1. **Цель работы:**
   * Изучение классификации видов тестирования и их применение на практике.
   * Оценка различных подходов к организации контроля качества программных систем.
   * Приобретение навыков локализации дефектов и создания полных наборов тестов.
2. **Описание программного проекта:**
   * **Название:** TeaOnlineShop
   * **Цель:** разработка онлайн-магазина для продажи чая, где пользователи могут регистрироваться, просматривать каталог товаров, добавлять товары в корзину, оформлять заказы и производить оплату.
   * **Основные функции:**
     + Регистрация и аутентификация пользователей
     + Просмотр и поиск товаров
     + Добавление товаров в корзину и оформление заказа
     + Обработка платежей
     + Управление заказами для администраторов
     + Просмотр и поиск статей
     + Добавление статей
     + Управление информацией о профилем
   * **Технологии:** Использованы Java, Spring, HTML, CSS, JavaScript, React, PostgreSQL.
3. **Виды, типы, методы и уровни тестирования:**

**Уровни тестирования:**

* + **Компонентное (модульное) тестирование:** тестирование отдельных модулей или компонентов системы в изоляции. Этот уровень тестирования фокусируется на проверке отдельных функций или методов классов, чтобы удостовериться в их правильной работе. Пример: проверка функций регистрации, входа в систему, сброса пароля.
  + **Интеграционное тестирование:** тестирование взаимодействия между несколькими модулями или компонентами системы. Этот уровень тестирования фокусируется на проверке корректного обмена данными и взаимодействия между модулями. Пример: взаимодействие между модулем корзины и модулем оформления заказа: Проверка корректного переноса данных из корзины в форму заказа.
  + **Системное тестирование:** тестирование всей системы целиком, включая как функциональные, так и нефункциональные аспекты. Этот уровень тестирования фокусируется на проверке всей системы в реальных условиях эксплуатации. Пример: проверка функциональности регистрации, поиска товаров, добавления в корзину, оформления заказа и оплаты, проверка нефункциональных требований, таких как производительность системы при высокой нагрузке, безопасность транзакций.

**Типы тестирования:**

* + **Функциональное тестирование:** тестирование системы на предмет корректного выполнения заявленных функциональных требований. Фокусируется на том, что система делает. Пример: проверка функций регистрации и аутентификации пользователей.
  + **Нефункциональное тестирование:** тестирование системы на предмет нефункциональных требований, таких как производительность, безопасность, удобство использования и надежность. Фокусируется на том, как система работает. Пример: время отклика сервера при большом количестве запросов.

**Методы тестирования:**

* + **Статическое тестирование:** тестирование программного кода и документации без выполнения кода. Включает рецензирование, анализ кода и использование статических анализаторов. Пример: рецензирование кода на предмет соблюдения стандартов кодирования.
  + **Динамическое тестирование:** выполнение тестов, включающих методы черного ящика для функционального тестирования и методы белого ящика для проверки внутренней логики приложения. Пример: проверка функциональности регистрации, оформления заказа и оплаты без знания внутренней реализации.

1. **Примеры дефектов, характерных для каждого вида тестирования:**
   * **Компонентное тестирование:** ошибки в логике аутентификации пользователя, неправильная обработка форм ввода.
   * **Интеграционное тестирование:** проблемы с передачей данных между модулями, такие как некорректная работа корзины при оформлении заказа.
   * **Системное тестирование:** проблемы с масштабируемостью системы, уязвимости в безопасности при обработке платежей.
   * **Функциональное тестирование:** некорректное отображение товаров в каталоге, невозможность оформить заказ.
   * **Нефункциональное тестирование:** низкая производительность под высокой нагрузкой, неудобный интерфейс пользователя.
2. **Выводы по работе:**
   * Различные виды тестирования позволяют выявить широкий спектр дефектов на разных уровнях разработки и эксплуатации программного обеспечения.
   * Применение методов статического и динамического тестирования обеспечивает всестороннюю проверку как функциональных, так и нефункциональных требований к системе.
   * Анализ и тестирование проекта TeaOnlineShop показали важность комплексного подхода к тестированию для обеспечения высокого качества ПО.
3. **Список использованных источников:**
   * <https://github.com/BaFon77/TeaOnlineShop>