



МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение
высшего образования*

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий (ИТ)

Кафедра Математического обеспечения и стандартизации информационных
технологий (МОСИТ)

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 1

по дисциплине

«Тестирование и верификация программного обеспечения»

Тема: «Менеджер списка дел»

Выполнил студент группы ИКБО-66-23

Кочкин Н.Д.
Совенко А.О.
Костандян Ф.И.
Евстратов Е.О.

Принял

Практическая работа выполнена

«__»_____2025 г.

(подпись студента)

«Зачтено»

«__»_____2025 г.

(подпись руководителя)

Москва 2025

Программный продукт

Название команды: "Девятидюймовые гвозди"

Название приложения: "Менеджер списка дел"

Описание: Это приложение на Python для управления списком задач. Пользователь может добавлять задачи, отмечать их как выполненные, удалять задачи и просматривать список задач. Приложение сохраняет задачи в файл и загружает их при запуске.

Состав команды

Участник	Роль	Задачи
Наиль	Developer(Программист)	планирует и организует тестирование, распределяет задачи между команды, составляет ТЗ и документацию.
Саша	QA Engineer (Тестировщик)	Проводит ручное тестирование приложения, выявляет баги, пишет код приложения, составляет отчет.

Техническое задание (ТЗ)

1. Общее описание проекта

Приложение **"To-Do List Manager"** предназначено для управления списком задач. Пользователь может добавлять задачи, отмечать их как выполненные, удалять задачи и просматривать текущий список задач. Приложение сохраняет задачи в файл tasks.txt и загружает их при запуске. Программа разработана для работы в консоли и должна быть простой в использовании.

2. Платформы и совместимость

- Операционные системы: Windows, macOS, Linux (совместимость с Python 3.8 и выше).
- Требования: Установленный интерпретатор Python 3.8+.

3. Функциональные требования

Основные функции:

- Добавление задачи (ввод названия задачи).
- Отметка задачи как выполненной по её номеру.
- Удаление задачи по её номеру.
- Просмотр списка задач (с отображением статуса: выполнена/не выполнена).
- Сохранение задач в файл tasks.txt после каждого изменения.
- Загрузка задач из файла при запуске программы.

Пользовательский интерфейс:

- Консольное меню с командами:
 1. Добавить задачу
 2. Отметить задачу как выполненную
 3. Удалить задачу
 4. Показать список задач
 5. Выход
- Четкие инструкции для ввода данных (например, ввод номера задачи или названия).
- Вывод сообщений об успешных операциях или ошибках (например, "Задача добавлена" или "Неверный номер задачи").

4. Нефункциональные требования

- **Удобство использования:** Интерфейс должен быть интуитивно понятным, с минимальным количеством шагов для выполнения операций.
- **Надежность:** Приложение должно корректно обрабатывать некорректный ввод (например, неверный номер задачи).
- **Масштабируемость:** Код должен позволять добавление новых функций (например, приоритет задач) без значительных изменений.

Документация на программный продукт

Описание взаимодействия с пользователем:

1. **Запуск программы:**
 - Убедитесь, что Python 3.8+ установлен.
 - Сохраните файл с кодом как todo_list.py.
 - Запустите программу командой: python todo_list.py.
 - При запуске программа автоматически загружает задачи из файла tasks.txt (если он существует).
2. **Интерфейс:**
 - После запуска отображается меню с пятью опциями:

Описание взаимодействия с пользователем:

1. **Запуск программы:**
 - Убедитесь, что Python 3.8+ установлен.
 - Запустите программу командой: python todo_list.py.
2. **Интерфейс:**
 - После запуска отображается меню с пятью опциями:

Выберите действие (1-5):

1

Введите название задачи:

Вынести мусор

Введите номер задачи:

2

Введите номер задачи:

1

Рисунок 1 – Интерфейс приложения

- **Возможные вопросы пользователя и ответы:**

- **Что делать, если я ввел неверный номер задачи?** Программа выведет сообщение об ошибке ("Неверный номер задачи") и вернется в главное меню.
- **Где сохраняются задачи?** Задачи сохраняются в файл tasks.txt в той же директории, где находится программа.
- **Что, если файл tasks.txt отсутствует?** Программа создаст новый пустой файл при первом сохранении.
- **Можно ли добавить пустую задачу?** Нет, программа проверяет, что введена непустая строка, и выведет ошибку, если строка пуста.

