

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(content):
19     n = 0
20     for i in content:
21         data = datetime.strptime(i['data'], "%Y-%m-%d").date()
22         if data < datetime.now().date():
23             content[n]["completed"] = True
24             n += 1
25     save_content(content)
26
27
28 def view_task_long(tasks):
29     task_number = 1
30     statuses = ["завершено", "в будущем"]
31
32     for i in tasks:
33         print("\nНомер задачи <=", task_number)
34         task_number += 1
35         print("Статус:", statuses[int(i["completed"])])
36         print("Название задачи:", i['name'])
37         print("Описание задачи:", i['description'])
38         print("Дата окончания задачи:", i['data'])
39
40
41 def get_tasks_by_day(content, day):
42     for i in range(len(content)):
43         if content[i]['data'] == str(day):
44             print(f"{i + 1}. {content[i]['name']} - {content[i]['description']}")
45

```

```

81 def editing_task(content):
82     view_task_short(content)
83
84     task_index = int(input("Введите номер задачи для редактирования -> ")) - 1
85
86     if 0 <= task_index < len(content):
87         print("\nДля сохранения оставьте пустым")
88         task_name = input("Введите название задачи -> ")
89         task_description = input("Введите описание задачи -> ")
90         task_date = input("Введите дату окончания задачи(YYYY-MM-DD) -> ")
91
92         if task_name != "":
93             content[task_index]['name'] = task_name
94         if task_description != "":
95             content[task_index]['description'] = task_description
96         if task_date != "":
97             content[task_index]['data'] = validate_date(task_date)
98
99         save_content(content)
100        print("Задача отредактирована")
101    else:
102        print("Неверный номер задачи")
103
104
105 def view_tasks_for_today(content):
106     today = datetime.now().date()
107
108     print("Задачи на сегодня:")
109     get_tasks_by_day(content, today)
110
111
112 def view_tasks_for_tomorrow(content):
113     tomorrow = datetime.now().date() + timedelta(days=1)
114
115     print("Задачи на завтра:")
116     get_tasks_by_day(content, tomorrow)
117
118
119 def view_task_for_week(content):
120     week_start = datetime.now().date()
121     week_end = week_start + timedelta(days=7)
122     print("Задачи на неделю")
123     for i in range(len(content)):
124         data = datetime.strptime(content[i]['data'], "%Y-%m-%d").date()
125         if week_start <= data <= week_end:

```

```

118
119 def view_task_for_week(content):
120     week_start = datetime.now().date()
121     week_end = week_start + timedelta(days=7)
122     print("Задачи на неделю")
123     for i in range(len(content)):
124         data = datetime.strptime(content[i]['data'], "%Y-%m-%d").date()
125         if week_start <= data <= week_end:
126             print(f"{i + 1}. {content[i]['name']} - {content[i]['description']} (до {content[i]['data']})")
127
128
129 def show_tasks_by_completion(content, completion_state):
130     if completion_state:
131         print("Выполненные задачи")
132     else:
133         print("Невыполненные задачи")
134
135     for i in content:
136         if i["completed"] == completion_state:
137             print(f"{i['name']} - {i['description']} <- до {i['data']}")
138
139
140 def validate_date(string):
141     return str(datetime.strptime(string, "%Y-%m-%d").date())
142
143 check_dates(load_content())
144
145 while True:
146     tasks = load_content()
147     print("Действие с файлом:")
148     print("1. Добавить задачу")
149     print("2. Удалить задачу")
150     print("3. Редактировать задачу")
151     print("4. Просмотр задач")
152     print("5. Просмотр задач на сегодня")
153     print("6. Просмотр задач на завтра")
154     print("7. Просмотр задач на неделю")
155     print("8. Задачи на выполнение")
156     print("9. Задачи выполненные")
157     print("0. Выход")
158
159     task_number = int(input("Введите номер действия -> "))
160
161     if task_number == 1:
162         add_task(tasks)
163     elif task_number == 2:
164         delete_task(tasks)

```

```
163     elif task_number == 2:
164         |     delete_task(tasks)
165     elif task_number == 3:
166         |     editing_task(tasks)
167     elif task_number == 4:
168         |     view_task_long(tasks)
169     elif task_number == 5:
170         |     view_tasks_for_today(tasks)
171     elif task_number == 6:
172         |     view_tasks_for_tomorrow(tasks)
173     elif task_number == 7:
174         |     view_task_for_week(tasks)
175     elif task_number == 8:
176         |     show_tasks_by_completion(tasks, False)
177     elif task_number == 9:
178         |     show_tasks_by_completion(tasks, True)
179
180     if task_number == "0":
181         |     break
182
183     print()
184     input("Нажмите enter для продолжения")
```