

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

Задание доступно в электронном виде по ссылке: olimp.miet.ru/ppo_it

Оценка

Программный код (суммарно)

Уровень	Критерии	Максимальный балл 40	Комментарий
1	Код написан без соблюдения стилистики, имена переменных не несут смысловой нагрузки, код в целом трудно читаем	0	
2	Код читаем, разработчики в целом придерживаются одного стиля	20	
3	Разработка велась с помощью git	20	

Алгоритм (суммарно)

Уровень	Критерии	Максимальный балл 110	Комментарий
1	Алгоритм неработоспособен/программный код отсутствует.	0	
2	Карта собирается перебором с проверочным набором	30	По 2 балла за каждую из 15 частей. 15 баллов всего если собирается только тестовый набор.
3	Карта собирается на основании математической	10	5 баллов если собирается только

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**

Заключительный этап

Продуктовый сектор

Практика «ИТ»

Командно-практическое задание

	формулы с проверочным набором		тестовый набор.
4	Верно расположены все базовые станции	70	Балл уменьшается пропорционально отклонению от “эталонной” стоимости: $\frac{\text{эталонная}}{\text{полученная}} * 70$ Если стоимость получена меньше эталонной - 0

Работа с данными (накопительно)

Уровень	Критерии	Максимальный балл 50	Комментарий
1	Данные от API не разобраны	0	
2	Начата работа с данными, но данные неправильно интерпретированы	20	
3	Данные получены, разобраны, хранятся не в СУБД	30	
4	Данные получены, разобраны, хранятся в СУБД	50	

Пользовательский интерфейс (накопительно)

Уровень	Критерии	Максимальный балл 100	Комментарий
---------	----------	-----------------------	-------------

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**

Заключительный этап

Продуктовый сектор

Практика «ИТ»

Командно-практическое задание

1	Интерфейс пользователя отсутствует, в том числе реализован в командной строке с неполным соответствием ТЗ к пользовательскому интерфейсу	0	
2	Интерфейс реализован в командной строке (для инициализации, демонстрации карт и данных на них используются отдельные исполняемые файлы) Присутствуют все возможности просмотра информации о станциях.	20	
3	Разработан графический интерфейс на выбранной платформе. Но нет возможности просмотра области покрытия базовых станций.	40	
4	Разработан графический интерфейс на выбранной платформе. Присутствуют возможности просмотра информации о станциях.	60	
5	Разработан графический интерфейс на выбранной платформе. Присутствуют возможности просмотра информации о станциях. Карта и области покрытия демонстрируются в трехмерном формате.	100	

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

Выполнение технического задания

Уровень	Критерии	Максимальный балл	Комментарий
1	Произведен запуск программного продукта	20	
2	Отображение исходной тестовой карты с расположением модулей	20	
3	Отображение тестовой карты с базовыми станциями.	40	
4	Отображение тестовой карты с отображением области покрытия базовых станций.	40	
5	Отображение количества используемых базовых станций первого и второго типа.	30	
6	Ввод адреса сервиса для получения проверочного набора	20	
7	Отображение проверочной карты с базовыми станциями.	40	
8	Отображение проверочной карты с отображением области покрытия базовых станций.	50	
9	Отображение количества используемых базовых станций первого и второго типа.	30	
10	Предоставлены unit-тесты	10	

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**

Заключительный этап

Продуктовый сектор

Практика «ИТ»

