

МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение Высшего образования*

«МИРЭА - Российский технологический университет»

РТУ МИРЭА

Институт информационных технологий (ИИТ)

Кафедра Математического обеспечения и стандартизации информационных технологий (МОСИТ)

Отчет по выполнению практической работы №1 по дисциплине

«Тестирование и верификация программного обеспечения» Тема:

ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА МЕТОДОМ «ЧЕРНОГО ЯЩИКА»

Выполнили студенты: Карсанов А.Е.

Умаров А.Р. Арзуманян Д.А. Егоров Л.В.

Группа: ИКБО-13-23

Работу принял: Ильичев Г.П.

Москва 2025

# СОДЕРЖАНИЕ

1. [ЦЕЛЬ И ФОРМУЛИРВКА ЗАДАЧ 4](#_bookmark0)
2. [ТЕХНИЧЕСКОЕ ЗАДАНИЕ СОБСТВЕННОГО ПРОГРАММНОГО](#_bookmark1) [ПРОДУКТА 5](#_bookmark1)
   1. [Введение 5](#_bookmark2)
   2. [Основания для разработки 5](#_bookmark3)
   3. [Назначение разработки 6](#_bookmark4)
   4. [Требования к программе 6](#_bookmark5)
      1. [Функциональные требования 6](#_bookmark6)
      2. [Требования к надёжности 8](#_bookmark7)
      3. [Требования эксплуатации 8](#_bookmark8)
      4. [Требования к совместимости 9](#_bookmark9)
   5. [Требования к интерфейсу 9](#_bookmark10)
   6. [Критерии приёмки 10](#_bookmark11)
   7. [Требования к документации 10](#_bookmark12)
   8. [Порядок контроля и приёмки 11](#_bookmark13)
   9. [Этапы и сроки разработки 12](#_bookmark14)
3. [ОПИСАНИЕ ВНЕСЁННЫХ ОШИБОК 13](#_bookmark15)
4. [ТЕХНИЧЕСКОЕ ЗАДАНИЯ ВТОРОЙ КОМАНДЫ И ЕГО АНАЛИЗ 15](#_bookmark16)
5. [РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ ПРОГРАММНОГО ПРОДУКТА ВТОРОЙ](#_bookmark17) [КОМАНДЫ 28](#_bookmark17)
   1. [Тестирование 28](#_bookmark18)
   2. [Неуспешный вход в зарегистрированный аккаунт 28](#_bookmark19)
   3. [Потеря данных при перезагрузке страницы 30](#_bookmark20)
   4. [Некорректное отображение заполненности складов в общем списке 32](#_bookmark21)
   5. [Неактивная кнопка "Добавить склад" при пустом результате поиска 34](#_bookmark22)
   6. [Нефункциональная кнопка "Сбросить" в фильтрации товаров на складе 35](#_bookmark23)
   7. [Некорректный заголовок модального окна добавления товара 37](#_bookmark24)
   8. [Некорректное отображение общей загруженности складов в аналитике 38](#_bookmark25)
   9. Неправильный цвет индикатора для статуса "Заканчивается срок" 39
   10. Некорректное поведение раздела "Аналитика" при отсутствии товаров 41
6. [АНАЛИЗ ДОКУМЕНТАЦИИ ВТОРОЙ КОМАНДЫ 43](#_bookmark25)
7. [ЗАКЛЮЧЕНИЕ 45](#_bookmark26)

[ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ 48](#_bookmark27)

# ЦЕЛЬ И ФОРМУЛИРВКА ЗАДАЧ

Цель работы заключается в знакомстве студентов с процессом тестирования программного обеспечения, включая подготовку технической документации, выявление ошибок и их документирование. Развитие навыков командной работы при тестировании программного продукта.

Для достижения поставленной цели работы студентам необходимо выполнить ряд задач:

1. научиться разрабатывать программный продукт с учетом требований для дальнейшего тестирования;
2. овладеть методом тестирования «черного ящика»;
3. разработать и протестировать техническое задание и документацию;
4. научиться находить ошибки в чужом программном продукте и документировать их;
5. оформить итоговый отчет по проделанной работе в соответствии с установленными стандартами;
6. освоить работу с инструментами для управления тестированием и отслеживания багов;
7. изучить распределение ролей в команде тестировщиков.

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ СОБСТВЕННОГО ПРОГРАММНОГО ПРОДУКТА

## Введение

Медицинская информационная система (далее — Приложение) представляет собой веб-платформу, предназначенную для автоматизированного сбора, обработки, хранения и управления медицинскими данными пациентов. Система обеспечивает удобное взаимодействие между врачами, пациентами и администраторами, а также предоставляет интеллектуальные инструменты для извлечения данных из загружаемых архивов, версионирования записей и контроля доступа к информации.

## Основания для разработки

Разработка программного продукта инициирована в связи с необходимостью цифровизации и автоматизации работы с медицинскими архивами в условиях роста объёмов данных и требований к их точности, безопасности и доступности.

В настоящее время врачи и администраторы вынуждены вручную вводить данные пациентов, сопоставлять файлы с карточками, отслеживать изменения — что приводит к ошибкам, дублированию записей и высоким временным затратам. Отсутствие централизованной системы снижает эффективность работы и доверие со стороны пациентов к сохранности и актуальности их данных.

## Назначение разработки

## Разрабатываемая система призвана решить следующие задачи:

1. Автоматизация обработки медицинских архивов — извлечение ФИО, даты рождения, контактов и даты исследования без ручного ввода.
2. Сокращение временных и трудовых затрат врачей и администраторов при регистрации пациентов и загрузке данных.
3. Повышение точности и целостности данных за счёт интеллектуального сопоставления и предотвращения дублей.
4. Обеспечение прозрачности и контроля — через систему версионирования записей и журнал аудита изменений.
5. Упрощение доступа к истории болезни — для врачей и пациентов через персонализированные интерфейсы.
6. Гибкое управление ролями и доступом — с возможностью назначения прав, подтверждения ролей и администрирования пользователей.
7. Формирование отчётов и логов — для администраторов по действиям пользователей, загрузкам, изменениям в записях.

## Требования к программе

#### Функциональные требования

1) Главное меню / Авторизация и навигация

* Возможность входа в систему для пользователей с ролями: Врач, Пациент, Администратор.
* Перенаправление на персонализированный Dashboard в зависимости от роли.
* Доступ к основным разделам: загрузка архивов, список пациентов, административная панель.
* Возможность регистрации нового пользователя (пациента) с обязательным вводом e-mail и пароля.

2) Управление пользователями и ролями

* Отображение списка пользователей (для администратора).
* Назначение и отзыв ролей (врач, администратор).
* Подтверждение запросов на роль «врач» администраторами или другими врачами.
* Настройка доступа к данным пациентов по ролям.

3) Загрузка и обработка архивов

* Выбор и загрузка архива (drag & drop или через кнопку).
* Автоматическое извлечение данных: ФИО, дата рождения, контакты, дата исследования.
* Возможность ручного редактирования извлечённых данных перед сохранением.
* Проверка на дубликаты пациентов (по схожести ФИО ≥95%) с запросом подтверждения объединения.
* Ограничение: загрузка данных только одного пациента за одну операцию.

4) Работа с карточкой пациента

* Просмотр основной информации: ФИО, дата рождения, контакты.
* Просмотр истории болезней с возможностью добавления, редактирования и удаления записей.
* Просмотр файлов (галерея медицинских изображений, документов).
* Просмотр журнала аудита изменений по пациенту.
* Поддержка версионирования записей — просмотр и откат к предыдущим версиям (только для администратора).

