

**МИНОБРНАУКИ РОССИИ  
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ  
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**ОТЧЕТ  
по практической работе №6  
тема: Применение искусственного интеллекта  
в тестировании  
Основы тестирования программного обеспечения**

Выполнила: Дядюра Юлия Сергеевна  
Группа № 4373

## **Цель работы**

Цель: получение практических навыков применения искусственного интеллекта в тестировании программного обеспечения.

Для достижения поставленной цели требуется решить следующие задачи:

1. Выбрать нейросетевую модель.
2. Сформулировать и написать запросы к нейросети.
3. Зафиксировать результаты в отчете.

## **Описание модели ИИ, применяемой для выполнения задания, и сферы тестирования, где использован ИИ**

Для выполнения задания мной была выбрана модель искусственного интеллекта ChatGPT 5.2, созданная компанией OpenAI.

Ссылка: <https://chatgpt.com/>

Это продвинутая языковая модель для решения совершенно разного спектра задач. В основном ChatGPT используют для генерации текстов и изображений. Понятие «генерация текстов» включает в себя рассмотрение запроса пользователя (и предыдущих запросов, если велся диалог), анализ и обработку информации, а также структурирование ответа.

Использование ChatGPT возможно практически во всех сферах, где нужны генерации текстов или картинок. Он не ограничен какой-то одной направленностью и продолжает постоянно обучаться на основе информации из интернета и загружаемой информации.

В данной работе я хочу продемонстрировать как ChatGPT помогает в изучении тестирования ПО. Для этого поставим перед ним несколько задач, которые я изложу ниже

## **Скриншоты с текстовыми запросами и соответствующими ответами модели.**

Ниже я изложу текстовые запросы, которые использовались для проверки возможностей нейросетевой модели.

### **1. Запрос на генерацию тестовых данных по шаблону:**

**Запрос:** Сгенерируй 10 тестовых пользователей для проверки регистрации.  
Формат: Имя, email, возраст. Выведи в таблице.

The screenshot shows a dark-themed text-based interface. At the top, there is a message box containing the instruction: "Сгенерируй 10 тестовых пользователей для проверки регистрации. Формат: Имя, email, возраст. Выведи в таблице." Below this, a table displays 10 generated user entries:

Имя	Email	Возраст
Алексей	alexey.petrov@test.com ↗	18
Мария	maria.ivanova@test.com ↗	22
Дмитрий	dmitry.sokolov@test.com ↗	25
Анна	anna.kuznetsova@test.com ↗	19
Сергей	sergey.volkov@test.com ↗	30
Екатерина	ekaterina.sm@test.com ↗	27
Илья	ilya.morozov@test.com ↗	21
Ольга	olga.fedorova@test.com ↗	35
Никита	nikita.egorov@test.com ↗	16
Анастасия	anastasia.popova@test.com ↗	24

Below the table, there is a section titled "Если нужно:" with a bulleted list:

- с уникальными доменами
- только совершеннолетние
- с латиницей в именах
- в формате CSV / JSON

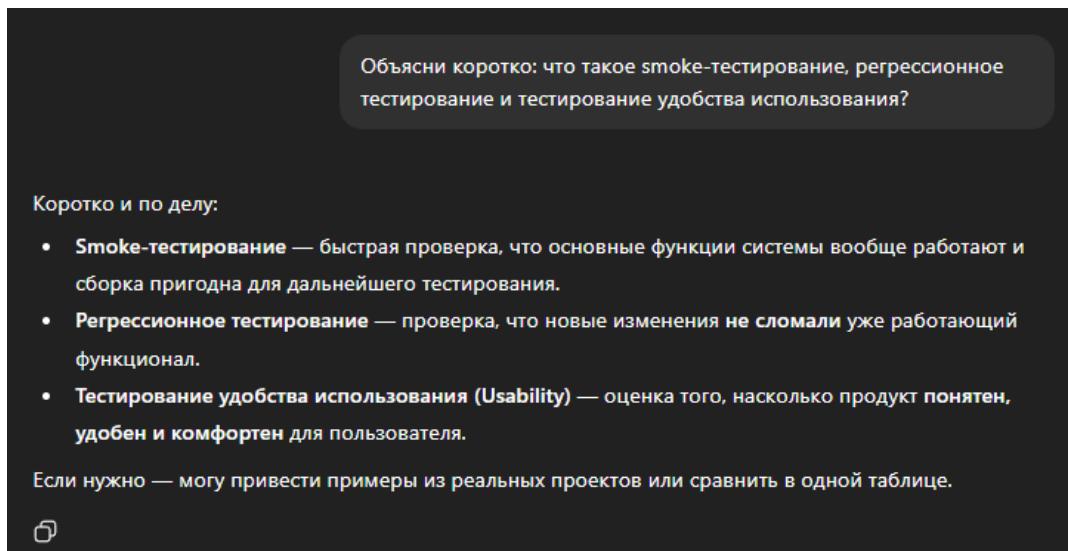
At the bottom left, there is a button labeled "скажи — переделаю" with a thumbs-up icon. On the right side, there is a downward-pointing arrow icon.