Командно-практическое задание

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

Задание доступно в электронном виде по ссылке: olimp.miet.ru/ppo_it

Решение задачи

```
final-predprof-2025/
├── .env
├── .gitignore
├── .idea/
│   ├── .gitignore
│   ├── final-predprof-2025.iml
│   ├── inspectionProfiles/
│   │   ├── Project_Default.xml
│   │   └── profiles_settings.xml
│   ├── misc.xml
│   ├── modules.xml
│   ├── vcs.xml
│   └── workspace.xml
├── _pycache/
│   ├── main.python-312.pyc
├── backend/
│   ├── __init__.py
│   ├── _pycache/
│   │   ├── __init__.python-311.pyc
│   │   ├── __init__.python-312.pyc
│   │   ├── algorithm.python-311.pyc
│   │   ├── algorithm.python-312.pyc
│   │   ├── tiles.python-311.pyc
│   │   └── tiles.python-312.pyc
│   └── algorithm.py
├── database/
│   ├── __all_models.py
│   ├── _pycache/
│   │   ├── __all_models.python-311.pyc
│   │   ├── __all_models.python-312.pyc
│   │   ├── db_session.python-311.pyc
│   │   └── db_session.python-312.pyc
│   ├── db_session.py
│   └── models/
│       ├── __pycache/
│       │   ├── map.python-311.pyc
│       │   ├── map.python-312.pyc
│       │   ├── modules.python-311.pyc
│       │   ├── modules.python-312.pyc
│       │   ├── stations.python-311.pyc
│       │   ├── stations.python-312.pyc
│       │   ├── stations_types.python-311.pyc
│       │   └── stations_types.python-312.pyc
│       ├── map.py
│       ├── modules.py
│       ├── stations.py
│       └── stations_types.py
├── predprof.db
├── forms/
│   ├── __pycache/
│   │   ├── service_addresses.python-311.pyc
│   │   ├── service_addresses.python-312.pyc
│   │   └── service_addresses.py
├── loaders/
│   ├── __pycache/
│   │   ├── dependency_injection.python-311.pyc
│   │   ├── dependency_injection.python-312.pyc
│   │   ├── graphics.python-311.pyc
│   │   ├── graphics.python-312.pyc
│   │   ├── users.python-311.pyc
│   │   └── dependency_injection.py
│   ├── graphics.py
│   └── users.py
├── services/
│   ├── __pycache/
│   │   ├── database_requests.python-311.pyc
│   │   ├── database_requests.python-312.pyc
│   │   └── database_requests.py
├── src/
│   ├── __pycache/
│   │   ├── api_requests.python-311.pyc
│   │   ├── api_requests.python-312.pyc
│   │   ├── api_requests.py
│   │   └── api_loader.json
├── tiles.py
├── tiles_files/
│   ├── 1.json
│   ├── 10.json
│   ├── 11.json
│   ├── 12.json
│   ├── 13.json
│   ├── 14.json
│   ├── 15.json
│   ├── 16.json
│   ├── 2.json
│   ├── 3.json
│   ├── 4.json
│   ├── 5.json
│   ├── 6.json
│   ├── 7.json
│   ├── 8.json
│   └── 9.json
├── main.py
├── map_v1.json
├── mytest.py
├── out/
│   ├── 0.txt
│   ├── 1.txt
│   ├── 10.txt
│   ├── 11.txt
│   ├── 12.txt
│   ├── 13.txt
│   ├── 14.txt
│   ├── 15.txt
│   ├── 2.txt
│   ├── 3.txt
│   ├── 4.txt
│   ├── 5.txt
│   ├── 6.txt
│   ├── 7.txt
│   ├── 8.txt
│   └── 9.txt
├── requirements.txt
├── static/
│   ├── css/
│   │   ├── base.css
│   │   ├── bootstrap.css
│   │   ├── bootstrap.css
│   │   ├── bootstrap.css
│   │   ├── bootstrap.css
│   │   ├── bootstrap.css
│   │   └── bootstrap.css
│   ├── js/
│   │   ├── bootstrap.js
│   │   ├── bootstrap.js
│   │   ├── bootstrap.js
│   │   ├── bootstrap.js
│   │   ├── bootstrap.js
│   │   └── bootstrap.js
├── templates/
│   ├── base.html
│   ├── coverage.html
│   ├── null.html
│   ├── index.html
│   ├── map.html
│   ├── module.html
│   ├── station.html
│   ├── table.html
├── test.py
└── 23 directories, 110 files
```