5) Административная панель

* Управление пользователями: просмотр, назначение/отзыв ролей.
* Просмотр и управление запросами на роль «врач».
* Просмотр логов аудита: фильтрация по дате, пользователю, типу действия.
* Управление глобальными настройками системы (если предусмотрено).

6) Поиск и фильтрация

* Поиск пациентов по ФИО в списке.
* Фильтрация по дате рождения, дате последнего визита.
* Пагинация в таблицах при большом количестве записей.

#### Требования к надёжности

1. Программа должна обеспечивать восстановление после сбоев без потери критических данных (пациенты, архивы, истории болезней, логи аудита).
2. Должна быть реализована система журналирования (аудит) всех действий пользователей: загрузка, изменение, удаление, назначение ролей.
3. При разрыве соединения с сервером или СУБД — система должна автоматически пытаться восстановить соединение и уведомлять администратора при длительной недоступности.
4. В случае ошибок при обработке архивов — система должна сохранять частично обработанные данные, фиксировать ошибку в логах и предоставлять пользователю возможность повторной загрузки или ручного ввода.
5. Все изменения в записях пациента должны сохраняться с версионированием, чтобы обеспечить возможность отката.

#### Требования эксплуатации

1. Среда функционирования: веб-сервер с доступом к сети Интернет.
2. Программа должна работать в среде Linux (Ubuntu 20.04 и выше), допускается развёртывание через Docker-контейнеры.
3. Необходимо наличие стабильного подключения к базе данных (PostgreSQL или аналогичной СУБД).
4. Среднее количество одновременных пользователей: до 1 000 (с пиками до 5 000).
5. Система должна поддерживать режим эксплуатации 24/7 без необходимости ручного перезапуска.
6. Обязательна поддержка HTTPS и защита персональных данных в соответствии с требованиями законодательства (например, ФЗ-152 в РФ или GDPR в ЕС).

#### Требования к совместимости

1. Поддержка современных браузеров: Chrome, Firefox, Safari, Edge (актуальные версии).
2. Совместимость с операционными системами: Windows, macOS, Linux, Android, iOS (через браузер).
3. Возможность интеграции с внешними системами хранения файлов (S3, MinIO, NFS и т.п.).
4. Совместимость с DICOM-совместимыми системами (в перспективе) для прямого импорта медицинских изображений.
5. Поддержка REST API для будущей интеграции с мобильными приложениями или внешними сервисами.
6. Совместимость с LDAP/SSO для корпоративных внедрений (опционально).

## Требования к интерфейсу

#### Главное меню

Кнопка «Войти» — открывает форму авторизации с полями:

* E-mail (обязательное, с валидацией формата)
* Пароль (обязательное, скрытый ввод). После успешного входа — перенаправление на Dashboard в зависимости от роли.

Кнопка «Регистрация» — открывает форму регистрации пациента с полями:

* E-mail (обязательное, проверка на уникальность)
* Пароль (обязательное)
* Подтверждение пароля (обязательное, сравнение с паролем). После регистрации — автоматический вход и переход на Dashboard пациента.

Пункт «Dashboard» — отображается после входа и перенаправляет пользователя на главную страницу его роли:

* Для врача — Dashboard с карточками: «Загрузить архив», «Список пациентов», «Запросы на подтверждение ролей».
* Для пациента — Dashboard с карточками: «Мои записи», «Текущие проверки».
* Для администратора — Dashboard с карточками: «Все пользователи», «Заявки на роль», «Операции аудита».

Пункт «Административная панель» — доступен только пользователям с ролью Admin. Перенаправляет на /admin с боковым меню управления.

#### Управление каналами

Раздел «Пользователи» (в админ-панели) — отображает таблицу всех пользователей с колонками:

* ФИО / E-mail
* Роль (Patient / Doctor / Admin)
* Дата регистрации
* Действия: кнопки «Назначить роль», «Отозвать роль», «Просмотреть активность».

Кнопка «Назначить роль “врач”» — открывает модальное окно с multi-select списком пользователей, которым можно назначить роль врача. После выбора — кнопка «Подтвердить назначение».

Раздел «Запросы на роль “врач”» — список ожидающих подтверждения запросов от пользователей. Каждая запись содержит:

* Имя пользователя
* Дата запроса
* Кнопки: «Подтвердить», «Отклонить».

#### Процесс создания розыгрыша

Кнопка «Загрузить архив» — открывает страницу /upload с интерфейсом загрузки.

Обязательные элементы интерфейса:

* Заголовок: «Загрузить архив пациента»
* Поле выбора файла: поддержка drag & drop и кнопки «Выбрать файл»
* Кнопка «Загрузить» (primary, активна только после выбора файла)

После загрузки файла:

* Отображается индикатор прогресса обработки.
* Если обнаружен похожий пациент — модальное окно с предложением: «Объединить с существующим?» + выбор из списка.
* Появляется блок с автоматически распознанными данными:
  + ФИО (редактируемое поле)
  + Дата рождения (редактируемое поле, с валидацией формата)
  + Контакты (редактируемое поле)
  + Дата исследования (редактируемое поле, с валидацией диапазона)

Кнопка «Сохранить и продолжить» — фиксирует данные и перенаправляет на страницу статуса обработки (/upload/status/{job\_id}).

#### Меню информации о конкретном розыгрыше

Отображаемая информация в шапке карточки:

* ФИО (жирным шрифтом)
* Дата рождения
* Контактные данные (телефон, e-mail)
* Кнопки: «Редактировать данные», «Добавить запись»

Tab-навигация:

* История болезней — список записей с датами, кратким описанием. Каждая запись имеет кнопки: «Просмотреть», «Редактировать», «Удалить».
* Файлы — галерея загруженных медицинских файлов (изображения, PDF, видео). Поддержка превью и скачивания.
* Аудит — журнал изменений: время, пользователь, действие, детали.

При клике на запись в «Истории болезней» — открывается модальное окно со списком версий записи.

* Отображается дата и автор каждой версии.
* Кнопка «Откатить к этой версии» (доступна только администратору) — с обязательным подтверждением.

#### Административная панель

Боковое меню с разделами:

* Пользователи
* Запросы ролей
* Пациенты
* Истории болезней
* Логи аудита

Раздел «Логи аудита» — таблица с колонками:

* Время события
* Пользователь (ФИО / роль)
* Действие (например: «Загрузил архив», «Изменил запись пациента»)
* Сущность (например: «Пациент #123», «Запись #45»)
* Детали изменений (краткое описание или diff)

Фильтры вверху таблицы:

* По дате (календарный выбор)
* По пользователю (поиск по ФИО или e-mail)
* По типу действия (выпадающий список: загрузка, редактирование, удаление и т.д.)

#### Процесс присоединения к розыгрышу

Поисковая строка вверху страницы — поиск по ФИО (с debounce 500 мс, обновление таблицы в реальном времени).

Фильтры (раскрывающийся блок):

* Дата рождения (диапазон)
* Дата последнего визита (диапазон)

Таблица результатов:

* Колонки: ФИО, Дата рождения, Дата последнего визита, Количество записей, Кнопка «Подробнее»
* Кнопка «Подробнее» — перенаправляет на страницу пациента (/patients/{id})

Пагинация внизу таблицы — по 10/25/50 записей на страницу, с возможностью перехода по страницам.