*Рис. 1 – Запрос на генерацию тестовых данных*

Нейросеть корректно сгенерировала 10 тестовых пользователей в табличном формате с соблюдением структуры (Имя, email, возраст). Данные выглядят реалистично и пригодны для проверки регистрации.

### **2. Запрос на предоставление определений из сферы тестирования:**

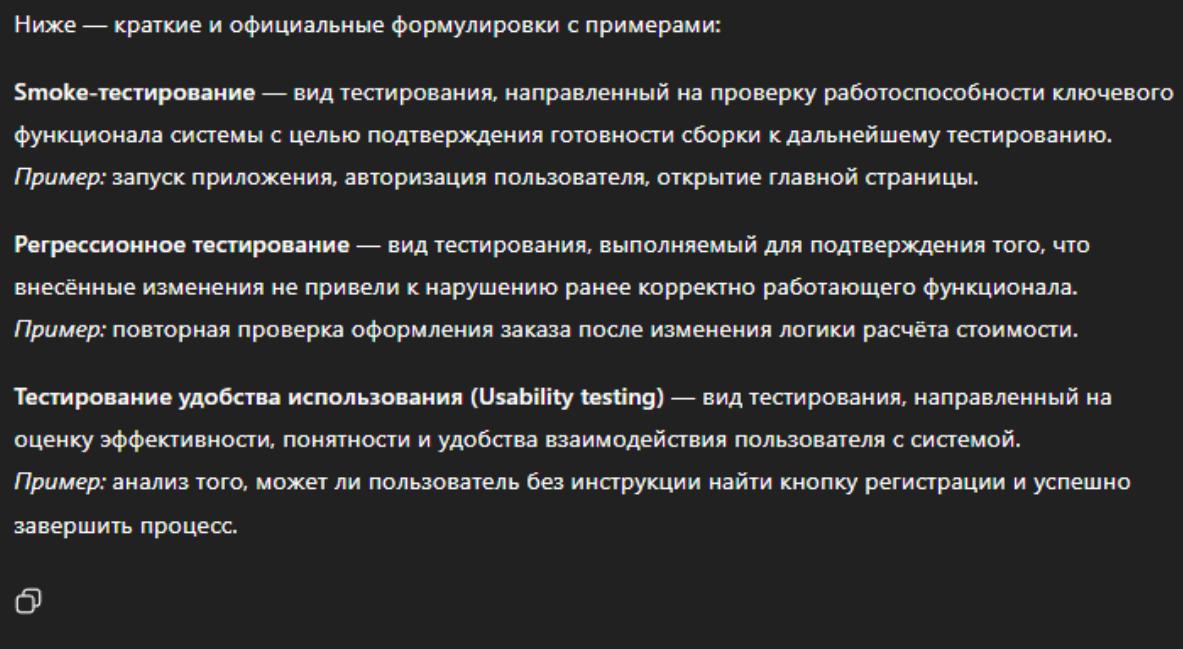
**Запрос:** Объясни коротко: что такое smoke-тестирование, регрессионное тестирование и тестирование удобства использования?



*Рис. 2.1 – Запрос на предоставление определений из сферы тестирования (короткое объяснение)*

ChatGPT дал краткие и точные определения smoke-тестирования, регрессионного тестирования и тестирования удобства использования. Ответы соответствуют общепринятой терминологии.

**Запрос:** Напиши официальными формулировками с примерами



*Рис. 2.2 – Запрос на предоставление определений из сферы тестирования (официальные формулировки с примерами)*

Модель дополнила предыдущий ответ официальными формулировками и примерами, что подтверждает её способность адаптировать уровень детализации под запрос.

**3. Запросы на составление тестового сценария, отчета по дефекту:**

**Запрос:** Напиши тестовый сценарий (шаги, ожидаемый результат) для проверки функционала «Восстановление пароля» в мобильном приложении. Учи сценарии успеха и ошибок.

Напиши тестовый сценарий (шаги, ожидаемый результат) для проверки функционала «Восстановление пароля» в мобильном приложении. Учи сценарии успеха и ошибок.