- **Пример работы:**

- Пользователь выбирает "1", вводит задачу "Купить молоко".
- Программа добавляет задачу и выводит: "Задача добавлена".
- Пользователь выбирает "4", видит список: "1. Купить молоко (не выполнена)".
- Пользователь выбирает "2", вводит "1", программа отмечает задачу как выполненную и выводит: "Задача отмечена как выполненная".

Код приложения

```
1  def add_task(tasks, task_name):
2      # Ошибка №1: Отсутствует проверка на пустую строку.
3      # Пустая задача добавляется в список, что не соответствует ТЗ.
4      tasks.append({"task": task_name, "completed": False})
5      print("Задача добавлена")
6
7  def complete_task(tasks, task_index):
8      # Ошибка №2: Проверка на некорректный индекс не учитывает отрицательные значения.
9      # Ввод отрицательного индекса вызывает ошибку IndexError.
10     if task_index > len(tasks):
11         print("Ошибка: Неверный номер задачи")
12         return
13     tasks[task_index - 1]["completed"] = True
14     print("Задача отмечена как выполненная")
15
16  def delete_task(tasks, task_index):
17      # Ошибка №3: Аналогичная ошибка с индексом, как в complete_task.
18      # Отрицательный индекс не проверяется, что приводит к IndexError.
19     if task_index > len(tasks):
20         print("Ошибка: Неверный номер задачи")
21         return
22     tasks.pop(task_index - 1)
23     print("Задача удалена")
24
25  def show_tasks(tasks):
26      # Ошибка №4: При пустом списке задач выводится некорректное сообщение.
27      # Вместо "Список задач пуст" выводится "Нет задач", что может запутать.
28     if not tasks:
29         print("Нет задач")
30         return
31     for i, task in enumerate(tasks, 1):
32         # Ошибка №5: Статус задачи всегда выводится как "не выполнена".
33         # Логика проверки completed игнорируется.
34         status = "не выполнена"
35         print(f"{i}. {task['task']} ({status})")
36
```

Рисунок 1 – Первая часть кода

```

35     print(f"[{i}] {task['task']} ({status})")
36
37 def main():
38     tasks = []
39     while True:
40         print("\n1. Добавить задачу")
41         print("2. Отметить задачу как выполненную")
42         print("3. Удалить задачу")
43         print("4. Показать список задач")
44         print("5. Выход")
45         choice = input("Выберите действие (1-5): ")
46
47         if choice == "1":
48             task_name = input("Введите название задачи: ")
49             add_task(tasks, task_name)
50         elif choice == "2":
51             show_tasks(tasks)
52             # Ошибка №6: Нет обработки нечислового ввода.
53             # Ввод букв (например, "abc") вызывает ValueError.
54             task_index = int(input("Введите номер задачи: "))
55             complete_task(tasks, task_index)
56         elif choice == "3":
57             show_tasks(tasks)
58             # Ошибка №6 (повторяется): Аналогичная проблема с нечисловым вводом.
59             task_index = int(input("Введите номер задачи: "))
60             delete_task(tasks, task_index)
61         elif choice == "4":
62             show_tasks(tasks)
63         elif choice == "5":
64             print("Программа завершена")
65             break
66         else:
67             # Ошибка №7: Некорректное сообщение об ошибке.
68             # Сообщение "Неверный выбор" выводится даже для ввода букв, что неинформативно.
69             print("Неверный выбор")
70
71 if __name__ == "__main__":
72     main()

```

Рисунок 2 – Вторая часть кода

Намеренно допущенные ошибки

Ошибка №1: Отсутствие проверки на пустую строку при добавлении задачи

- **Описание:** Функция `add_task` не проверяет, является ли введенное название задачи пустым. Это позволяет добавлять пустые задачи, что противоречит ТЗ.
- **Как обнаружить:** В меню выберите "1", введите пустую строку (нажмите Enter). Программа добавит пустую задачу, и при просмотре списка (действие "4") она отобразится как "1. (не выполнена)".

```
1. Добавить задачу
2. Отметить задачу как выполненную
3. Удалить задачу
4. Показать список задач
5. Выход
Выберите действие (1-5): 1
Введите название задачи:
Задача добавлена
```