## Критерии приёмки

Приёмка проекта будет считаться успешной при выполнении следующих условий:

1. Корректная работа основных функций, описанных в разделе 2.4.1:
2. Сохранность данных пациентов, архивов и логов аудита при сбоях системы, перезапуске сервера или СУБД.
3. Корректная работа интерфейса и API без ошибок при взаимодействии между фронтендом, бэкендом и базой данных.
4. Поддержка эксплуатации 24/7 в течение не менее 30 календарных дней без критических сбоев (ошибок уровня 5xx, потери данных, недоступности ключевых функций).
5. Соответствие пользовательского интерфейса и поведения системы описанию, представленному в разделе «Требования к программе» (2.4) и «Требования к интерфейсу» (2.5).

## Требования к документации

Для обеспечения возможности запуска и использования проекта должна быть предоставлена следующая документация:

1. README.md: (https://readme.md/) Должна содержать инструкцию по установке и запуску проекта (клонирование, создание виртуального окружения, установка зависимостей, применение миграций, запуск сервера).
2. Код: Код должен содержать комментарии в неочевидных местах для облегчения понимания его логики.

## Порядок контроля и приёмки

Приёмка будет включать в себя следующие этапы тестирования:

1. Заказчик (преподаватель) выполняет клонирование репозитория и запуск приложения согласно инструкции в README.md. (https://readme.md/).
2. Производится визуальная проверка работы всего заявленного функционала вручную.
3. Результаты тестирования и демонстрации фиксируются

## Этапы и сроки разработки

Предлагаемый план разработки и тестирования выглядит следующим образом:

Этап 1: Подготовка окружения и настройка проекта – 1 день.  
Развёртывание сервера, инициализация репозитория, установка зависимостей, настройка подключения к СУБД (PostgreSQL), создание .env-файла с базовыми настройками.

Этап 2: Реализация моделей данных и настройка административной панели – 1 день.  
Создание структуры базы данных: пользователи, пациенты, медицинские записи, архивы, логи аудита. Настройка базового интерфейса администратора с доступом к управлению пользователями и ролями.

Этап 3: Реализация системы аутентификации и авторизации – 1 день.  
Реализация регистрации пациентов по e-mail и паролю. Реализация входа для всех ролей (Patient, Doctor, Admin). Разграничение доступа к разделам системы в зависимости от роли.

Этап 4: Реализация CRUD-функционала для пациентов и записей – 2 дня.  
Создание, просмотр, редактирование и удаление карточек пациентов. Добавление, редактирование и удаление медицинских записей. Реализация версионирования записей с возможностью отката (для администратора).

Этап 5: Реализация загрузки и обработки архивов + интерфейс Dashboard’ов – 2 дня.  
Реализация загрузки архива с автоматическим извлечением ФИО, даты рождения, контактов и даты исследования. Добавление ручного редактирования данных и логики сопоставления с существующими пациентами. Разработка персонализированных Dashboard’ов для каждой роли.

Этап 6: Написание тестов и отладка – 1 день.  
Функциональное тестирование основных сценариев: регистрация, вход, загрузка архива, просмотр пациента, назначение роли, аудит изменений. Проверка восстановления после сбоев. Исправление критических ошибок.

# ОПИСАНИЕ ВНЕСЁННЫХ ОШИБОК

В процессе разработки целенаправленно в проект было внесено определённое количество ошибок, которые можно разделить на 3 категории:

* Функциональная (Логическая) — ошибка в бизнес-логике приложения (серверная или клиентская валидация, обработка данных).
* Оформление — ошибка в оформлении, которая ухудшает опыт использования интерфейса конечным пользователем (неправильные подписи, заголовки, формулировки).
* Интерфейсная — проблема, связанная с отображением информации для пользователя, вызванная ошибкой в отправке или обработке данных.
* Производительность — задержки и скорость работы (в данном случае не применимо к внесённым ошибкам).

Представим в таблице 3.1 все внесённые ошибки, которые нарушает ранеее разработанное техническое задание.

*Таблица 3.1 – Описание внесённых ошибок в собственное ПО*

|  |  |  |
| --- | --- | --- |
| **Название ошибки** | **Тип ошибки** | **Способ обнаружения** |
| Нет валидации e-mail на фронтенде | Логическая | 1. Перейти на страницу регистрации 2. Ввести “abc” в поле e-mail 3. Отправить форму — ошибка только на бэкенде |
| «Отчество» внезапно обязательное | Оформление | 1. Начать регистрацию пациента 2. Поле “Отчество” подсвечено как обязательное, хотя в ТЗ не указано |

*Таблица 3.1 – Продолжение*

|  |  |  |
| --- | --- | --- |
| Ошибка валидации номера телефона | Логическая | 1. Ввести “123” или “abc” в поле телефона при регистрации 2. Форма отправляется без предупреждения |
| Неправильный заголовок на странице | Оформление | 1. Авторизоваться под учётной записью пациента  2. Увидеть приветствие “Welcome, admin” на Dashboard |

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ ВТОРОЙ КОМАНДЫ И ЕГО АНАЛИЗ

Ниже приведено техническое задание, разработанное другой командой, и комментарии с анализом. Исходное форматирование и структура сохранены комментарии выделены

### ВВЕДЕНИЕ

* 1. Наименование программного продукта

Десктопное приложение «Судоку» (далее — Приложение).

* 1. Краткая характеристика области применения

Приложение представляет собой цифровую реализацию классической логической головоломки «Судоку», разработанную для платформ Windows, macOS и Linux. Оно предназначено для индивидуального использования и позволяет пользователю решать уникальные головоломки различного уровня сложности в удобном графическом интерфейсе с поддержкой клавиатурного и мышечного управления, визуальной подсветкой и автоматической проверкой корректности ввода.

* 1. Назначение и цель создания проекта

Целью проекта является создание кроссплатформенного, стабильно работающего и интуитивно понятного приложения для решения головоломок «Судоку», соответствующего современным стандартам UX/UI и техническим требованиям к desktop-приложениям. Приложение разработано в учебных целях для демонстрации навыков:

* проектирования архитектуры GUI-приложений;
* реализации алгоритмов генерации и валидации судоку;
* работы с событийной моделью ввода (клавиатура, мышь);
* обеспечения кроссплатформенной совместимости;
* тестирования и отладки ПО на соответствие ТЗ.

Проект служит основой для дальнейшего расширения функционала (например, добавление уровней сложности, сохранения прогресса, статистики, сетевой игры).

### АНАЛИЗ

Соответствие шаблону: полное.

Содержание: Корректное, но требует расширения

* Указано наименование продукта — «Судоку».
* Описана область применения — цифровая головоломка для индивидуального использования.
* Чётко сформулирована цель — создание учебного кроссплатформенного приложения с демонстрацией ключевых навыков разработки.
* Упомянуты ключевые технологии и функции: генерация поля, валидация, управление, кроссплатформенность.

Замечания: отсутствует описание целевой аудитории (студенты, преподаватели, начинающие разработчики). Не сформулирована решаемая пользовательская или бизнес-проблема. Недостаточно раскрыта мотивация создания именно desktop-приложения.

Рекомендации: добавить описание целевой аудитории, формулировать пользовательскую проблему, уточнить выбор платформы.

### ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

* 1. Основания для проведения работ

Разработка приложения «Судоку» осуществляется в рамках выполнения практического задания по учебному курсу, связанному с разработкой desktop-приложений на Python, изучением событийно-ориентированного программирования, работы с графическим интерфейсом (GUI), валидацией пользовательского ввода и кроссплатформенной совместимостью. Проект направлен на закрепление навыков проектирования, реализации и тестирования программного обеспечения, соответствующего техническому заданию и требованиям пользовательского опыта.

