
Console task manager

Выпуск 1.0.0

Anton

нояб. 15, 2025

Содержание:

Добро пожаловать в документацию по **Консольному** менеджеру задач!

Модуль taskmanager

Существующие команды для main.py: usage: main.py [-h] {add,list,done,delete,view} . . .

positional arguments:

{add,list,done,delete,view}

Команды

add Добавить новую задачу list Показать список задач done Отметить задачу как выполненную delete Удалить задачу view Просмотреть детали задачи

options:

-h, --help show this help message and exit

Подробное описание команды add: usage: main.py add [-h] [-description DESCRIPTION] [-priority {low,medium,high}] [-due-date DUE_DATE] title

positional arguments:

title Название задачи

options:

-h, --help show this help message and exit

--description, -d DESCRIPTION Описание задачи

-priority, -p {low,medium,high}

Приоритет задачи

--due-date DUE_DATE Дата выполнения (YYYY-MM-DD)

Подробное описание команды list: usage: main.py list [-h] [-status {Ожидание,Выполнено}] [-priority {low,medium,high}] [-hide-completed]

options:

-h, --help show this help message and exit

-status {Ожидание,Выполнено}

Фильтр по статусу

-priority {low,medium,high}

Фильтр по приоритету

--hide-completed Скрыть выполненные задачи

Подробное описание команды done: usage: main.py done [-h] task_id

positional arguments:

task_id ID задачи

options:

-h, --help show this help message and exit

Подробное описание команды delete: usage: main.py delete [-h] task_id

positional arguments:

task_id ID задачи

options:

-h, --help show this help message and exit

Подробное описание команды view: usage: main.py view [-h] task_id

positional arguments:

task_id ID задачи

options:

-h, --help show this help message and exit

1.1 Submodules

1.1.1 taskmanager.models module

Этот файл содержит класс Task, который описывает структуру одной задачи. Здесь определяются, какие данные хранятся в каждой задаче.

```
class taskmanager.models.Task(title, description, priority, due_date=None, id=None, status=None,
                              created_date=None, completed_date=None)
```

Базовые классы: object

Класс служит для описания структуры одной задачи.

id

Уникальный идентификатор задачи

Type

int, optional

title

Название задачи

Type

str

description

Подробное описание задачи

Type

str

`status`

Статус выполнения задачи

Type

`str`

`priority`

Уровень приоритета задачи

Type

`str`

`created_date`

Дата создания в формате ISO

Type

`str`

`due_date`

Дата выполнения в формате ISO

Type

`str, optional`

`completed_date`

Дата завершения в формате ISO

Type

`str, optional`

`__init__(title, description, priority, due_date=None, id=None, status=None, created_date=None, completed_date=None)`

Метод инициализирует необходимые атрибуты.

Параметры

- `title (str)` – Название задачи;
- `description (str)` – Подробное описание задачи;
- `priority (str)` – Уровень приоритета („low“, „medium“, „high“);
- `due_date (str, optional)` – Дата выполнения в формате ISO. По умолчанию `None`;
- `id (int, optional)` – Уникальный идентификатор. По умолчанию `None`;
- `status (str, optional)` – Статус выполнения. По умолчанию „Ожидание“;
- `created_date (str, optional)` – Дата создания. По умолчанию текущая дата;
- `completed_date (str, optional)` – Дата завершения. По умолчанию `None`.

`__str__()`

Метод отображает информацию о задачах.

Результат

Строка с форматированием, показывающая основную информацию о задаче.

Тип результата

`str`

`change_task_execution_status()`

Метод изменяет статус задачи на „Выполнено“ и устанавливает текущую дату и время, записывая в `completed_date`.

`classmethod from_dict(data)`

Метод для создания объекта из словаря.

Параметры

`data (dict)` – Словарь, содержащий данные задачи, полученный из `to_dict()`.

Результат

Новый объект задачи с данными из словаря.

Тип результата

Task

Исключение

`KeyError` – Если в словаре отсутствуют обязательные ключи.

`to_dict()`

Метод для преобразования в словарь для сохранения.

Результат

Словарь, содержащий данные задачи, где ключи - названия полей, значения - соответствующие им данные.

Тип результата

`dict`

1.1.2 taskmanager.storage module

В этом файле размещены функции, отвечающие за действия над задачами.

