

# МПОШ. Продуктовый сектор.

## Раздел "ИТ". Практика заключительного этапа.

Задание доступно в электронном виде по ссылке: [dt.miet.ru/ppo\\_it](https://dt.miet.ru/ppo_it)

### Условие

Модест - студент третьего курса Международного Института Эстетических Технологий. Модест с детства играет на терменвоксе. Он небезосновательно считает, что его талант способен украсить любое творческое студенческое объединение. К сожалению, не все однокурсники разделяют это мнение.

Модест знает, что традиционно, в преддверии праздников, студенты в общежитии собираются на репетиции перед концертами студенческой самодеятельности. Стоя перед общежитием, Модест видит, в каких окнах горит свет, а значит полным ходом идет репетиция.

Помогите Модесту определить количество и номера комнат, в которых проходят репетиции, для того, чтобы он смог к ним присоединиться со своим терменвоксом.

### Техническое задание

Необходимо разработать программный продукт, позволяющий демонстрировать, в каких комнатах в общежитии репетируют студенты. Входные данные, получаемые посредством API, представляют собой информацию:

- количество комнат на этаже;
- количество окон в комнатах;
- информация о том, в каких окнах горит свет;
- дата получения информации.

Также API предоставляет возможность проверки, правильно ли определены комнаты.

Необходимо разработать алгоритм, позволяющий найти номера комнат, в которых горит свет.

Пользовательский интерфейс должен предоставлять следующие возможности:

- графическое отображение информации о том, в каких окнах горит свет;
- графическое отображение информации о том, в каких окнах горит свет, по отдельности по каждой дате;
- отображение по запросу пользователя номеров комнат, в которых репетируют студенты;
- отображение того, правильно ли определены искомые комнаты;
- позволять запрашивать данные по заданной пользователем датой с последующим отображением;
- введение пользователем информации о количестве комнат на этаже, количестве окон в комнате, и информации о том, в каких окнах горит свет с последующим графическим отображением и поиском комнат, в которых репетируют студенты.

## Входные данные

Получаемые данные разделены по датам и включают в себя информацию о количестве комнат на этаже, количестве окон в комнатах и информации о том, в каких окнах горит свет.

Данные доступны по адресу: [https://olimp.miet.ru/ppo\\_it\\_final](https://olimp.miet.ru/ppo_it_final). В заголовках запроса необходимо указывать логин для Яндекс.контест одного (любого) из участников команды, формата: *ppo\_00\_00000*. Логин участника можно получить в ЕСР.

Формат указания заголовка: `{X-Auth-Token: <login>}`, например:

```
{X-Auth-Token: ppo_12_000000}
```

## Описание API:

`GET /date` - получение списка дат, по которым есть данные.

Общий формат ответа сервера:

```
{
  "message": [
    "<date_1:string>",
    "<date_2:string>",
    "...",
    "<date_n:stting>"
  ]
}
```

`GET ?day=<day>&month=<month>&year=<year>` - получение данных по определенной дате.

Общий формат ответа сервера:

```
{
  "message": {
    "date": {
      "data": "<timestamp:int>",
      "description": "<description:string>"
    },
    "rooms_count": {
      "data": "<rooms_count:int>",
      "description": "<description:string>"
    },
    "windows_for_room": {
      "data": [
        "<windows_for_room_1:int>",
        "<windows_for_room_2:int>",
        "...",
        "<windows_for_room_n:int>"
      ],
      "description": "<description:string>"
    },
    "windows": {
```

```

    "data": {
      "floor_1": [
        "<light_1:bool>",
        "<light_2:bool>",
        "...",
        "<light_m:bool>"
      ],
      "floor_2": [
        "<light_1:bool>",
        "<light_2:bool>",
        "...",
        "<light_m:bool>"
      ],
      "floor_3": [
        "<light_1:bool>",
        "<light_2:bool>",
        "...",
        "<light_m:bool>"
      ],
      "floor_4": [
        "<light_1:bool>",
        "<light_2:bool>",
        "...",
        "<light_m:bool>"
      ]
    },
    "description": "<description:string>"
  }
}

