;
* обработка исключений;
* отчетность;
* актуализация.

1. Правила разработки тестовых сценариев:

* согласование с требованиями;
* простота и понятность;
* независимость и изоляция;
* конкретность и четкость;
* покрытие функциональности;
* комментарии и документация;
* регулярное обновление.

Задание: написать программу, найти минимальный набор тестов для программы своего варианта. Решать можно с применением любых коллекций Python.

Вариант №2

1. Даны натуральное число N и одномерный массив A 1 , A 2 , …, A N целых чисел. Получить массив, который отличается от исходного тем, что все нечетные элементы удвоены, а четные получены сложением собственного значения с первоначальным значением последующего нечетного.

Таблица 1 – Тестовые проверки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер теста | N | Ожидаемый результат | Что проверяется | Результат проверки |
| 1 | 5 | [60, 64, 31, 78, 15] | Вывод данных | Да |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | [91, 95, 62, 93, 30] |  |  |
| 2 | -4 | Сообщение | Введенное число N не является натуральным | Да |
| 3 | 0 | Сообщение | Введенное число N не является натуральным | Да |
| 4 | а | Сообщение | Введенное значение N не является числом | Да |