Рисунок 1: Структура решения

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

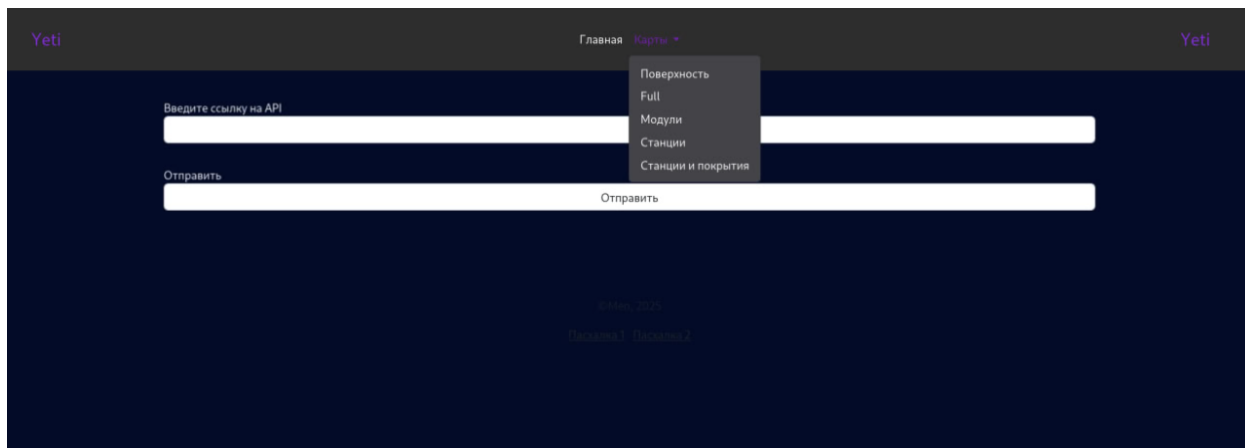


Рисунок 2 : Снимок экрана разработанного программного продукта

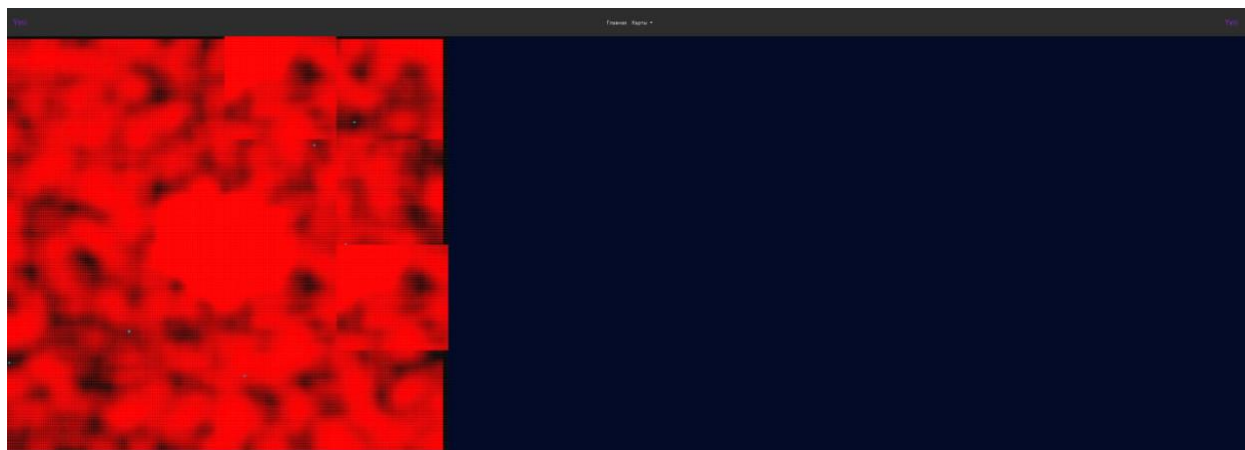


Рисунок 3 : Снимок экрана разработанного программного продукта

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

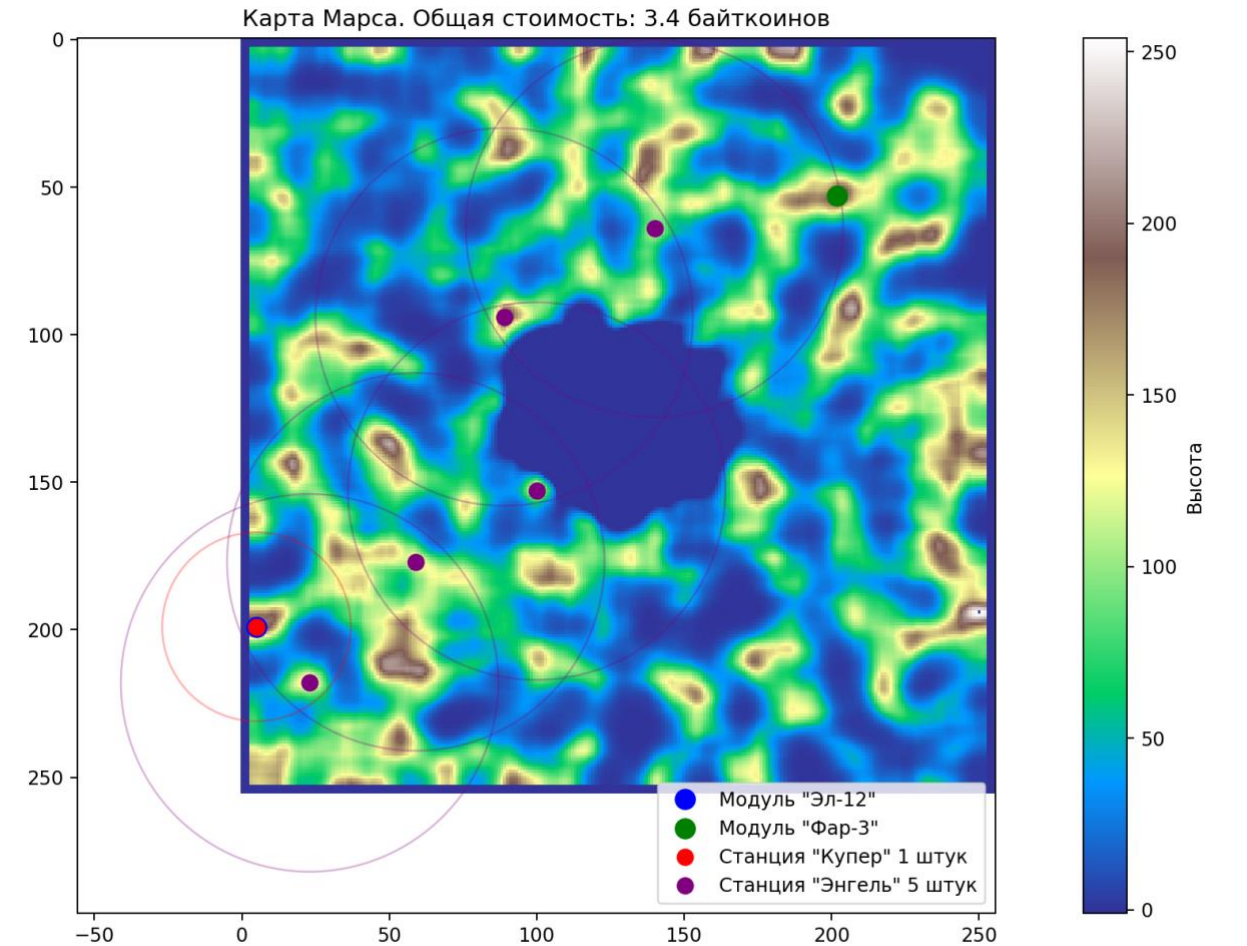


Рисунок 4 : Вариант решения

map	CREATE TABLE map (id INTEGER NOT NULL, x INTEGER, y INTEGER NOT NULL, value INTEGER NOT NULL, map_id INTEGER, PRIMARY KEY (id))
id	"id" INTEGER NOT NULL
x	"x" INTEGER
y	"y" INTEGER NOT NULL
value	"value" INTEGER NOT NULL
map_id	"map_id" INTEGER
modules	CREATE TABLE modules (id INTEGER NOT NULL, x INTEGER NOT NULL, y INTEGER NOT NULL, type VARCHAR NOT NULL, map_id INTEGER, PRIMARY KEY (id))
id	"id" INTEGER NOT NULL
x	"x" INTEGER NOT NULL
y	"y" INTEGER NOT NULL
type	"type" VARCHAR NOT NULL
map_id	"map_id" INTEGER
stations	CREATE TABLE stations (id INTEGER NOT NULL, x INTEGER, y INTEGER NOT NULL, type VARCHAR NOT NULL, cost FLOAT NOT NULL, radius INTEGER NOT NULL, map_id INTEGER, PRIMARY KEY (id))
id	"id" INTEGER NOT NULL
x	"x" INTEGER
y	"y" INTEGER NOT NULL
type	"type" VARCHAR NOT NULL
cost	"cost" FLOAT NOT NULL
radius	"radius" INTEGER NOT NULL
map_id	"map_id" INTEGER
stations_types	CREATE TABLE stations_types (id INTEGER NOT NULL, type VARCHAR NOT NULL, cost FLOAT NOT NULL, radius INTEGER NOT NULL, map_id INTEGER, PRIMARY KEY (id))
id	"id" INTEGER NOT NULL
type	"type" VARCHAR NOT NULL
cost	"cost" FLOAT NOT NULL
radius	"radius" INTEGER NOT NULL
map_id	"map_id" INTEGER

Рисунок 5 : Снимок экрана СУБД 1

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

Таблица: map

	id	x	y	value	map_id
	Фильтр	Фильтр	Фильтр	Фильтр	Фильтр
1	1	0	0	255	0
2	2	0	1	255	0
3	3	0	2	255	0
4	4	0	3	255	0
5	5	0	4	255	0
6	6	0	5	255	0
7	7	0	6	255	0
8	8	0	7	255	0
9	9	0	8	255	0
10	10	0	9	255	0
11	11	0	10	255	0
12	12	0	11	255	0
13	13	0	12	255	0
14	14	0	13	255	0
15	15	0	14	255	0
16	16	0	15	255	0
17	17	0	16	255	0
18	18	0	17	255	0
19	19	0	18	255	0
20	20	0	19	255	0
21	21	0	20	255	0
22	22	0	21	255	0
23	23	0	22	255	0
24	24	0	23	255	0
25	25	0	24	255	0

Рисунок 6 : Снимок экрана СУБД 2

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

Таблица: modules

	id	x	y	type	map_id
	Фи...	Фи...	Фи...	Фильтр	Филь...
1	1	53	202	listener	0
2	2	199	5	sender	0
3	3	53	202	listener	1
4	4	199	5	sender	1

Рисунок 7 : Снимок экрана СУБД 3

Таблица: stations

	id	x	y	type	cost	radius	map_id
	Фи...	Фи...	Фи...	Фил...	Фи...	Фил...	Филь...
1	1	199	5	cuper	0.4	32	0
2	2	176	38	engel	0.6	64	0