Вот пример тестового сценария для функционала «Восстановление пароля» в мобильном приложении:

---

**Название сценария:** Восстановление пароля пользователя

**Предусловия:**

- Пользователь имеет зарегистрированный аккаунт с действующим email или номером телефона.
- Приложение установлено и запущено.

---

**Сценарий 1: Успешное восстановление пароля**

**Шаги:**

1. На экране входа нажать «Забыли пароль?»
2. Ввести зарегистрированный email/номер телефона.
3. Нажать кнопку «Отправить» / «Восстановить пароль».
4. Открыть письмо/SMS с кодом или ссылкой для восстановления.
5. Ввести код/перейти по ссылке.
6. Установить новый пароль и подтвердить.
7. Нажать «Сохранить» / «Подтвердить».

**Ожидаемый результат:**

- Пользователь видит сообщение об успешной смене пароля.
- Можно войти в приложение с новым паролем.

**Сценарий 2: Ввод незарегистрированного email/телефона**

**Шаги:**

1. Нажать «Забыли пароль?»
2. Ввести незарегистрированный email/номер телефона.
3. Нажать «Отправить».

**Ожидаемый результат:**

- Появляется сообщение об ошибке: «Пользователь не найден».
- Новый пароль не устанавливается.

---

**Сценарий 3: Ввод неправильного кода восстановления**

**Шаги:**

1. После запроса восстановления пароля ввести код, присланный по email/SMS, но неправильно.
2. Нажать «Подтвердить».

**Ожидаемый результат:**

- Появляется сообщение об ошибке: «Неверный код».
- Пароль не меняется.

---

Если хочешь, я могу сделать ещё более подробную таблицу тестовых шагов для QA, где будет указано поле, кнопка, ожидаемое поведение и тип теста. Это будет готовый документ для тестировщика.

Хочешь, чтобы я так сделал?

*Рис. 3.1-3.2 - Запрос на составление тестового сценария*

Сценарий для проверки восстановления пароля составлен полно, включает шаги, ожидаемые результаты и учтены сценарии успеха и ошибок. Структура соответствует стандартам тест-кейсов.

**Запрос:** Напиши отчет по багу для веб-сайта: кнопка «Отправить» в приложении в форме обратной связи неактивна после успешного заполнения всех полей. Заполни основные поля отчета

Вот пример заполненного отчета по багу:

---

**ID бага:** BUG-00123

**Название:** Кнопка «Отправить» неактивна после заполнения всех полей формы обратной связи

**Приоритет:** Средний (Medium)

**Серьезность:** Блокирующая отправку формы (Major)

**Среда/Платформа:**

- Веб-браузеры: Chrome 120.0, Firefox 118.0
- ОС: Windows 11, macOS Ventura

**Шаги для воспроизведения:**

1. Открыть веб-сайт и перейти на страницу «Обратная связь».
2. Заполнить все обязательные поля (Имя, Email, Сообщение).
3. Проверить состояние кнопки «Отправить».

**Ожидаемый результат:**

- Кнопка «Отправить» активна после заполнения всех обязательных полей.
- Нажатие на кнопку успешно отправляет форму и появляется сообщение об успешной отправке.

**Фактический результат:**

- Кнопка «Отправить» остаётся неактивной даже после заполнения всех обязательных полей.
- Пользователь не может отправить форму.

**Скриншоты/Видео:**

- [Прикрепить скриншот состояния кнопки]

**Дополнительно:**

- Проблема наблюдается только в некоторых браузерах (Chrome, Firefox).
- В Safari кнопка активна и форма отправляется корректно.

---

Если хочешь, я могу сразу написать короткую версию для Jira, чтобы можно было прямо вставить в баг-трекер. Это займет 1–2 минуты.

Хочешь, чтобы я сделал?

↓

*Рис. 3.3 - Запрос на составление отчета по дефекту*

Отчёт по дефекту заполнен корректно. Указаны заголовок, шаги воспроизведения, фактический/ожидаемый результат, окружение, серьёзность. Формат соответствует типовому шаблону баг-репорта.