```

**POST** запрос с полезной нагрузкой (payload) в виде количества комнат и номеров комнат в формате:

```

{
  "data": {

```

```
"count": "<count:int>",
"rooms": [
  "<room_number_1:int>",
  "<room_number_2:int>",
  "...",
  "<room_number_count:int>"
]
},
"date": "<date:str>"
}
```

Пример запроса:

```
{
  "data": {
    "count": 4,
    "rooms": [
      3,
      5,
      9,
      10
    ]
  },
  "date": "23-05-21"
}
```

В случае правильного формата запроса, ответ может принимать два значения, а код ответа 200:

```
{
  "message": "correct answer"
}
```

или

```
{
  "message": "wrong answer"
}
```

В случае неверного формата данных или заголовка POST запроса, код ответа примет значение 400.

При отправке POST запроса необходимо указывать в заголовке в поле `Content-Type` значение `application/json`.

Получаемые данные необходимо хранить в системе управления базами данных (СУБД). Выбор СУБД не регламентируется.

## Рекомендации к выполнению

Программный продукт должен работать либо на двух и более платформах, либо иметь веб-интерфейс.

Рекомендуется разделить программный продукт на модули: back-end и front-end.

Разработку рекомендуется вести с помощью системы контроля версий git.

Рекомендуется использовать unit-тестирование при разработке продукта.

Схематичное изображение примера пользовательского интерфейса для набора данных:

```
{
  "date": {
    "data": 1674594000,
    "description": "Татьянин день"
  },
  "rooms_count": {
    "data": 3,
    "description": "Количество комнат на этаже"
  },
  "windows_for_room": {
```

```
    "data": [
      3,
      2,
      1
    ],
    "description": "Количество окон в каждой из комнат на
этаже слева направо"
  },
  "windows": {
    "data": {
      "floor_1": [
        false,
        true,
        false,
        true,
        false,
        false
      ],
      "floor_2": [
        true,
        false,
        true,
        false,
        false,
        true
      ],
      "floor_3": [
        false,
        false,
        true,
        false,
        true,
        false
      ],
      "floor_4": [
        false,
```

```

        false,
        false,
        true,
        false,
        true
    ]
},
"description": "Окна по этажам, в которых горит свет"
}
}

```

представлено на рисунке ниже:

25 января 2023 года								
10	10	10	11	11	12	Входные данные:	Количество комнат на этаже:	3
7	7	7	8	8	9		Окна на этаже:	3 2 1
4	4	4	5	5	6	Ответ:	Количество комнат:	8
1	1	1	2	2	3		Номера комнат:	1, 2, 4, 6, 7, 8, 11, 12

## Регламент испытаний

1. Запуск программного продукта.
2. Отображение окон по одному из дней.
3. Отображение всех дней, по которым получена информация от API.
4. Отображение окон с отметкой, какие горят, какие нет, по одному дню.
5. Отображение окон с отметкой, какие горят, какие нет, по всем дням, по которым получена информация от API.
6. Отображение количества комнат и их номеров по всем дням, с отметкой о том, корректно ли найдены занятые комнате.



7. Введение входных данных жюри с последующим отображением информации о горящих окнах, количества комнат и их номеров;
8. Ведение даты, по которой ранее не получались данные и отображение информации о горящих окнах, количестве комнат и их номерах;
9. Демонстрация работы СУБД, в том числе, включающая в себя проверку персистентного хранения данных.
10. Участники демонстрируют работу unit-тестов.

## Загрузка решений

По окончании защиты необходимо загрузить все исходные файлы проекта:

- архив с репозиторием/программный код;
- файлы с данными;
- файлы БД (дамп БД);
- снимки экрана

в облачную папку в соответствии с наименованием вашей команды.

Адрес облачной папки:

<https://cloud.predprof.olimpiada.ru/index.php/s/dfPJPW73PJYr2M3>