3	3	178	111	cuper	0.4	32	0
4	4	90	104	engel	0.6	64	0
5	5	5	168	engel	0.6	64	0
6	6	23	233	engel	0.6	64	0
7	7	53	202	cuper	0.4	32	0
8	8	199	5	engel	0.6	64	1
9	9	180	73	engel	0.6	64	1
10	10	207	139	engel	0.6	64	1
11	11	127	197	engel	0.6	64	1
12	12	67	179	engel	0.6	64	1
13	13	53	202	engel	0.6	64	1

Рисунок 8 : Снимок экрана СУБД 4

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

Таблица: stations_types

	id	type	cost	radius	map_id
	Фи...	Фил...	Фи...	Фил...	Филь...
1	1	cuper	0.4	32	0
2	2	engel	0.6	64	0
3	3	cuper	0.4	32	1
4	4	engel	0.6	64	1

Рисунок 9 : Снимок экрана СУБД 5

Модуль app

Подмодуль main.py

```
import os

import requests

import json

from dotenv import load_dotenv

from flask import Flask, render_template, abort

from backend.database import db_session

from backend.routers import graphics, users

from backend.services.database_requests import get_stations, get_map,
get_custom_map, fill_database

from backend.forms.service_addresses import ServiceAddress
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

```
app = Flask(__name__)

app.config['SECRET_KEY'] = "Otcheburashim"


def count_price():

    d = {}

    stations_list = get_stations()

    prices = {'cuper': 0, 'engel': 0}

    for i in stations_list:

        if i[2] not in d:

            prices[i[2]] = i[3]

            d[i[2]] = 1

        else:

            d[i[2]] += 1

    return (d['cuper'], prices['cuper'], d['engel'], prices['engel'])


@app.route('/', methods=['GET', 'POST'])

def index():

    form = ServiceAddress()

    if form.validate_on_submit():

        url = form.url.data

        try:
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

```
        req = requests.get(url)

    except Exception:

        abort(400)

    with open('backend/src/url_addr.json', 'w') as f:

        json.dump({"url": url}, f)

    with open('map_id.json', 'r') as f:

        map_id = json.load(f)['map_id']

    with open('map_id.json', 'w') as f:

        json.dump({"map_id": map_id + 1}, f)

    fill_database(map_id + 1)

    return render_template("index.html", form=form)

return render_template("index.html", form=form)

@app.route('/map', methods=['GET', 'POST'])
def map():

    with open('map_id.json', 'r') as f:

        map_id = json.load(f)['map_id']

    map = get_map(map_id)

    return render_template("map.html", map=map, len=len(map),
p=count_price())

@app.route('/full', methods=['GET', 'POST'])
def full():
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**

Заключительный этап

Продуктовый сектор

Практика «ИТ»

Командно-практическое задание