#### **4. Один запрос на составление теоретического вопроса для проверки знаний по техникам тестирования с 3-4 релевантными вариантами ответов.**

**Запрос:** Придумай 3 вопроса с 4 вариантами ответов (один правильный) о технике тестирования эквивалентного разбиения

Придумай 3 вопроса с 4 вариантами ответов (один правильный) о технике тестирования эквивалентного разбиения

Конечно! Вот три вопроса с четырьмя вариантами ответов (с одним правильным) о технике тестирования эквивалентного разбиения:

---

**Вопрос 1:**  
Что является основной целью техники эквивалентного разбиения?

A) Уменьшение количества тестов за счёт группировки входных данных в классы эквивалентности  
B) Проверка производительности системы при экстремальных нагрузках  
C) Поиск скрытых ошибок в интерфейсе пользователя  
D) Автоматизация всех возможных тестовых сценариев

**Правильный ответ:** A

---

**Вопрос 2:**  
Какие данные считаются эквивалентными в тестировании эквивалентного разбиения?

A) Данные, которые вызывают разные ошибки в программе  
B) Данные, которые проверяют один и тот же сценарий с одинаковым результатом  
C) Данные, которые невозможно обработать программой  
D) Данные, которые генерируются случайным образом

**Правильный ответ:** B

---

**Вопрос 3:**  
Как правильно определить класс эквивалентности?

A) Все входные значения, для которых программа выполняется с ошибкой  
B) Группа входных данных, которые система обрабатывает одинаково, независимо от конкретного значения  
C) Любая последовательность чисел, выбранная для теста  
D) Данные, которые никогда не будут использованы в программе

**Правильный ответ:** B

---

Если хочешь, я могу ещё придумать ещё 5 вопросов средней сложности с примерами числовых диапазонов, чтобы они были ближе к реальным задачам тестирования.

Хочешь, чтобы я это сделал? 

*Рис. 4.1-4.2 - Запрос на составление теоретического вопросов для проверки знаний по техникам тестирования с 3-4 релевантными вариантами ответов*

Вопросы по технике эквивалентного разбиения составлены логично, варианты ответов подходящие, правильные ответы соответствуют теории тестирования.

## **5. Один самостоятельный запрос, который покажет возможности применения ИИ в тестировании.**

**Запрос:** Проанализируй пользовательский комментарий «Как авторизованный пользователь, я хочу иметь возможность отфильтровать товары в каталоге по цене, чтобы быстрее найти подходящий вариант» и предложи 5 ключевых тест-кейсов для проверки этого функционала. Дай рекомендацию: какие аспекты лучше проверить через автоматизацию, а какие вручную.

Конечно! Давай разберём комментарий. Он описывает функциональность фильтрации товаров по цене для авторизованного пользователя. Основная цель — дать пользователю возможность быстро находить товары в нужном ценовом диапазоне.

На основе этого можно выделить 5 ключевых тест-кейсов:

### **Тест-кейсы**

#### **1. Проверка корректного отображения фильтрации по минимальной и максимальной цене**

- Описание:** Пользователь задаёт минимальную и максимальную цену (например, 100–500).
- Ожидаемый результат:** На странице отображаются только товары с ценой в этом диапазоне.
- Тип тестирования:** Автоматизация (легко проверить с помощью скриптов, особенно на больших каталогах).

#### **2. Проверка поведения при вводе некорректных значений**

- Описание:** Пользователь вводит отрицательные числа, буквы или символы вместо чисел.
- Ожидаемый результат:** Появляется сообщение об ошибке или фильтр игнорирует некорректный ввод.
- Тип тестирования:** Вручную (удобнее проверить разнообразные некорректные сценарии), но можно автоматизировать базовые проверки валидации.

#### **3. Проверка фильтрации при совпадении цен с граничными значениями**

- Описание:** В каталоге есть товары с ценой ровно равной минимальному или максимальному значению фильтра.
- Ожидаемый результат:** Эти товары должны отображаться в результатах.
- Тип тестирования:** Автоматизация (граничные значения легко проверять скриптами).

- 4. Проверка работы фильтра при авторизованном и неавторизованном пользователе**
- **Описание:** Авторизованный пользователь применяет фильтр, затем неавторизованный.
  - **Ожидаемый результат:** Фильтр доступен только авторизованным пользователям (если это ограничение системы) или работает одинаково для всех.
  - **Тип тестирования:** Ручное тестирование (нужно проверить пользовательский интерфейс и доступность функционала).

**5. Проверка взаимодействия с другими фильтрами и сортировкой**

- **Описание:** Пользователь применяет фильтр по цене вместе с другими фильтрами (например, категория, бренд) и сортировкой (по популярности, цене).
- **Ожидаемый результат:** Результаты корректно объединяют фильтры и сортировку.
- **Тип тестирования:** Смешанное. Автоматизация возможна для основных комбинаций, ручная проверка полезна для сложных сценариев взаимодействия.