`class taskmanager.storage.Json(filename)`

Базовые классы: `object`

Класс служит для взаимодействия программы с json-файлом.

Attributes:

`filename (str)`: Путь к json-файлу.

Позволяет:

создавать, записывать, сохранять изменения в этот файл,

читать, выводить список задач из файла,

создавать новый ID, удалять задачу, перезаписывая json-файл.

`__init__(filename)`

Метод инициализирует атрибут класса.

Параметры

`filename (str)` – Путь к json-файлу.

`create_a_file_if_it_does_not_exist()`

Метод создаст json-файл с пустым списком, если он не существует.

`creating_a_new_id(tasks)`

Метод создает следующее значение идентификатора.

Параметры

`tasks (list)` – Список существующих задач.

Результат

Новый уникальный идентификатор для новой задачи ИЛИ 1, если это первая задача.

Тип результата

int

`delete_task(task_id)`

Метод удаляет задачи.

Параметры

`task_id (int)` – Идентификатор задачи.

Результат

True, если задача успешно удалена.

Тип результата

bool

Исключение

`ValueError` – Если задача не найдена по идентификатору.

`getting_all_tasks()`

Метод отображает все задачи.

Результат

Список объектов Task, преобразованных из json-файла.

Тип результата

list

`load_tasks()`

Метод считывает информацию о существующих задачах из json-файла.

Результат

Список словарей с информацией о задачах из json-файла ИЛИ пустой список в случае ошибки.

Тип результата

list

`save_tasks(task)`

Метод записывает или обновляет информацию о текущей задачи в json-файл. Если задача существует, то производит обновление данных об этой задаче.

Параметры

`task (Task)` – Объект задачи для сохранения.

Результат

Сохраненная задача с обновлёнными данными. Новые задачи вернутся с присвоенным ID.

Тип результата

Task

`write_tasks(tasks)`

Запись изменений в json-файл.

Параметры

`tasks (list)` – Список задач для записи их в json-файл.

1.1.3 taskmanager.commands module

В этом файле содержатся обработчики команд для командной строки main.py.

```
class taskmanager.commands.Command(storage: Json = None)
```

Базовые классы: object

Класс служит для управления задачами через командную строку.

storage

Объект для работы с хранилищем задач.

```
__init__(storage: Json = None)
```

Метод инициализирует атрибут класса.

Параметры

storage – Объект хранилища задач. Если не указан, используется Json(«tasks.json»).

```
add_task(title, description, priority, due_date)
```

Метод добавляет новую задачу.

Параметры

- **title** – Название задачи.
- **description** – Описание задачи.
- **priority** – Приоритет задачи (low, medium, high).
- **due_date** – Дата выполнения в формате YYYY-MM-DD.

Результат

Результат операции сохранения.

Тип результата

bool

```
complete_task(task_id)
```

Метод отмечает задачу, как выполненную.

Параметры

task_id – Идентификатор задачи для изменения статуса.

Результат

True если операция выполнена успешно.

Тип результата

bool

Исключение

ValueError – Если задача с указанным ID не найдена.

```
delete_task(task_id)
```

Метод удаляет задачу.

Параметры

task_id – Идентификатор задачи для удаления.

Результат

Результат операции удаления.

Тип результата

bool

`filter_task(status=None, priority=None, due_date=None, filter_flag=False)`

Метод возвращает отфильтрованный список.

Параметры

- `status` – Статус задачи для фильтрации («Ожидание», «Выполнено»).
- `priority` – Приоритет задачи для фильтрации («low», «medium», «high»).
- `due_date` – Дата выполнения для фильтрации.
- `filter_flag` – Если True, скрывает выполненные задачи.

Результат

Отфильтрованный список объектов Task.

Тип результата

list

`get_task(task_id)`

Метод возвращает информацию о задаче по её идентификатору.

Параметры

`task_id` – Идентификатор задачи.

Результат

Если объект задачи найден. None: Если Объект задачи не найден.

Тип результата

Task

`taskmanager.commands.setup_parser()`

Метод создает команды для командной строки.

Результат

Настроенный парсер аргументов.

Тип результата

argparse.ArgumentParser

t

taskmanager, ??

taskmanager.commands, ??

taskmanager.models, ??

taskmanager.storage, ??