```
with open('map_id.json', 'r') as f:

    map_id = json.load(f)['map_id']

    map = get_custom_map(modules=True, stations=True, coverage=True,
map_id=map_id)

    return render_template("full.html", map=map, len=len(map),
p=count_price())

@app.route('/station', methods=['GET', 'POST'])
def station():

    with open('map_id.json', 'r') as f:

        map_id = json.load(f)['map_id']

        map = get_custom_map(modules=False, stations=True, coverage=False,
map_id=map_id)

        return render_template("station.html", map=map, len=len(map),
p=count_price())

@app.route('/coverage', methods=['GET', 'POST'])
def coverage():

    with open('map_id.json', 'r') as f:

        map_id = json.load(f)['map_id']

        map = get_custom_map(modules=False, stations=True, coverage=True,
map_id=map_id)

        return render_template("coverage.html", map=map, len=len(map),
p=count_price())
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

```
@app.route('/module', methods=['GET', 'POST'])

def module():

    with open('map_id.json', 'r') as f:

        map_id = json.load(f)['map_id']

        map = get_custom_map(modules=True, stations=False, coverage=False,
map_id=map_id)

        return render_template("module.html", map=map, len=len(map),
p=count_price())

if __name__ == "__main__":

    db_session.init()

    if not os.path.exists('backend/database/predprof.db'):

        fill_database()

    app.register_blueprint(graphics.blueprint)

    app.register_blueprint(users.blueprint)

    load_dotenv()

    app.run(port=int(os.getenv('SERVER_PORT', 8080)),
host=os.getenv('SERVER_HOST', '127.0.0.1'), threaded=True)
```

static/css

```
.cen {
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

```
display: block;

margin-left: auto;

margin-right: auto;
}

td {

width: 3px;

height: 3px;

/*border: 0.5px solid #ff0000;*/

/*cursor: pointer;*/

background-color: white;

color: #ffffff;

font-size: 2px;
}

@import 'color.css';

body {

background-position: center;

background-repeat: no-repeat; /*повторение фона*/

background-attachment: fixed; /*фон фиксирован*/

padding-top: 70px; /*отступ сверху для header*/

background: rgb(3, 11, 40);

color: var(--text-light);

min-height: 100vh;

display: flex;

flex-direction: column;
}
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

```
@import 'color.css';

.btn-red {

    background: rgb(165, 60, 60);

    color: white;

}

.btn-red:hover {

    background: rgb(165, 60, 60);

    color: white;

}


.btn-green {

    background: rgb(79, 143, 56);

    color: white;

}

.btn-green:hover {

    background: rgb(69, 136, 38);

    color: white;

}


.btn-yellow {

    background: var(--yellow-primary);

    color: #000000;

}

.btn-yellow:hover {
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

```
background: var(--yellow-active);

color: #000000;

}

.btn-orange {

background: var(--orange-primary);

color: #000000;

}

.btn-orange:hover {

background: var(--orange-active);

color: #000000;

}

.btn-purple {

background: var(--purple-primary);

color: #ffffff;

}

.btn-purple:hover {

background: var(--purple-active);

color: #ffffff;

}

.card-body {

background: var(--card-bg);

background: var(--card-bg);

border-radius: 0.7rem;
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

```
}

.card-body-active {

  background: var(--card-bg);

  border: 1px solid var(--purple-active);

  background: var(--card-bg);

  border-radius: 0.7rem;

}

.tablo {

  border: 1px solid #dddddd;

  border-radius: 0.5rem;

  background-color: var(--card-bg);

  min-height: 1vh;


  margin-top: 1rem;

  padding: 15px 15px 15px 15px;

}

:root {

  --station-32: #47ff00;

  --coverage-32: #c9f6b3;

  --station-64: #00daff;

  --coverage-64: #b2efff;

  --module: #d9007c;


  --text-yellow: #fae6b2;

  --yellow-primary: #ffc400;
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

```
--yellow-active: #e59917;

--orange-primary: #ff9123;

--orange-active: #f37516;

--orange-dark: #b77500;

--purple-primary: #8a2be2;

--purple-active: #7a1db6;

--card-bg: rgb(32, 32, 32);

--dark-bg: #1a1a1a;

--input-bg: #2d2d2d;

--text-light: #f5f5f5;

--text-dark: #2d2d2d;

--text-grey: #c4c4c4;

}

@import 'color.css';

.dropdown-item {

    color: white;

}

.dropdown-item {

    color: white;

}

.dropdown-item:hover {

    background-color: rgba(147, 23, 229, 0.3);
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**

Заключительный этап

Продуктовый сектор

Практика «ИТ»

Командно-практическое задание

```
color: #ffffff;

}

.dropdown-item:focus {

    outline: none;

    background-color: #5e5672;

    color: #ffffff;

}

.dropdown-menu {

    background-color: #444348;

}

.my-nav {

    background-color: var(--text-dark);

    position: fixed;

    top: 0;

    left: 0;

    right: 0;

    z-index: 1050;

}

.w-text {

    color: white;

}
```