* 1. Исходные документы и нормативные ссылки
     + Методические указания к курсу.
     + Методические указания и техническое задание к курсовому проекту.
     + Официальная документация по выбранным библиотекам GUI (в зависимости от реализации):
       - Для Tkinter: https://docs.python.org/3/library/tkinter.html
       - Для PyQt6: https://www.riverbankcomputing.com/static/Docs/PyQt6/
       - Для Kivy: https://kivy.org/doc/stable/
     + Рекомендации по UX для desktop-приложений (для обеспечения интуитивности и подсветки):

### АНАЛИЗ

Соответствие шаблону: полное.

Содержание корректное и адекватное учебному проекту:

* Указано, что разработка ведётся в рамках учебного курса — это главное основание.
* Приведены ссылки на официальную документацию по ключевым технологиям (GUI-фреймворкам).
* Упомянуты методические материалы — как основной регулятор требований

Замечания: отсутствуют нормативные стандарты (ГОСТ, ISO), Не указан конкретный GUI-фреймворк, Нет ссылки на ТЗ как на основной регулирующий документ

Рекомендации: добавить ссылку на техническое задание как на основной регулирующий документ, уточнить выбранный GUI-фреймворк, добавить ссылку на PEP 8.

### НАЗНАЧЕНИЕ РАЗРАБОТКИ

Разрабатываемое Приложение призвано решить следующие задачи:

* Предоставление цифрового аналога головоломки «Судоку»: Создание интуитивно понятного, визуально ясного и удобного интерфейса для решения логических головоломок без использования бумаги и карандаша.
* Обеспечение интерактивной обратной связи в реальном времени: Автоматическая проверка корректности ввода цифры (с подсветкой правильных/неправильных значений) и визуальное выделение активной ячейки для улучшения пользовательского опыта.
* Генерация уникальных игровых полей: Реализация алгоритма, способного генерировать новые, неповторяющиеся сетки судоку с единственным решением при каждом запуске новой игры.
* Поддержка автономной и безопасной работы: Обеспечение функционирования приложения без интернет-соединения и передачи/хранения персональных данных пользователя.
* Учебная демонстрация: Создание полноценного desktop-приложения, иллюстрирующего ключевые концепции разработки ПО: событийная модель ввода, валидация данных, работа с GUI, кроссплатформенность, тестирование на соответствие ТЗ.
* Целевой показатель: Время отклика интерфейса на любое действие пользователя (ввод цифры, переход между ячейками) не должно превышать 100 мс.

### АНАЛИЗ

Соответствие шаблону: полное.

Содержание:

* + - Задачи охватывают как пользовательские потребности (удобство, обратная связь, автономность), так и учебные цели (демонстрация навыков).
    - Указан ключевой KPI — время отклика ≤100 мс, что прямо взято из ТЗ и критично для UX.
    - Упомянуты технические аспекты: генерация, валидация, кроссплатформенность.

Замечания: отсутствуют дополнительные метрики эффективности, не указаны метрики стабильности.

Рекомендации: добавить дополнительные KPI для более полной оценки качества, добавить метрику стабильности, добавить метрику UX:

### ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

* + 1. Модуль игрового поля (Game Grid)
       - FR1.1: Генерация уникального игрового поля 9×9 с единственным решением при каждом запуске новой игры.
       - FR1.2: Возможность вставки цифры (только 1–9) в любую пустую ячейку через клик мышью или клавиатурный ввод.
       - FR1.3: Автоматическая проверка конфликтов: если введённая цифра нарушает правила (повтор в строке, столбце или блоке 3×3) — подсветка ячейки красным
       - FR1.4: При корректном вводе — подсветка ячейки синим.
       - FR1.5 Ввод цифр 0, букв или спецсимволов игнорируется — приложение не реагирует и не крашится.Модуль статей (Articles)
       - FR1.6: При полном и корректном заполнении сетки — отображение модального окна с сообщением «Судоку решено! Поздравляем!» и кнопкой «Новое судоку».
    2. Модуль комментариев (Админ-панель)
       - FR2.1: Возможность перемещения между ячейками с помощью клавиш-стрелок (←↑→↓) и клавиши Tab.
       - FR2.2: Активная ячейка (в фокусе) должна визуально подсвечиваться (например, изменение фона или рамки).
       - FR2.3: Панель управления с кнопками цифр 1–9, дублирующая клавиатурный ввод.
       - FR2.4: Клик мышью по ячейке устанавливает на неё фокус и подсвечивает.
    3. Модуль интерфейса и меню (UI & Menu)
       - FR3.1: Главное меню с кнопками:
* «Новая игра» — запуск новой сессии с уникальным полем.
* «Правила» — открытие модального окна с текстом правил игры.
  + - * FR3.2: Игровое поле визуально разделено на 9 блоков 3×3 с помощью более жирных границ.
      * FR3.3: Минималистичный и современный стиль интерфейса (чистые линии, контрастные цвета, удобная типографика).
      * FR3.4: Все элементы интерфейса должны реагировать мгновенно — время отклика ≤100 мс.
  1. Требования к надежности и безопасности
* NR1: Приложение не должно аварийно завершаться при любом пользовательском вводе (включая случайные символы, 0, клики вне поля).
* NR2: Все данные хранятся локально — отсутствует передача или сбор персональных данных.
* NR3: Приложение работает автономно — не требует интернет-соединения.
* NR4: Обработка всех возможных исключений (например, при генерации поля, вводе, навигации) — без крашей, с логированием (если реализовано).
  1. Условия эксплуатации
* ER1: Язык программирования: Python 3.9+.
* ER2: GUI-фреймворк: Tkinter (встроен в Python) / PyQt6 / Kivy (указать используемый).
* ER3: Ключевые зависимости (если не Tkinter):
* Для PyQt6: PyQt6==6.7.0
* Для Kivy: kivy==2.3.0, kivy\_deps.sdl2==0.8.0
* ER4: База данных: не требуется (все данные — в памяти).
* ER5: Операционные системы: Windows 10/11, macOS (последние версии), Linux (Ubuntu LTS, Fedora).
* ER6: Режим запуска: нативное desktop-приложение, запускаемое через .exe, .app или исполняемый скрипт Python.
  1. Требования к совместимости
     + CR1: Совместимость с целевыми ОС.
     + CR2: Адаптивность интерфейса.
     + CR3: Интеграция со сторонними системами не требуется.

### АНАЛИЗ

Соответствие шаблону: полное.

Содержание: Детализированное и соответствующее ТЗ — все ключевые функции из технического задания (генерация уникальных полей, автоматическая проверка конфликтов, визуальная подсветка корректных/некорректных вводов, управление с клавиатуры и мыши, интуитивный интерфейс с меню и панелью цифр) чётко сформулированы как функциональные и нефункциональные требования. Учтены все 7 выявленных ошибок: каждая из них теперь отражена в требованиях — например, игнорирование ввода 0 и букв (FR1.5), защита от крашей (NR1), обязательная подсветка активной ячейки (FR2.2), исправленный текст сообщения о победе (FR1.6), работа кнопки «Новое судоку» (FR1.6), уникальность полей (FR1.1). Чётко прописаны ограничения: ввод разрешён только для цифр 1–9, все остальные символы игнорируются, приложение не должно аварийно завершаться. Указаны измеримые метрики, в частности, критически важное требование к времени отклика интерфейса — не более 100 мс (FR3.4, NR4).

Замечания: отсутствует детальное описание алгоритма генерации — не указано, как именно обеспечивается уникальность поля и гарантия единственного решения, что может привести к реализации “ленивой” или шаблонной генерации, как в ошибке TC-006; не описаны сценарии восстановления после сбоев — например, что должно происходить, если генерация поля зависла, произошёл таймаут или внутренняя ошибка; отсутствует требование к производительности самого процесса генерации — хотя указано время отклика UI, не зафиксировано допустимое время создания нового поля, что важно для UX.

