


УП Практическое задание № 1.7

Выполнил: Смирнов Степан Михайлович и
Сабельфельд Василий Алексеевич
Группа: 632 группы

Задание. Напишите консольное приложение, некое подобие ежедневника. Пользователь должен иметь возможность добавлять/удалять/редактировать задачи, а также просматривать задачи на сегодня/завтра/неделю. Помимо этого, пользователь может посмотреть список всех задач и список задач, которые ещё предстоит выполнить, а также список задач, которые уже прошли. О задаче достаточно хранить, её название, описание и дату до которой её нужно выполнить. Данные(задачи) должны храниться в json-файле.

```

1 import json
2 from datetime import datetime, timedelta
3
4
5 def load_content():
6     try:
7         with open("notebook.json", 'r') as file:
8             return json.load(file)
9     except:
10        return []
11
12
13 def save_content(content):
14     with open("notebook.json", 'w') as file:
15         json.dump(content, file, indent=2)
16
17
18 def check_dates_for_completion(content):
19     n = 0
20     for i in content:
21         content[n]["completed"] = check_date_completion(i['date'])
22         n += 1
23      save_content(content)
24
25
26 def view_task_long(tasks):
27     task_number = 1
28
29     for i in tasks:
30         print("\nНомер задачи <-", task_number)
31         task_number += 1
32         print("Статус:", "завершено" if i["completed"] else "в будущем")
33         print("Название задачи:", i['name'])
34         print("Описание задачи:", i['description'])
35         print("Дата окончания задачи:", i['date'])
36
37
38 def get_tasks_by_day(content, day):
39     for i in range(len(content)):
40         if content[i]['date'] == str(day):

```

```

41         print(f"{i + 1}. {content[i]['name']} - {content[i]['description']}")
42
43
44 def view_task_short(content):
45     print("Список задач")
46     for i in range(len(content)):
47         print(
48             f"{i + 1}. {content[i]['name']} - {content[i]['description']} (до {content[i]['date']})")
49
50
51 def add_task(content):
52     name_task = input("Введите название задачи -> ")
53     task_description = input("Введите описание задачи -> ")
54     task_date = validate_date(input("Введите дату окончания задачи(YYYY-MM-DD) -> "))
55     task = {
56         "completed": check_date_completion(task_date),
57         "name": name_task,
58         "description": task_description,
59         'date': task_date
60     }
61     content.append(task)
62     save_content(content)
63     print("Задача добавлена")
64
65
66 def delete_task(content):
67     view_task_short(content)
68     delete_task = int(input("Введите номер задачи для удаления -> ")) - 1
69
70     if 0 <= delete_task < len(content):
71         content.pop(delete_task)
72         save_content(content)
73         print("Задача удалена")
74     else:
75         print("Неверный номер задачи")
76
77
78 def editing_task(content):
79     view_task_short(content)

```

```

80
81     task_index = int(input("Введите номер задачи для редактирования -> ")) - 1
82
83     if 0 <= task_index < len(content):
84         print("\nДля сохранения оставьте пустым")
85         task_name = input("Введите название задачи -> ")
86         task_description = input("Введите описание задачи -> ")
87         task_date = input("Введите дату окончания задачи(YYYY-MM-DD) -> ")
88
89         if task_name != "":
90             content[task_index]['name'] = task_name
91         if task_description != "":
92             content[task_index]['description'] = task_description
93         if task_date != "":
94             content[task_index]['date'] = validate_date(task_date)
95             content[task_index]['completed'] = check_date_completion(task_date)
96
97         save_content(content)
98         print("Задача отредактирована")
99     else:
100         print("Неверный номер задачи")
101
102
103     def view_tasks_for_today(content):
104         today = datetime.now().date()
105
106         print("Задачи на сегодня:")
107         get_tasks_by_day(content, today)
108
109
110     def view_tasks_for_tomorrow(content):
111         tomorrow = datetime.now().date() + timedelta(days=1)
112
113         print("Задачи на завтра:")
114         get_tasks_by_day(content, tomorrow)
115

```

```

115
116
117 def view_task_for_week(content):
118     week_start = datetime.now().date()
119     week_end = week_start + timedelta(days=7)
120     print("Задачи на неделю")
121     for i in range(len(content)):
122         data = datetime.strptime(content[i]['date'], _format: "%Y-%m-%d").date()
123         if week_start <= data <= week_end:
124             print(f"{i + 1}. {content[i]['name']} - {content[i]['description']} (до {content[i]['date']})")
125
126
127 def show_tasks_by_completion(content, completion_state):
128     if completion_state:
129         print("Выполненные задачи")
130     else:
131         print("Невыполненные задачи")
132
133     for i in content:
134         if i["completed"] == completion_state:
135             print(f"{i['name']} - {i['description']} <- до {i['date']}")
136
137
138 def validate_date(string):
139     return str(datetime.strptime(string, _format: "%Y-%m-%d").date())
140
141
142 def check_date_completion(date_to_check):
143     date = datetime.strptime(date_to_check, _format: "%Y-%m-%d").date()
144     if date < datetime.now().date():
145         return True
146     return False
147
148
149 check_dates_for_completion(load_content())
150
151 while True:
152     tasks = load_content()
153     print("Действие с файлом:")
154     print("1. Добавить задачу")

```



```
155     print("2. Удалить задачу")
156     print("3. Редактировать задачу")
157     print("4. Просмотр задач")
158     print("5. Просмотр задач на сегодня")
159     print("6. Просмотр задач на завтра")
160     print("7. Просмотр задач на неделю")
161     print("8. Задачи на выполнение")
162     print("9. Задачи выполненные")
163     print("0. Выход")
164
165     task_number = int(input("Введите номер действия -> "))
166
167     if task_number == 1:
168         add_task(tasks)
169     elif task_number == 2:
170         delete_task(tasks)
171     elif task_number == 3:
172         editing_task(tasks)
173     elif task_number == 4:
174         view_task_long(tasks)
175     elif task_number == 5:
176         view_tasks_for_today(tasks)
177     elif task_number == 6:
178         view_tasks_for_tomorrow(tasks)
179     elif task_number == 7:
180         view_task_for_week(tasks)
181     elif task_number == 8:
182         show_tasks_by_completion(tasks, completion_state: False)
183     elif task_number == 9:
184         show_tasks_by_completion(tasks, completion_state: True)
185
186     if task_number == "0":
187         break
188
189     print()
190     input("Нажмите enter для продолжения")
191
```