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ

Заключительный этап

Продуктовый сектор

Практика «ИТ»

Командно-практическое задание

```
y-text {  
  
    color: var(--text-yellow);  
  
}  
  
.dim {  
  
    color: #bbbbbb  
  
}  
  
.bright {  
  
    color: white;  
  
}
```

static/js

```
const  cssFiles  =  ['/static/css/base.css',    '/static/css/body.css',  
                    '/static/css/button.css',    '/static/css/cards.css',  
                    '/static/css/color.css', '/static/css/nav.css', '/static/css/text.css'];  
  
cssFiles.forEach(file => {  
  
    const link = document.createElement('link');  
  
    link.rel = 'stylesheet';  
  
    link.href = file;  
  
    document.head.appendChild(link);  
  
});
```

templates

base.html

```
<!DOCTYPE html>  
  
<html lang="en">
```

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ

Заключительный этап

Продуктовый сектор

Практика «ИТ»

Командно-практическое задание

```
<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link rel="icon" type="image/png" href="/static/media/logo.png">

  <title></title>


  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.
css"

    rel="stylesheet"

    integrity="sha384-
9ndCyUaIbzAi2FUVXJi0CjmCapSm07SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM"

    crossorigin="anonymous"> <!-- Bootstrap CSS -->

  <link                href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.10.5/font/bootstrap-icons.css" rel="stylesheet"> <!-- т иконки от
Bootstrap -->

  <link                                rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-
beta3/css/all.min.css"> <!-- т набор иконок Font Awesome -->


  <script src="static/js/connect_styles.js"></script> <!-- -->

  <script
src="https://cdnjs.cloudflare.com/ajax/libs/highlight.js/11.7.0/highlight.
min.js"></script>

  <script>hljs.highlightAll();</script> <!-- подсвечивает все кодовые
блоки -->

  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle
.min.js"
```

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ

Заключительный этап

Продуктовый сектор

Практика «ИТ»

Командно-практическое задание

```
integrity="sha384-
geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz"

crossorigin="anonymous"></script> <!-- модальные окна, выпадающие
СПИСКИ -->

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script> <!--
графики -->

<style>

    .nav-link {

        color: white;

    }

    .nav-link:hover {

        color: var(--purple-primary);

    }

    .nav-item.active .nav-link {

        color: var(--purple-active);

    }

    .navbar-toggler-icon {

        background-color: white;

    }

</style>

</head>

<body>

<nav class="navbar navbar-expand-md my-nav py-3 px-5">
```

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ

Заключительный этап

Продуктовый сектор

Практика «ИТ»

Командно-практическое задание

```
<span class="bs-icon-sm bs-icon-rounded bs-icon-primary d-flex justify-
content-center align-items-center me-2 bs-icon" >

    <!--  -->

    <a class="navbar-brand d-flex align-items-center brand-logo"
href="/">

        <!--  --
>

        <span class="h4 mb-0" style="color: var(--purple-
primary);">Yeti</span>

    </a>

</span>

<button class="navbar-toggler" data-bs-toggle="collapse" data-bs-
target="#navbarNav" style="background-color: white;">

    <span class="navbar-toggler-icon" style="background-color:
white;"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

    <ul class="navbar-nav mx-auto">

        <li class="nav-item">

            <a class="nav-link" href="/">Главная</a>

        </li>
```

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ

Заключительный этап

Продуктовый сектор

Практика «ИТ»

Командно-практическое задание

```
<li class="nav-item dropdown">

    <a class="nav-link dropdown-toggle" href="#"
id="navbarDropdown2" role="button" data-bs-toggle="dropdown" aria-
expanded="false">

        Карты

    </a>

    <ul class="dropdown-menu" aria-
labelledby="navbarDropdown2">

        <li><a class="dropdown-item"
href="/map">Поверхность</a></li>

        <li><a class="dropdown-item" href="/full">Full</a></li>

        <li><a class="dropdown-item"
href="/module">Модули</a></li>

        <li><a class="dropdown-item"
href="/station">Станции</a></li>

        <li><a class="dropdown-item" href="/coverage">Станции и
покрытия</a></li>

    </ul>

</li>

</ul>

<div class="text-end py-2">

    <span class="bs-icon-sm bs-icon-rounded bs-icon-primary d-flex
justify-content-center align-items-center me-2 bs-icon" >

        <!--  -->

        <a class="navbar-brand d-flex align-items-center brand-logo"
href="/">
```

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

```
        <!-- 
-->

        <span      class="h4      mb-0"      style="color:      var(--purple-
primary);">Yeti</span>

        </a>

    </span>

</div>

</div>
</nav>

<script>

    document.querySelectorAll('.nav-item').forEach(item => {

        item.addEventListener('click', function() {

            // Удаляем класс active у всех nav-item

            document.querySelectorAll('.nav-item').forEach(nav      =>
nav.classList.remove('active'));

            // Добавляем класс active к текущему nav-item

            this.classList.add('active');

        });

    });

</script>
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**

Заключительный этап

Продуктовый сектор

Практика «ИТ»

Командно-практическое задание