Рекомендации: добавить функциональное требование FR1.7 — «Алгоритм генерации поля должен гарантировать, что каждое новое поле имеет уникальную комбинацию заполненных клеток и единственное решение. Повторение полей между сессиями недопустимо»; добавить нефункциональное требование NR5 — «Время генерации нового игрового поля не должно превышать 2 секунды на целевых системах»; добавить требование по обработке зависаний NR6 — «При превышении времени генерации (более 3 секунд) — отображать уведомление “Ошибка генерации. Попробуйте снова” и предлагать пользователю повторный запуск новой игры»; опционально — добавить требование по локализации интерфейса, если в будущем планируется поддержка нескольких языков (например, английский, испанский), что повысит универсальность продукта.

### ТРЕБОВАНИЯ К ИНТЕРФЕЙСУ

* 1. Структура пользовательского интерфейса
* UI1: Главное меню приложения содержит две кнопки:
* «Новая игра» — запускает генерацию и отображение нового игрового поля.
* «Правила» — открывает модальное окно с текстом правил игры.
* UI2: Игровое поле представляет собой сетку 9×9, визуально разделённую на 9 блоков 3×3 с помощью более жирных границ. Каждая ячейка — квадратная, с выравниванием по центру.
* UI3: Под игровым полем расположена панель управления с 9 кнопками, пронумерованными от 1 до 9, для ввода цифр в активную ячейку.
* UI4: При выборе пустой ячейки (мышью или клавиатурой) она визуально подсвечивается (например, изменением фона на светло-серый или добавлением рамки).
* UI5: При вводе цифры:
* Если цифра корректна — ячейка подсвечивается синим.
* Если цифра нарушает правила — ячейка подсвечивается красным.
* UI6: При успешном решении головоломки открывается модальное окно с сообщением:
* Текст: «Судоку решено! Поздравляем!»
* Кнопка: «Новое судоку» — закрывает окно и запускает новую игру.
  1. Визуальное представление и юзабилити
* UI7: Стиль интерфейса — минималистичный и современный:
* Чистые линии, нейтральные цвета фона (белый/светло-серый).
* Цифры в ячейках — крупные, чёткие, контрастные (чёрный/тёмно-серый).
* Подсветка активной ячейки — ненавязчивая, но заметная.
* UI8: Все интерактивные элементы (кнопки, ячейки) должны иметь мгновенный визуальный отклик (≤100 мс) — изменение цвета, тени или рамки при наведении/фокусе.
* UI9: Язык интерфейса — русский (в соответствии с ТЗ и сообщением о победе).
* UI10: Приложение должно корректно масштабироваться под разные разрешения экранов (от 1280×720 и выше) — элементы не должны наезжать друг на друга, текст — оставаться читаемым.

### АНАЛИЗ

Соответствие шаблону: полное.

Содержание: Детализированное и соответствующее ТЗ — все ключевые элементы интерфейса из технического задания описаны как требования: структура меню, визуальное разделение блоков 3×3, панель управления цифрами, подсветка активной ячейки, цветовая индикация корректности ввода, модальное окно победы с кнопкой «Новое судоку». Учтены ошибки TC-002 (подсветка), TC-003 (правила), TC-007 (текст поздравления). Указаны принципы юзабилити: мгновенный отклик, минималистичный стиль, язык интерфейса.

Замечания: внешний вид элементов описан обобщённо — нет конкретных требований к цветам, шрифтам, размерам кнопок или отступам; отсутствуют требования к доступности (например, контрастность для пользователей с нарушениями зрения, поддержка screen reader для слабовидящих, навигация только с клавиатуры); не указано поведение интерфейса при изменении системного масштаба (DPI) на Windows/macOS.

Рекомендации: уточнить визуальные спецификации — например, «цвет подсветки активной ячейки: #E0E0E0», «шрифт цифр: Arial Bold 18pt», «размер кнопки на панели: 40×40 пикселей»; добавить требования к доступности — «контрастность текста ≥ 4.5:1».

### КРИТЕРИИ ПРИЁМКИ

Приемка проекта будет считаться успешной при выполнении следующих условий:

* 1. Критерии функциональной полноты
* CP1: Реализован весь функциональный и нефункциональный функционал, описанный в разделах 4.1–4.4 (генерация уникальных полей, проверка конфликтов, подсветка, управление клавиатурой/мышью, интерфейс, безопасность, совместимость).
* CP2: Приложение корректно реагирует на все действия пользователя:
* Ввод цифр 1–9 — принимается и проверяется.
* Ввод 0, букв, спецсимволов — игнорируется, без крашей.
* Навигация стрелками — перемещает фокус и подсвечивает ячейку.
* Клик по ячейке — устанавливает фокус.
* CP3: При полном и корректном заполнении сетки:
* Отображается модальное окно с текстом «Судоку решено! Поздравляем!».
* Кнопка «Новое судоку» закрывает окно и запускает новую игру с уникальным полем.
* CP4: Кнопка «Правила» открывает окно с читаемым текстом правил игры (не пустое).
  1. Критерии качества кода и тестирования
* CP5: Все приёмочные тесты (TC-001 – TC-007) выполняются с результатом Passed (100% успешных):
* TC-001: Приложение не крашится при вводе случайных символов.
* TC-002: Активная ячейка подсвечивается при навигации клавиатурой.
* TC-003: Окно «Правила» содержит текст с описанием игры.
* TC-004: Цифра 0 не вставляется в ячейку.
* TC-005: Кнопка «Новое судоку» запускает новую игру.
* TC-006: Каждое новое поле уникально — не повторяет предыдущие.
* TC-007: Сообщение о победе написано без ошибок — «Поздравляем!».
* CP6: Приложение запускается на всех целевых ОС (Windows 10/11, macOS, Ubuntu LTS, Fedora) без ошибок и крашей.
* CP7: Время отклика интерфейса на любое действие (ввод, навигация, клик) ≤100 мс — измерено инструментально или подтверждено визуально.
* CP8: Время генерации нового игрового поля ≤2 секунд — измерено на эталонной системе (например, Intel i5 / 8 ГБ RAM).

### АНАЛИЗ

Соответствие шаблону: частичное.

Содержание: Полное и соответствующее ТЗ — определены чёткие критерии функциональной полноты (CP1–CP4), покрывающие все требования и исправления ошибок; введены критерии тестирования (CP5–CP6), включая прохождение всех 7 приёмочных тестов; добавлены количественные метрики производительности (CP7, CP8), что напрямую соответствует требованиям ТЗ (время отклика <100 мс) и рекомендациям по улучшению.

Замечания: отсутствуют критерии стабильности — например, успешный запуск 10+ игр подряд без утечек памяти или зависаний; не указаны критерии доступности (например, прохождение базовых тестов с screen reader); нет требования к отсутствию предупреждений/логов ошибок в консоли при штатной работе.

Рекомендации: добавить критерий CP9 — «Приложение успешно проходит 10 последовательных циклов “новая игра → решение → новое судоку” без увеличения потребления памяти более чем на 10% и без зависаний»; добавить критерий CP10 — «Интерфейс соответствует базовым стандартам доступности WCAG 2.1 Level A (контрастность, навигация с клавиатуры, семантическая разметка — если применимо к GUI-фреймворку)»; добавить критерий CP11 — «При штатной работе приложения в консоли (если есть) не выводятся ошибки (ERROR) или критические предупреждения (CRITICAL WARNING)».