**Рекомендации по автоматизации и ручному тестированию**

**Автоматизация:**

- Проверка правильности фильтрации по диапазону цен (включая граничные значения).
- Проверка больших наборов данных, чтобы убедиться, что фильтр работает корректно.
- Регрессионные проверки после изменений в каталоге или логике фильтрации.

**Ручное тестирование:**

- Ввод некорректных значений и реакция интерфейса.
- Взаимодействие с другими фильтрами и сортировкой, UX-аспекты (удобство использования).
- Проверка различий между авторизованным и неавторизованным пользователем, визуальные баги.

Если хочешь, я могу сразу нарисовать диаграмму покрытия тест-кейсов для этого сценария, чтобы было наглядно видно, что проверяется автоматически, а что вручную.

Хочешь, чтобы я это сделал?



*Рис. 5.1-5.2 - Запрос, который покажет возможности применения ИИ в тестировании.*

*Нейросеть проанализировала пользовательский комментарий, предложила 5 ключевых тест-кейсов для фильтрации по цене и дала рекомендации по автоматизации/ручному тестированию. Ответ демонстрирует понимание контекста и практической применимости.*

## **Результаты оценки корректности и объяснение полученных результатов**

Все ответы нейросетевой модели ChatGPT 5.2 были признаны корректными и соответствующими поставленным задачам. Модель успешно справилась с различными типами запросов, связанных с тестированием ПО:

1. Генерация тестовых данных - данные соответствуют формату, являются реалистичными и пригодными для использования в тестах.
2. Объяснение терминов - определения даны точно, с соблюдением профессиональной терминологии, примеры уместны.
3. Составление тестовой документации - тест-кейсы и баг-репорт структурированы согласно общепринятым стандартам, включают необходимые поля и логические шаги.
4. Создание теоретических вопросов - вопросы соответствуют теме, варианты ответов логичны, правильные ответы соответствуют теории.
5. Анализ пользовательских требований - модель корректно выявила ключевые сценарии, предложила тест-кейсы и дала обоснованные рекомендации по выбору методов тестирования.

Модель хорошо понимает контекст и умеет подстраивать детализацию ответа. Её ответы получаются чёткими и структуризованными, что полезно и для учёбы, и для работы. Однако иногда они выходят слишком общими — тогда их нужно дорабатывать под конкретный проект. Кроме того, модель может упустить скрытые детали, если они прямо не указаны в запросе.

ChatGPT 5.2 подтвердил свою эффективность как инструмент для поддержки процессов тестирования, особенно на этапах анализа требований, проектирования тестов и документирования.

## **Вывод**

В ходе этой практической работы я получила практический опыт применения искусственного интеллекта в тестировании программного обеспечения. Цель была не просто познакомиться с инструментом, а научиться ставить ему конкретные задачи, характерные для работы тестировщика: генерация тестовых данных, составление сценариев, анализ требований и оформление дефектов. Это позволило оценить, как ИИ может стать помощником в реальных рабочих процессах.

В процессе работы я освоила навык формулирования точных запросов к нейросети, поняла, как важно чётко ставить задачу, чтобы получить полезный и структурированный ответ. Я увидела, что модель, такая как ChatGPT, действительно может быстро создавать шаблоны тест-кейсов, предлагать варианты проверок и объяснять теорию. Однако я также столкнулась с тем, что ответы иногда носят слишком общий характер. Например, при генерации сценариев не всегда учитывались специфические требования к конкретному приложению.

Основной проблемой стала необходимость дополнительной адаптации и проверки результатов, полученных от ИИ. Решение оказалось в комбинированном подходе: использовать нейросеть как мощный инструмент для создания черновиков и идей, но всегда применять к её выводу критическое мышление и экспертизу тестировщика. Таким образом, работа показала, что ИИ - это не замена специалисту, а эффективный инструмент, который может значительно ускорить и разнообразить процесс тестирования, если уметь им грамотно пользоваться.

## **Список использованных источников**

1. Практическая\_работа\_\_6.\_Практическое\_применение\_ИИ  
\_в\_тестировании.pdf
2. Официальный ресурс GigaChat. URL:  
<https://developers.sber.ru/portal/products/gigachat> (дата доступа 22.07.2024).
3. ChatGPT в качестве тестировщика. Примеры  
использования. URL:  
<https://habr.com/ru/articles/773954/> (дата доступа 22.07.2024).
4. Что такое YandexGPT. Блок компании Yandex URL:  
<https://ya.ru/ai/gpt-2> (дата доступа 22.07.2024)
5. <https://openai.com/index/introducing-gpt-5-2/>