```
<br>

<div class="p-2">

{% block main_content %}

{% endblock %}

</div>

<footer class="text-center py-4">

    <div class="container">

        <div class="row row-cols-1">

            <div class="col">

                <p class="text-muted my-2">©Men, 2025</p>

            </div>

            <div class="col">

                <ul class="list-inline my-2">

                    <li class="list-inline-item"><a class="link-dark"
href="https://rutube.ru/video/c6cc4d620b1d4338901770a44b3e82f4/?r=wd">Пасх
алка 1</a></li>

                    <li class="list-inline-item"><a class="link-dark"
href="https://vkvideo.ru/video-122455412_456247462">Пасхалка 2</a></li>

                </ul>

            </div>

        </div>

    </div>

</footer>
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

```
</body>
```

```
</html>
```

coverage.html

```
{% extends "base.html" %}

{% block main_content %}

    <table style="width: 192px">

        {% for i in range(len) %}

            <tr>

                {% for j in range(len) %}

                    {% if map[i][j][0] == 1 %}

                        <td style="background-color: var(--module)"></td>

                    {% elif map[i][j][0] == 2 %}

                        {% if map[i][j][1] == 32 %}

                            <td style="background-color: var(--station-
32)"></td>

                        {% else %}

                            <td style="background-color: var(--station-
64)"></td>

                        {% endif %}

                    {% elif map[i][j][0] == 3 %}

                        {% if map[i][j][1] == 32 %}

                            <td style="background-color: var(--coverage-
32)"></td>

                        {% else %}

                            <td style="background-color: var(--coverage-
64)"></td>
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

```
        {% endif %}

        {% else %}

            <td style="background-color: rgb({{255 - map[i][j][1]}},
0, 0)"> {{ map[i][j][1] }}</td>

            {% endif %}

        {% endfor %}

    </tr>

{% endfor %}

</table>

{% endblock %}
```

full.html

```
{% extends "base.html" %}

{% block main_content %}

<div class="container py-4 py-xl-5">

    <form action="" method="post">

        {{ form.hidden_tag() }}

        <div class="form-group mb-2">

            {{ form.url.label }}<br>

            {{ form.url(class="form-control") }}<br>

            {% for error in form.url.errors %}

                <div class="alert alert-danger" role="alert">
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

```
        {{ error }}

    </div>

    {% endfor %}

</div>

<div class="form-group mb-2">

    {{ form.submit.label }}<br>

    {{ form.submit(class="form-control") }}<br>

    {% for error in form.submit.errors %}

        <div class="alert alert-danger" role="alert">

            {{ error }}

        </div>

    {% endfor %}

</div>

</form>

</div>

{% endblock %}
```

index.html

```
{% extends "base.html" %}

{% block main_content %}

<div class="container py-4 py-xl-5">

    <form action="" method="post">

        {{ form.hidden_tag() }}

    </form>

</div>

{% endblock %}
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

```
<div class="form-group mb-2">

    {{ form.url.label }}<br>

    {{ form.url(class="form-control") }}<br>

    {% for error in form.url.errors %}

        <div class="alert alert-danger" role="alert">

            {{ error }}

        </div>

    {% endfor %}

</div>


<div class="form-group mb-2">

    {{ form.submit.label }}<br>

    {{ form.submit(class="form-control") }}<br>

    {% for error in form.submit.errors %}

        <div class="alert alert-danger" role="alert">

            {{ error }}

        </div>

    {% endfor %}

</div>


</form>

</div>

{% endblock %}
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

Модуль backend

Подмодуль tiles.py

```
import json

from backend.src.api_requests import get_all_tiles

cnt2 = 4

cnt = 16

sz = 64

A = 256

def get_mat(tile):

    mat = [[0] * cnt2 for i in range(cnt2)]

    for i in range(cnt):

        top = tile[i][0][sz // 2] == 255

        bottom = tile[i][sz - 1][sz // 2] == 255

        left = tile[i][sz // 2][0] == 255

        right = tile[i][sz // 2][sz - 1] == 255

        if top and left:

            mat[0][0] = i

        elif top and right:

            mat[0][cnt2 - 1] = i

        elif bottom and left:

            mat[cnt2 - 1][0] = i

        elif bottom and right:
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

```
        mat[cnt2 - 1][cnt2 - 1] = i

    elif not top and not bottom and not right and not left:

        if tile[i][0][0] == 0:

            mat[2][2] = i

        elif tile[i][0][sz - 1] == 0:

            mat[2][1] = i

        elif tile[i][sz - 1][0] == 0:

            mat[1][2] = i

        else:

            mat[1][1] = i

for i in range(cnt):

    top = tile[i][0][sz // 2] == 255

    bottom = tile[i][sz - 1][sz // 2] == 255

    left = tile[i][sz // 2][0] == 255

    right = tile[i][sz // 2][sz - 1] == 255

    if top and left:

        pass

    elif top and right:

        pass

    elif bottom and left:

        pass

    elif bottom and right:

        pass

    elif not top and not bottom and not right and not left:

        pass
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

```
else:

    if top:

        diff1 = 0

        diff2 = 0

        for k in range(sz):

            diff1 += tile[mat[0][0]][k][sz - 1] - tile[i][k][0]

            diff2 += tile[mat[0][3]][k][0] - tile[i][k][sz - 1]

            if diff1 < diff2:

                mat[0][1] = i

            else:

                mat[0][2] = i

    elif bottom:

        diff1 = 0

        diff2 = 0

        for k in range(sz):

            diff1 += tile[mat[3][0]][k][sz - 1] - tile[i][k][0]

            diff2 += tile[mat[3][3]][k][0] - tile[i][k][sz - 1]

            if diff1 < diff2:

                mat[3][1] = i

            else:

                mat[3][2] = i

    elif right:

        diff1 = 0

        diff2 = 0

        for k in range(sz):
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