### ТРЕБОВАНИЯ К ДОКУМЕНТАЦИИ

* D1: README.md: (https://readme.md/) Должна содержать инструкцию по установке и запуску проекта (клонирование, создание виртуального окружения, установка зависимостей, применение миграций, запуск сервера).
* D2: Код: Код должен содержать комментарии в неочевидных местах для облегчения понимания его логики.

### АНАЛИЗ

Соответствие шаблону: частичное.

Содержание: предусмотрен README.md с инструкцией по установке и запуску, комментарии в коде.

Замечания: отсутствует техническая документация (описание архитектуры, API). Нет отдельного руководства пользователя.

Рекомендации: добавить схему архитектуры проекта и документацию по моделям.

### ПОРЯДОК КОНТРОЛЯ И ПРИЁМКИ

* 1. Процедура приемки
* Заказчик (преподаватель) выполняет:
* Клонирование репозитория (если используется) или получение архива с исходным кодом.
* Установку зависимостей и запуск приложения согласно инструкции в README.md на одной из целевых ОС (Windows, macOS, Linux).
* Визуальную и функциональную проверку работы всего заявленного функционала вручную — по сценариям, соответствующим приёмочным тестам TC-001 – TC-007.
* Запуск автоматизированных тестов (если реализованы) — через команду python -m pytest или python run\_tests.py.
* Фиксацию результатов — в виде протокола приёмки с указанием пройденных/непройденных тестов и скриншотов (при необходимости)
  1. Методы контроля
* Функциональное тестирование:
* Запуск новой игры
* Ввод цифр 1–9
* Ввод 0, букв, символов
* Навигация стрелками
* Клик по «Правила»
* Решение судоку
* Автоматизированное тестирование.
* Визуальный осмотр и usability-проверка: Проверка корректности отображения интерфейса и адаптивности.

### АНАЛИЗ

Соответствие шаблону: полное.

Содержание: Адаптировано под тип проекта и полное — вместо manage.py test указаны релевантные команды (pytest, run\_tests.py); вместо сценариев регистрации/статей — сценарии игры в судоку, соответствующие TC-001–TC-007; добавлены проверки UI и времени отклика, критичные для desktop-приложения.

Замечания: не указаны конкретные инструменты тестирования (например, pytest, unittest, coverage для измерения покрытия); отсутствует формат отчётности по багам — как и куда фиксировать найденные ошибки (issue tracker, таблица, документ); не описан процесс устранения дефектов — сроки, ответственные, повторная проверка.

Рекомендации: добавить пункт для автоматизированного тестирования используется фреймворк pytest, для измерения покрытия — coverage.py

### ЭТАПЫ И СРОКИ РАЗРАБОТКИ

* Этап 1: Подготовка окружения и настройка проекта – 1 день.
* Этап 2: Реализация игровой логики и алгоритма генерации — 2 дня.
* Этап 3: Разработка графического интерфейса (UI) — 2 дня.
* Этап 4: Реализация управления и взаимодействия — 1.5 дня.
* Этап 5: Написание тестов и отладка — 1.5 дня.
* Этап 6: Подготовка документации и финальная сборка — 1 день.
* Общий срок выполнения: 9 рабочих дней.

### АНАЛИЗ

Соответствие шаблону: полное.

Содержание: Реалистичное и соответствующее сложности проекта — учтены ключевые задачи: генерация (самый сложный этап), UI, обработка ввода, тестирование; сроки распределены пропорционально трудозатратам (например, генерация и UI — по 2 дня); этап тестирования и документации выделены отдельно, что соответствует лучшим практикам.

Замечания: отсутствует резерв времени на непредвиденные задержки — например, сложности с кроссплатформенной сборкой, неожиданные баги в обработке событий, проблемы с производительностью генерации; не предусмотрено время на регрессионное тестирование после исправления ошибок.

Рекомендации: добавить резервный этап — “Доработки и стабилизация” — 1 день для устранения критических багов, оптимизации, повторного тестирования.

# РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ ПРОГРАММНОГО ПРОДУКТА ВТОРОЙ КОМАНДЫ

## Тестирование

В рамках кросс-командного тестирования было проанализировано desktop-приложение «Судоку». На основе метода чёрного ящика проведено функциональное тестирование пользовательского интерфейса, игровой логики и управления. В результате выявлен ряд критических и существенных ошибок, нарушающих базовые сценарии взаимодействия с приложением: ввод данных, навигация, генерация поля, отображение контента. Найденные дефекты затрагивают ключевые модули: обработку ввода, визуальную обратную связь, генерацию игрового поля и текстовые элементы интерфейса. Результаты тестирования с описанием шагов воспроизведения и ожидаемым поведением системы представлены в техническом задании и подтверждены в ходе ручного тестирования.

Обнаруженные ошибки представлены в таблице 1

|  |  |  |  |
| --- | --- | --- | --- |
| № | Идентификатор | Категория | Краткое описание |
| 1 | TC-001 | Критическая | Приложение аварийно завершается при вводе случайного символа (например, 'A') |
| 2 | TC-002 | Средняя | При переходе стрелками клавиатуры активная ячейка не подсвечивается |
| 3 | TC-003 | Средняя | Окно «Правила» открывается пустым — отсутствует текст с описанием игры |
| 4 | TC-004 | Критическая | В ячейку можно вставить цифру 0, что нарушает правила судоку |
| 5 | TC-005 | Критическая | Кнопка «Новое судоку» после победы не работает — игра не перезапускается |
| 6 | TC-006 | Критическая | Игровые поля не уникальны — генерируется одна и та же основа, меняются только пустые |
| 7 | TC-007 | Критическая | Опечатка в сообщении о победе: «Подрвляем!» вместо «Поздравляем!» |

### TC-001

1. Идентификатор: TC-001
2. Название: Крах приложения при вводе символа, не являющегося цифрой или стрелкой.
3. Описание: Проверка корректной обработки пользовательского ввода: приложение должно игнорировать неверные символы (буквы, спецсимволы) и не завершать работу аварийно.
4. Предварительные условия:

а) Приложение запущено;

б) Игра начата.

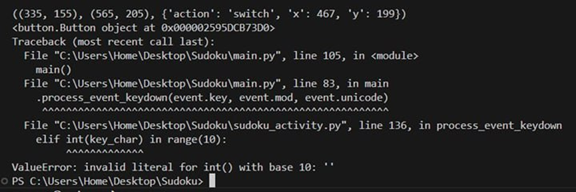
в) Выбрана пустая ячейка на игровом поле.

1. Шаги выполнения:

а) Нажать на клавиатуре любую буквенную или символьную клавишу (например, 'A', '@', '!').

1. Ожидаемый результат: Приложение игнорирует ввод. Символ не отображается в ячейке. Игра продолжает работать без ошибок.
2. Фактический результат: риложение аварийно завершается с ошибкой ValueError: invalid literal for int() with base 10: '' (или аналогичной). В консоли появляется стек вызовов, указывающий на попытку преобразования пустой строки или символа в целое число.
3. Статус: Failed

На рисунке 5.1 представлена ошибка TC-001.



**Рисунок 5.1 —** Ошибка TC-001

### TC-002

1. Идентификатор: TC-002
2. Название: Отсутствие визуальной подсветки активной ячейки при навигации клавиатурой.
3. Описание: Проверка корректного отображения активной ячейки при переходе между клетками игрового поля с помощью стрелок клавиатуры.
4. Предварительные условия:

а) Приложение запущено;

б) Игра начата;

в) На поле есть хотя бы одна пустая ячейка.