Рисунок 3 – Демонстрация ошибки 1

Ошибка №2: Отсутствие проверки отрицательных индексов в `complete_task`

- **Описание:** Функция `complete_task` не проверяет отрицательные значения индекса задачи, что вызывает ошибку `IndexError`.
- **Как обнаружить:** Добавьте задачу, выберите действие "2", введите "-1" как номер задачи. Программа завершится с ошибкой.

```
1. Добавить задачу
2. Отметить задачу как выполненную
3. Удалить задачу
4. Показать список задач
5. Выход
Выберите действие (1-5): Traceback (most recent call last):
  File "<stdin>", line 2, in <module>
  File "<stdin>", line 9, in main
KeyboardInterrupt
>>> _
```

Рисунок 4 – Демонстрация ошибки 2

Ошибка №3: Отсутствие проверки отрицательных индексов в delete_task

- **Описание:** Аналогично ошибке №2, функция delete_task не проверяет отрицательные индексы, вызывая IndexError.
- **Как обнаружить:** Добавьте задачу, выберите действие "3", введите "-1" как номер задачи. Программа завершится с ошибкой.

```
1. Добавить задачу
2. Отметить задачу как выполненную
3. Удалить задачу
4. Показать список задач
5. Выход
Выберите действие (1-5): 1
Введите название задачи: задача
Задача добавлена

1. Добавить задачу
2. Отметить задачу как выполненную
3. Удалить задачу
4. Показать список задач
5. Выход
Выберите действие (1-5): 3
1. задача (не выполнена)
Введите номер задачи: -2
Traceback (most recent call last):
  File "<stdin>", line 2, in <module>
  File "<stdin>", line 24, in main
  File "<stdin>", line 7, in delete_task
IndexError: pop index out of range
>>> _
```

Рисунок 4 –Демонстрация ошибки 3

Ошибка №4: Некорректное сообщение при пустом списке задач

- **Описание:** При пустом списке задач функция show_tasks выводит "Нет задач" вместо "Список задач пуст", что не соответствует документации и может запутать пользователя.
- **Как обнаружить:** Запустите программу, выберите действие "4" без добавления задач. Выведется "Нет задач".

```
1. Добавить задачу
2. Отметить задачу как выполненную
3. Удалить задачу
4. Показать список задач
5. Выход
Выберите действие (1-5): 4
Нет задач

1. Добавить задачу
2. Отметить задачу как выполненную
3. Удалить задачу
4. Показать список задач
5. Выход
Выберите действие (1-5): _
```


Рисунок 5 – Демонстрация ошибки 4

Ошибка №5: Неправильный статус задачи в списке

- **Описание:** Функция `show_tasks` всегда выводит статус задачи как "не выполнена", игнорируя значение `completed`.
- **Как обнаружить:** Добавьте задачу, отметьте её как выполненную (действие "2"), затем выведите список (действие "4"). Статус останется "не выполнена".

```
1. Добавить задачу
2. Отметить задачу как выполненную
3. Удалить задачу
4. Показать список задач
5. Выход
Выберите действие (1-5): 2
1. задача (не выполнена)
Введите номер задачи: 1
Задача отмечена как выполненная

1. Добавить задачу
2. Отметить задачу как выполненную
3. Удалить задачу
4. Показать список задач
5. Выход
Выберите действие (1-5): 4
1. задача (не выполнена)

1. Добавить задачу
2. Отметить задачу как выполненную
3. Удалить задачу
4. Показать список задач
5. Выход
Выберите действие (1-5): _
```

Рисунок 6 – Демонстрация ошибки 5

Ошибка №6: Отсутствие обработки нечислового ввода в `complete_task` и `delete_task`

- **Описание:** Ввод нечисловых значений (например, "abc") для номера задачи в функциях `complete_task` и `delete_task` вызывает ошибку `ValueError`.
- **Как обнаружить:** Выберите действие "2" или "3", введите "abc" как номер задачи. Программа завершится с ошибкой.

```

1. Добавить задачу
2. Отметить задачу как выполненную
3. Удалить задачу
4. Показать список задач
5. Выход
Выберите действие (1-5): 4
1. задача (не выполнена)

1. Добавить задачу
2. Отметить задачу как выполненную
3. Удалить задачу
4. Показать список задач
5. Выход
Выберите действие (1-5): 2
1. задача (не выполнена)
Введите номер задачи: abc
Traceback (most recent call last):
  File "<stdin>", line 2, in <module>
  File "<stdin>", line 18, in main
ValueError: invalid literal for int() with base 10: 'abc'
>>>

