МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ

ФЕДЕРАЦИИ

ФГБОУ ВО «ДонНТУ»

Факультет Интеллектуальных систем и программирования

Кафедра «Программная инженерия» им. Л.П. Фельдмана

Лабораторная работа №5

по курсу: «Профессиональная практика программной инженерии»

по теме: «Разработка ручной документации»

Выполнил:

ст. гр. ПИ-21б

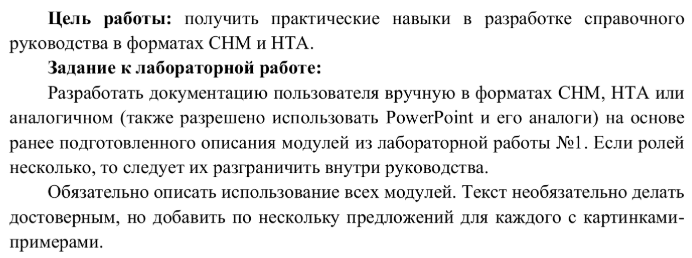
Вавилин М.Е.

Проверил:

асс. каф. ПИ

Ищенко А.П.

Донецк – 2025



**Тема:** планировщик задач

**Репозиторий:** https://github.com/the-real-mhlvvln/pppi

**Документация в формате HTML Application:**

**<!DOCTYPE html>**

**<html lang="ru">**

**<head>**

**<title>Руководство по планировщику задач</title>**

**<HTA:APPLICATION APPLICATIONNAME="Task Planner Documentation" ID="TaskPlannerDoc" VERSION="1.0" BORDER="thin"**

**BORDERSTYLE="normal" CAPTION="yes" SHOWINTASKBAR="yes" SINGLEINSTANCE="yes" SYSMENU="yes" WINDOWSTATE="normal" />**

**<style>**

**body {**

**font-family: Arial, sans-serif;**

**margin: 20px;**

**background-color: #f4f4f9;**

**}**

**h1,**

**h2 {**

**color: #333;**

**}**

**pre {**

**background-color: #f9f9f9;**

**padding: 10px;**

**border: 1px solid #ddd;**

**border-radius: 5px;**

**overflow-x: auto;**

**}**

**.section {**

**margin-bottom: 30px;**

**}**

**.highlight {**

**background-color: #fffbdd;**

**padding: 5px;**

**border-radius: 5px;**

**}**

**</style>**

**</head>**

**<body>**

**<h1 class="highlight">Руководство по планировщику задач</h1>**

**<p>Это руководство описывает функциональность планировщика задач.</p>**

**<!-- Глава 1: Модуль управления задачами -->**

**<div class="section">**

**<h2>Модуль управления задачами (TaskManager)</h2>**

**<p>Этот модуль отвечает за управление задачами, включая их создание, обновление, удаление и отметку как выполненных.**

**</p>**

**<pre><code class="python">**

**class TaskManager:**

**def \_\_init\_\_(self, storage):**

**"""**

**Инициализация TaskManager с использованием DataStorage.**

**:param storage: Экземпляр класса DataStorage для хранения задач.**

**"""**

**self.storage = storage**

**def add\_task(self, title):**

**"""**

**Добавление новой задачи.**

**:param title: Название задачи.**

**:type title: str**

**:return: Сообщение об успешном добавлении задачи.**

**:rtype: str**

**"""**

**tasks = self.storage.load\_tasks()**

**task\_id = len(tasks) + 1**

**new\_task = {"id": task\_id, "title": title, "completed": False}**

**self.storage.add\_task(new\_task)**

**return f"Задача '{title}' добавлена."**

**</code></pre>**

**</div>**

**<!-- Глава 2: Модуль хранения данных -->**

**<div class="section">**

**<h2>Модуль хранения данных (DataStorage)</h2>**

**<p>Этот модуль отвечает за сохранение и загрузку задач из файла JSON.</p>**

**<pre><code class="python">**

**import json**

**import os**

**class DataStorage:**

**def \_\_init\_\_(self, file\_path="tasks.json"):**

**"""**

**Инициализация DataStorage.**

**:param file\_path: Путь к файлу для хранения задач.**

**:type file\_path: str**

**"""**

**self.file\_path = file\_path**

**def load\_tasks(self):**

**"""**

**Загрузка задач из файла.**

**:return: Список задач.