```
        diff1 += tile[mat[0][0]][sz - 1][k] - tile[i][0][k]

        diff2 += tile[mat[3][0]][0][k] - tile[i][sz - 1][k]

    if diff1 < diff2:

        mat[1][3] = i

    else:

        mat[2][3] = i

elif left:

    diff1 = 0

    diff2 = 0

    for k in range(sz):

        diff1 += tile[mat[0][3]][sz - 1][k] - tile[i][0][k]

        diff2 += tile[mat[3][3]][0][k] - tile[i][sz - 1][k]

    if diff1 < diff2:

        mat[1][0] = i

    else:

        mat[2][0] = i

return mat

def get_field():

    print("Составление матрицы...")

    tiles = get_all_tiles()

    field = [[0] * A for i in range(A)]

    mat = get_mat(tiles)

    for ti in range(cnt2):
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

```
        for tj in range(cnt2):

            for i in range(sz):

                for j in range(sz):

                    field[ti * sz + i][tj * sz + j] =
tiles[mat[ti][tj]][i][j]

    print("Матрица собрана.")

    return field

# def get_tiles_from_files():
#
#     data = []
#
#     for i in range(16):
#
#         with open(f"tyles_files/{i + 1}.json", "r") as f:
#
#             relief = json.load(f)
#
#             data.append(relief)
#
#     return data
```

Подмодуль algorithm.py

```
import heapq

from random import randint

size = 256

cnt = 50

def calc_pos(field, from_i, from_j, to_i, to_j, r1, c1, r2, c2):
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

```
print("Расстановка вышек...")

g = [[[[[] for _ in range(2)] for j in range(size)] for i in range(size)]

for i in range(size):
    for j in range(size):
        for k in range(cnt):
            ei = randint(i - 120, i + 120)
            ej = randint(j - 120, j + 120)
            if ei < 0 or ei >= size or ej < 0 or ej >= size:
                continue
            dist = ((ei - i) ** 2 + (ej - j) ** 2 + (field[i][j] -
field[ei][ej]) ** 2) ** 0.5
            if dist < r1 + r1:
                g[i][j][0].append((ei, ej, 0, c1))
            if dist < r1 + r2:
                g[i][j][0].append((ei, ej, 1, c2))
            if dist < r2 + r1:
                g[i][j][1].append((ei, ej, 0, c1))
            if dist < r2 + r2:
                g[i][j][1].append((ei, ej, 1, c2))

heap = []

dist = [[[1e18 for _ in range(2)] for j in range(size)] for i in
range(size)]

prev = [[(-1, -1, -1) for _ in range(2)] for j in range(size)] for i in
range(size)]

dist[from_i][from_j][0] = c1
dist[from_i][from_j][1] = c2
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

```
heapq.heappush(heap, (c1, from_i, from_j, 0))

heapq.heappush(heap, (c2, from_i, from_j, 1))

while len(heap) > 0:

    i = heap[0][1]

    j = heap[0][2]

    k = heap[0][3]

    heapq.heappop(heap)

    for (child_i, child_j, child_k, w) in g[i][j][k]:

        if dist[child_i][child_j][child_k] > dist[i][j][k] + w:

            dist[child_i][child_j][child_k] = dist[i][j][k] + w

            prev[child_i][child_j][child_k] = (i, j, k)

            heapq.heappush(heap, (dist[child_i][child_j][child_k],
child_i, child_j, child_k))

ans = 1e18

respath = None

for k in range(2):

    cand = dist[to_i][to_j][k]

    path = []

    cur_i = to_i

    cur_j = to_j

    cur_k = k

    while cur_i != -1:

        path.append((cur_i, cur_j, cur_k))

        (cur_i, cur_j, cur_k) = prev[cur_i][cur_j][cur_k]

    if cand < ans:
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Заключительный этап
Продуктовый сектор
Практика «ИТ»
Командно-практическое задание

```
ans = cand

respath = path

respath.reverse()

print("Вышки расставлены.", respath)

return respath
```

Подмодуль src/api_requests.py

```
import requests

import json

def get_url():

    with open("./backend/src/url_addr.json", "rt", encoding="utf-8") as f:

        return json.load(f)['url']

def get_all_tiles():

    print('Получение тайлов...')

    url = get_url()

    tyles = list()

    while len(tyles) != 16:

        resp = requests.get(url)

        if resp.status_code == 200:

            relief = resp.json()['message']['data']

            if relief not in tyles:
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**

Заключительный этап

Продуктовый сектор

Практика «ИТ»

Командно-практическое задание

```
        tyles.append(relief)

    print("Тайлы получены")

    return tyles

def get_coords() -> dict:

    url = get_url()

    while True:

        resp = requests.get(url + "/coords")

        if resp.status_code == 200:

            data = resp.json()['message']

            res = dict()

            res['listener'] = (int(data['listener'][0]),
int(data['listener'][1]))

            res['sender'] = (int(data['sender'][0]),
int(data['sender'][1]))

            res['price'] = {'cuper': float(data['price'][0]), 'engel':
float(data['price'][1])}

            return res

    print(resp.status_code)
```