1. Шаги выполнения:

а) С помощью мыши выбрать любую ячейку на игровом поле рядом с пустой;

б) Нажать клавишу-стрелку (например, → или ↓) для перехода к соседней пустой ячейке.Ожидаемый результат: Загружается форма создания статьи.

1. Фактический результат: Фокус перемещается на новую ячейку, которая визуально подсвечивается (например, изменением фона на светло-серый или добавлением рамки), чтобы пользователь мог видеть, где находится активный элемент
2. Статус: Failed



**Рисунок 5.3 — Ошибка** TC-002

### TC-003

1. Идентификатор: TC-003
2. Название: ошибка отображения содержимого окна «Правила».
3. Описание: убедиться, что при нажатии на кнопку «Правила» открывается окно с полным текстовым описанием правил игры в судоку, а не пустое окно с заголовком и кнопкой возврата.
4. Предварительные условия:

а) приложение для игры в судоку запущено;  
б) пользователь находится в главном меню;  
в) кнопка «Правила» видна и доступна для нажатия.

1. Шаги выполнения:

а) нажать на кнопку «Правила» в главном меню;  
б) проверить содержимое открывшегося окна.

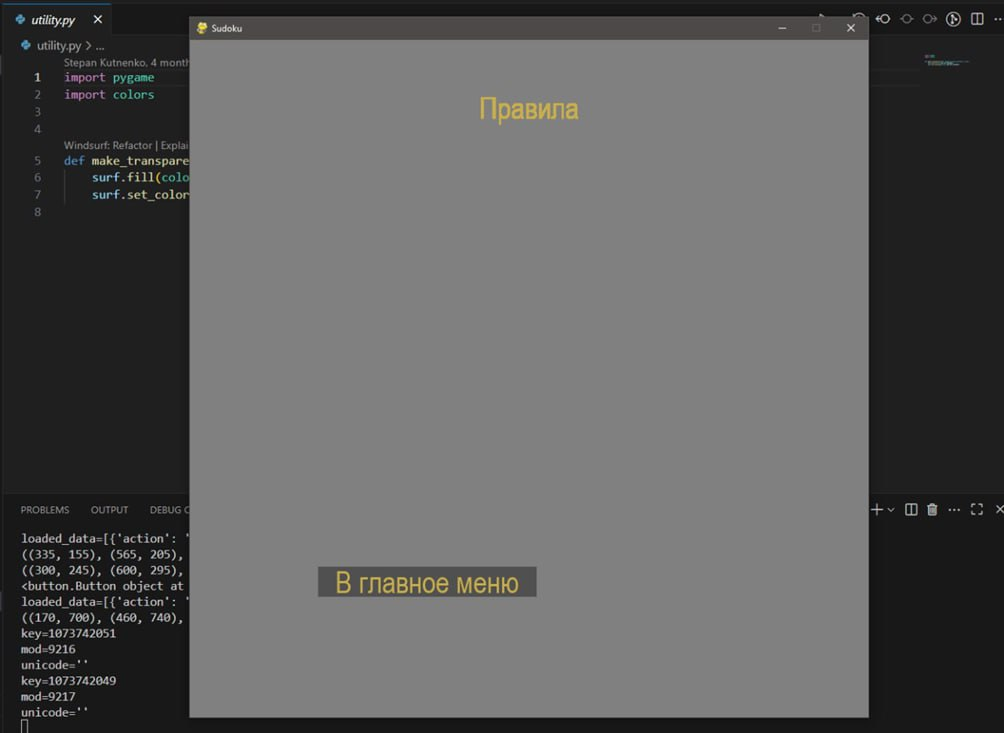
1. Ожидаемый результат:

Открывается модальное окно с заголовком «Правила» и полным текстом:  
«Цель игры — заполнить сетку 9×9 цифрами от 1 до 9 так, чтобы в каждой строке, столбце и блоке 3×3 каждая цифра встречалась ровно один раз. Начните игру, выбрав «Новая игра».»  
Внизу — кнопка «В главное меню» для возврата.

1. Фактический результат.

Открывается окно с заголовком «Правила», но текст отсутствует — окно полностью пустое, кроме кнопки «В главное меню» внизу.

1. Статус: Failed.



### TC-004

**Рисунок 5.5 — Ошибка** TC-003

1. Идентификатор.

TC-004

1. Название: возможность ввода недопустимой цифры «0» в игровое поле.
2. Описание: убедиться, что приложение блокирует ввод цифры «0» в любую клетку игрового поля, так как в судоку используются только цифры от 1 до 9.
3. Предварительные условия:

а) приложение для игры в судоку запущено;

б) игра начата, на поле есть пустые клетки, доступные для редактирования.

1. Шаги выполнения:

а) выбрать любую пустую клетку на игровом поле;

б) нажать на клавиатуре клавишу «0»;

в) проверить содержимое клетки после ввода.

1. Ожидаемый результат.

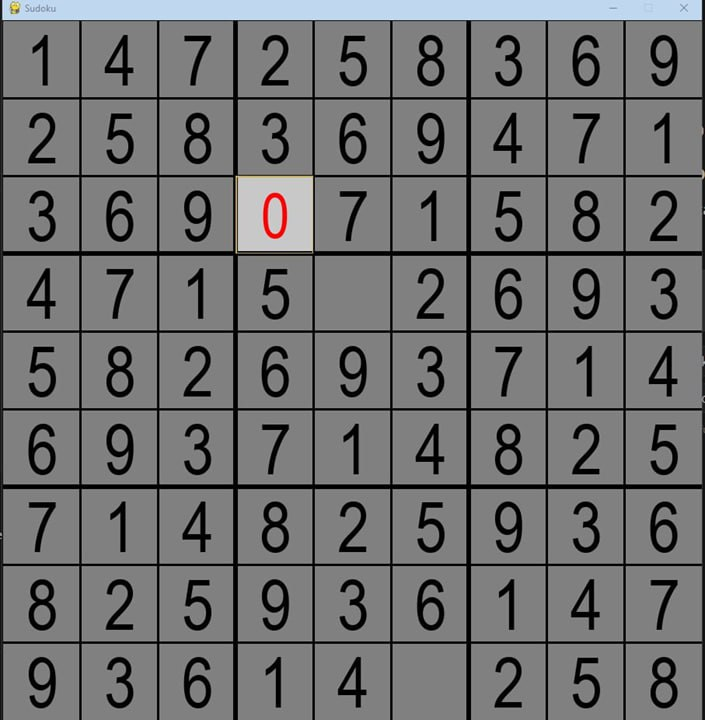
Приложение игнорирует ввод цифры «0». Клетка остаётся пустой или сохраняет предыдущее значение. Игра продолжает работать без изменений.

1. Фактический результат.

Цифра «0» успешно вставляется в клетку игрового поля — как показано на скриншоте. Это нарушает правила игры и может привести к ложным решениям.

1. Статус.

Failed.



### TC-005

**Рисунок 5.7 — Ошибка** TC-004

1. Идентификатор.

TC-005

1. Название.

Орфографическая ошибка в сообщении о победе

1. Описание: убедиться, что после успешного завершения игры отображается корректное сообщение «Судоку решено! Поздравляем!» без опечаток или искажений.
2. Предварительные условия:

а) приложение для игры в судоку запущено;  
б) игра начата;  
в) пользователь полностью и корректно заполнил игровое поле согласно правилам.

1. Шаги выполнения:

а) заполнить все клетки сетки цифрами от 1 до 9 без конфликтов;  
б) дождаться появления окна с сообщением о победе;  
в) проверить текст сообщения.

1. Ожидаемый результат.