**

**:rtype: list**

**"""**

**if not os.path.exists(self.file\_path):**

**return []**

**try:**

**with open(self.file\_path, "r", encoding="utf-8") as file:**

**return json.load(file)**

**except json.JSONDecodeError:**

**print("Ошибка: Файл поврежден или содержит некорректные данные.")**

**return []**

**</code></pre>**

**</div>**

**<!-- Глава 3: Модуль пользовательского интерфейса -->**

**<div class="section">**

**<h2>Модуль пользовательского интерфейса (UI)</h2>**

**<p>Этот модуль предоставляет текстовый интерфейс для взаимодействия с пользователями.</p>**

**<pre><code class="python">**

**from task\_manager import TaskManager**

**from data\_storage import DataStorage**

**def main():**

**"""**

**Точка входа в программу. Предоставляет текстовый интерфейс для управления задачами.**

**"""**

**storage = DataStorage("tasks.json")**

**manager = TaskManager(storage)**

**while True:**

**print("\n=== Планировщик задач ===")**

**print("1. Добавить задачу")**

**print("2. Просмотреть задачи")**

**print("3. Отметить задачу как выполненную")**

**print("4. Удалить задачу")**

**print("5. Обновить задачу")**

**print("6. Выход")**

**choice = input("Выберите действие: ").strip()**

**try:**

**if choice == "1":**

**title = input("Введите название задачи: ").strip()**

**if not title:**

**print("Ошибка: Название задачи не может быть пустым.")**

**continue**

**print(manager.add\_task(title))**

**</code></pre>**

**</div>**

**<!-- Глава 4: API Reference -->**

**<div class="section">**

**<h2>API Reference</h2>**

**<p>Вот полный список доступных методов:</p>**

**<ul>**

**<li><strong>`add\_task(title)`</strong>: Добавляет новую задачу.</li>**

**<li><strong>`list\_tasks()`</strong>: Возвращает список всех задач.</li>**

**<li><strong>`complete\_task(task\_id)`</strong>: Отмечает задачу как выполненную.</li>**

**<li><strong>`delete\_task(task\_id)`</strong>: Удаляет задачу.</li>**

**<li><strong>`update\_task(task\_id, new\_title)`</strong>: Обновляет название задачи.</li>**

**</ul>**

**</div>**

**<!-- Глава 5: Пример использования -->**

**<div class="section">**

**<h2>Пример использования</h2>**

**<p>Вот пример работы с планировщиком задач:</p>**

**<pre><code class="python">**

**if \_\_name\_\_ == "\_\_main\_\_":**

**storage = DataStorage()**

**manager = TaskManager(storage)**

**manager.tasks = storage.load\_tasks()**

**while True:**

**print("\n=== Планировщик задач ===")**

**print("1. Добавить задачу")**

**print("2. Просмотреть задачи")**

**print("3. Отметить задачу как выполненную")**

**print("4. Удалить задачу")**

**print("5. Выход")**

**choice = input("Выберите действие: ")**

**if choice == "1":**

**title = input("Введите название задачи: ")**

**print(manager.add\_task(title))**

**elif choice == "2":**

**print(manager.list\_tasks())**

**elif choice == "3":**

**task\_id = int(input("Введите ID задачи для отметки: "))**

**print(manager.complete\_task(task\_id))**

**elif choice == "4":**

**task\_id = int(input("Введите ID задачи для удаления: "))**

**print(manager.delete\_task(task\_id))**

**elif choice == "5":**

**storage.save\_tasks(manager.tasks)**

**print("До свидания!")**

**break**

**else:**

**print("Неверный выбор. Попробуйте снова.")**

**</code></pre>**

**</div>**

**</body>**

**</html>**

**Скриншоты:**