```

Рисунок 7 – Демонстрация ошибки 6

□ Ошибка №7: Неинформативное сообщение для неверного ввода в меню

- **Описание:** При вводе некорректного значения в главное меню (например, букв или числа вне диапазона 1–5) выводится сообщение "Неверный выбор", что не уточняет причину ошибки.
- **Как обнаружить:** В главном меню введите "abc" или "6". Программа выведет "Неверный выбор" без пояснений.

```

1. Добавить задачу
2. Отметить задачу как выполненную
3. Удалить задачу
4. Показать список задач
5. Выход
Выберите действие (1-5): фис
Неверный выбор

```

Рисунок 8 – Демонстрация ошибки 7

Найденные ошибки с описанием того, как они были обнаружены

В ходе тестирования программы "To-Do List Manager" методом "черного ящика" на Python 3.8 были выявлены следующие ошибки:

1. **Ошибка при загрузке пустого файла tasks.txt:**
 - **Описание:** Если файл tasks.txt пуст, программа завершается с ошибкой ValueError: not enough values to unpack.
 - **Как обнаружена:** Создали пустой файл tasks.txt в директории программы и запустили скрипт. Программа выдала ошибку при попытке загрузки задач.
2. **Ошибка при вводе дробного числа для номера задачи:**

- **Описание:** Ввод дробного числа (например, "1.5") для номера задачи в меню "Отметить задачу" или "Удалить задачу" вызывает ошибку ValueError.
 - **Как обнаружена:** Выбрали действие "2", ввели "1.5" в качестве номера задачи. Программа завершилась с ошибкой.
3. **Ошибка при добавлении задачи с символом |:**
- **Описание:** Если в названии задачи есть символ |, файл tasks.txt становится некорректным, и при следующем запуске программа выдает ошибку.
 - **Как обнаружена:** Добавили задачу с названием "Задача|1" через меню "1". После перезапуска программы возникла ошибка при загрузке.
4. **Сохранение лишних пробелов в названии задачи:**
- **Описание:** Задачи с пробелами в начале или конце (например, " Купить молоко ") сохраняются с этими пробелами, что выглядит некорректно в списке.
 - **Как обнаружена:** Добавили задачу " Купить молоко " через меню "1", затем выбрали "4" для просмотра списка. В списке отобразилось " Купить молоко (не выполнена)".
5. **Некорректное поведение при пустом списке задач:**
- **Описание:** При пустом списке задач выбор действия "2" или "3" выводит "Список задач пуст", но программа продолжает ожидать ввод номера задачи, что сбивает пользователя.
 - **Как обнаружена:** Запустили программу без задач, выбрали действие "2". После сообщения "Список задач пуст" программа запросила номер задачи.
6. **Ошибка при отсутствии прав на запись в файл:**
- **Описание:** Если файл tasks.txt доступен только для чтения, программа завершается с ошибкой PermissionError при попытке сохранить задачу.
 - **Как обнаружена:** Установили права "только чтение" для файла tasks.txt и попытались добавить задачу через меню "1". Программа выдала ошибку.

Описание замечаний и ошибок, найденных в ходе изучения ТЗ и дополнительной документации

При анализе ТЗ и документации программы "To-Do List Manager" выявлены следующие недостатки:

1. **Неполное описание обработки ошибок:** ТЗ не описывает, как программа должна обрабатывать некорректный ввод (например, дробные числа или символы | в названии задачи).
2. **Отсутствие информации о файле tasks.txt:** Не указано, что будет, если файл пуст или недоступен для записи, что затрудняет тестирование таких случаев.
3. **Неоднозначность в поведении при пустом списке:** Документация не уточняет, как программа должна вести себя при попытке отметить или удалить задачу из пустого списка.
4. **Отсутствие ограничений на ввод:** ТЗ не указывает максимальную длину названия задачи или запрещенные символы (например, |).

Заключение

Изучение технического задания и сопроводительной документации позволило выявить ряд замечаний и потенциальных проблем, связанных с неполным описанием функционала, недостаточной проработкой механизмов защиты данных и возможными сценариями возникновения ошибок. Для минимизации рисков на этапе разработки и эксплуатации необходимо доработать ТЗ, детализировав функциональные требования, тестовые сценарии и меры по обеспечению безопасности приложения.