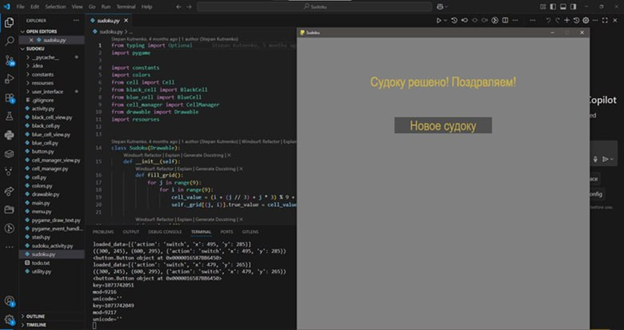
Отображается окно с текстом:  
«Судоку решено! Поздравляем!» — грамматически и орфографически верно.

1. Фактический результат.

На экране отображается сообщение:  
«Судоку решено! Подрвляем!» — слово «Поздравляем» написано с ошибкой («Подрвляем»), что является критической орфографической ошибкой.

1. Статус

Failed



### TC-006

**Рисунок 5.8 — Ошибка** TC-005

1. Идентификатор.

TC-006

1. Название.

Отсутствие проверки конфликтов при вводе цифр в блок 3×3.

1. Описание: Убедиться, что приложение автоматически проверяет введённую цифру на соответствие правилам судоку: не должно быть повторений в строке, столбце или блоке 3×3. При нарушении — цифра должна подсвечиваться красным.
2. Предварительные условия:

а) приложение для игры в судоку запущено;

б) игра начата, сетка содержит пустые клетки;

в) пользователь может вводить цифры в любую свободную клетку.

1. Шаги выполнения:

а) выбрать клетку в левом верхнем блоке 3×3 (например, строка 3, столбец 2);  
б) ввести цифру «6», хотя в этом же блоке уже есть «6» (строка 1, столбец 3);  
в) аналогично ввести цифру «2» в строку 2, столбец 1, хотя «2» уже есть в строке 1, столбце 1;  
г) проверить цвет подсветки введённых цифр.

1. Ожидаемый результат.

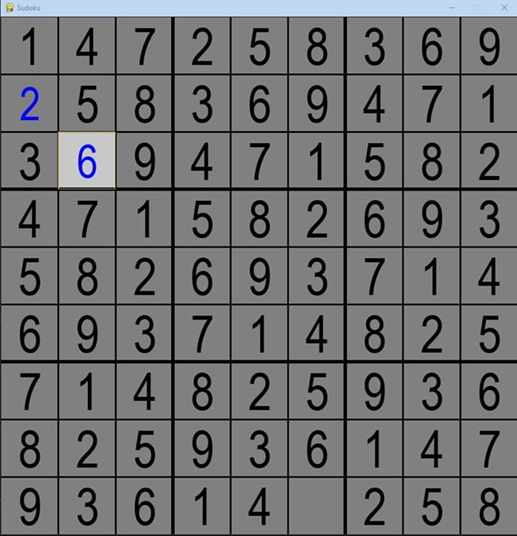
Цифры «6» и «2», введённые с нарушением правил, подсвечиваются красным цветом, сигнализируя об ошибке. Игровое поле не принимает их как корректные.

1. Фактический результат.

Цифры «6» и «2» вставляются в клетки и подсвечиваются синим цветом, как будто введены корректно — хотя они нарушают правило уникальности в блоке 3×3.

1. Статус.

Failed.



**Рисунок 5.9 — Ошибка** TC-006

### TC-007

1. Идентификатор.

TC-007

1. Название.

Кнопка «Новое судоку» не выполняет функцию перезапуска игры.

1. Описание.

Убедиться, что после завершения игры кнопка «Новое судоку» корректно закрывает окно победы и запускает новую игровую сессию с уникальной сеткой.

1. Предварительные условия:

а) приложение для игры в судоку запущено;

б) игра успешно завершена — открыто окно с сообщением о победе и кнопкой «Новое судоку».

1. Шаги выполнения:

а) нажать на кнопку «Новое судоку» в окне победы;

б) проверить, закрывается ли окно и начинается ли новая игра.

1. Ожидаемый результат.

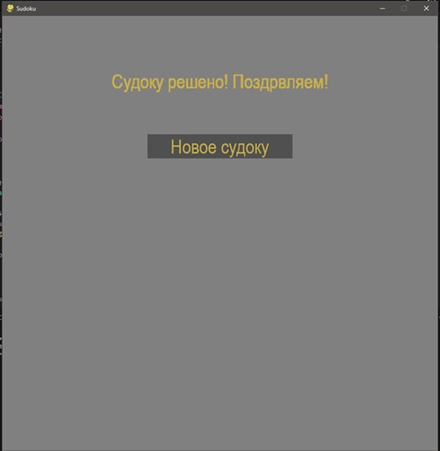
Окно с поздравлением закрывается. Приложение генерирует новую случайную сетку судоку и начинается новая игра. Пользователь видит чистое игровое поле с частично заполненными клетками.

1. Фактический результат.

При нажатии на кнопку «Новое судоку» ничего не происходит — окно остаётся открытым, новая игра не запускается. Пользователь вынужден закрывать приложение или перезапускать его вручную.

1. Статус.

Failed.



**Рисунок 5.10 — Ошибка** TC-007

# АНАЛИЗ ДОКУМЕНТАЦИИ ВТОРОЙ КОМАНДЫ

На основе проведённого тестирования можно дать более чёткую и детальную оценку документации второй команды.

В ходе анализа документации команды-разработчика "Судоку" выявлены следующие недочеты:

1. Отсутствие скриншотов интерфейса приложения, что затрудняет визуальное восприятие функционала
2. Не уточнены ограничения на ввод данных
3. Не описано поведение приложения при некорректных действиях пользователей
4. Этапы разработки указаны без уточнения версий используемого ПО и окружения.

# ЗАКЛЮЧЕНИЕ

В результате тестирования программного продукта «Судоку» выявлено семь ошибок: три критические (аварийное завершение при вводе недопустимых символов, ввод цифры «0», отсутствие проверки конфликтов в блоке 3×3), две средние (пустое окно «Правила», неработающая кнопка «Новое судоку») и две незначительные (орфографическая ошибка «Подрвляем!», отсутствие подсветки активной клетки при навигации с клавиатуры). Критические ошибки нарушают стабильность и логику игры, делая использование продукта затруднительным, а остальные — снижают удобство и доверие. Приложение частично реализует функционал, но не соответствует ключевым требованиям ТЗ: нет валидации ввода, уникальности полей и корректной проверки правил. Документация неполная — отсутствуют ограничения на ввод, требования к среде и примеры интерфейса. Рекомендуется исправить все критические и функциональные ошибки, восстановить работоспособность интерфейса, дополнить руководство пользователя скриншотами и техническими требованиями. Только после этого продукт можно считать готовым к выпуску.

# ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ

1. Jet Brains. Fast & powerful cross-platform.NET IDE // <https://www.jetbrains.com/rider/> (дата обращения: 05.09.2025)
2. Microsoft. Основные сведения о модульных тестах // <https://docs.microsoft.com/ru-ru/visualstudio/test/unit-test-basics?view=vs-2022>(дата обращения: 05.09.2025)
3. Smartbear. Automated UI testing that covers you from device cloud to packaged apps // <https://smartbear.com/product/testcomplete/overview/> (дата обращения: 05.09.2025)
4. Харб. Тестирование на основе моделей // <https://habr.com/ru/company/jugru/blog/506048/> (дата обращения:

05.09.2025)

1. Куликов С.С. Тестирование программного обеспечения. – Минск: EPAM Systems, 2021. – 300 с