**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №3

«Практика работы с API»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-33Б |  | преподаватель каф. ИУ5 |
| Нагапетян Валерий |  | Гапанюк Ю. Е. |
|  |  |  |

Москва, 2023 г.

**Задание**

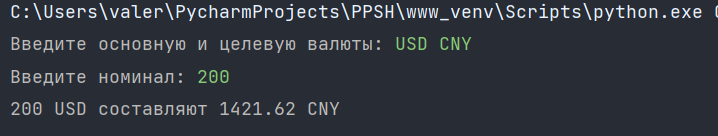
1. Написать программу, которая будет запрашивать у пользователя сумму в одной валюте и конвертировать ее в другую валюту. Программа должна выполнить конвертацию с помощью сетевого запроса к [API конвертера валют](https://freecurrencyapi.com/).
2. Написать программу, которая при помощи [API](https://ip-api.com/) вычисляет геопозицию заданного IP-адреса и отображает ее на карте при помощи библиотеки folium.
3. Используя программные навыки и [NASA API](https://api.nasa.gov/), создать специальную программу для скачивания изображений Земли и сохранения их в формате GIF.

**Задание №1**

**Текст программы**

*import* requests  
*import* config  
  
api\_key = config.API\_KEY\_Task1  
  
base\_currency, target\_currency = map(  
 str, input("Введите основную и целевую валюты: ").split()  
)  
amount = int(input("Введите номинал: "))  
url = f"https://api.freecurrencyapi.com/v1/latest?apikey={api\_key}&base\_currency={base\_currency}&currencies={target\_currency}"  
  
response = requests.get(url)  
  
info = response.json()  
  
*if* info["data"] *and* info["data"][target\_currency]:  
 converted = amount \* info["data"][target\_currency]  
 print(f"{amount} {base\_currency} составляют {converted:.2f} {target\_currency}")  
*else*:  
 print("Указанная валюта не найдена")

**Результат**

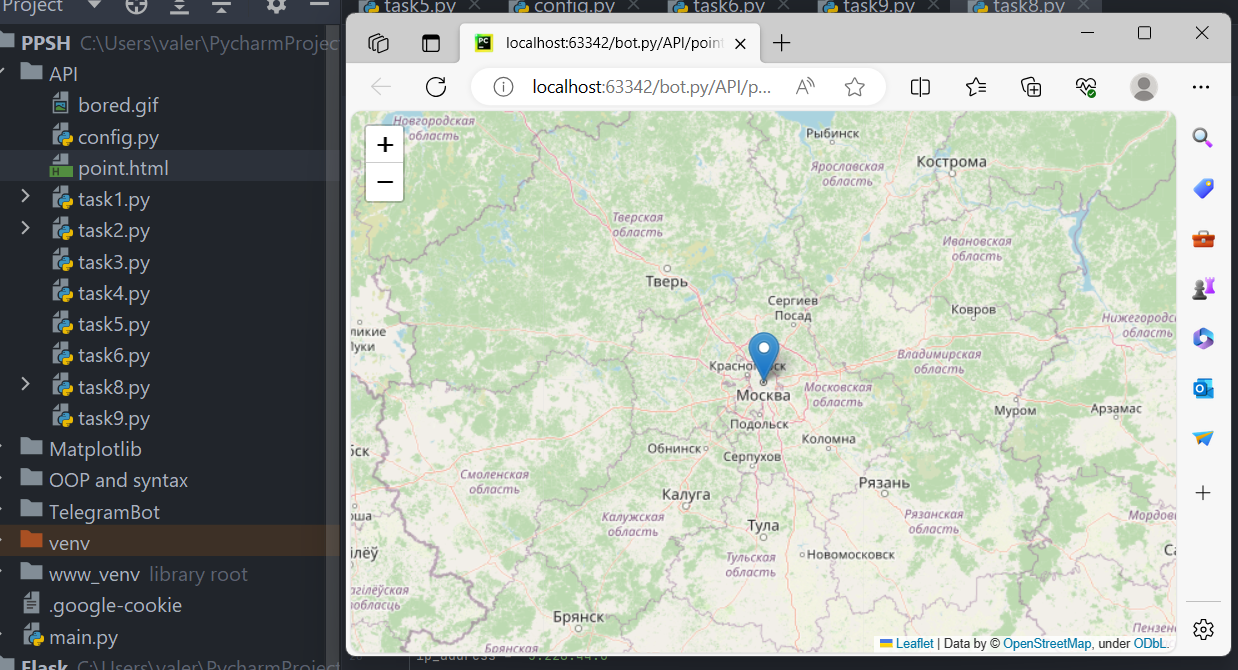
****

**Задание №2**

**Текст программы**

*import* requests  
*import* folium  
  
  
*def* create\_map\_with\_marker(ip):  
 url = f"http://ip-api.com/json/{ip}"  
 response = requests.get(url)  
  
 *if* response.status\_code != 200:  
 print("При извлечении информации об IP-адресе произошла ошибка.")  
 *return* info = response.json()  
 latitude = info.get("lat")  
 longitude = info.get("lon")  
  
 *if not* latitude *or not* longitude:  
 print("Неверный IP-адрес или информация о местоположении не найдена.")  
 *return* map\_point = folium.Map(location=[latitude, longitude], zoom\_start=6)  
 folium.Marker(location=[latitude, longitude], popup="My Point").add\_to(map\_point)  
 map\_point.save("point.html")  
  
  
ip\_address = "5.228.44.0"  
create\_map\_with\_marker(ip\_address)

**Результат**

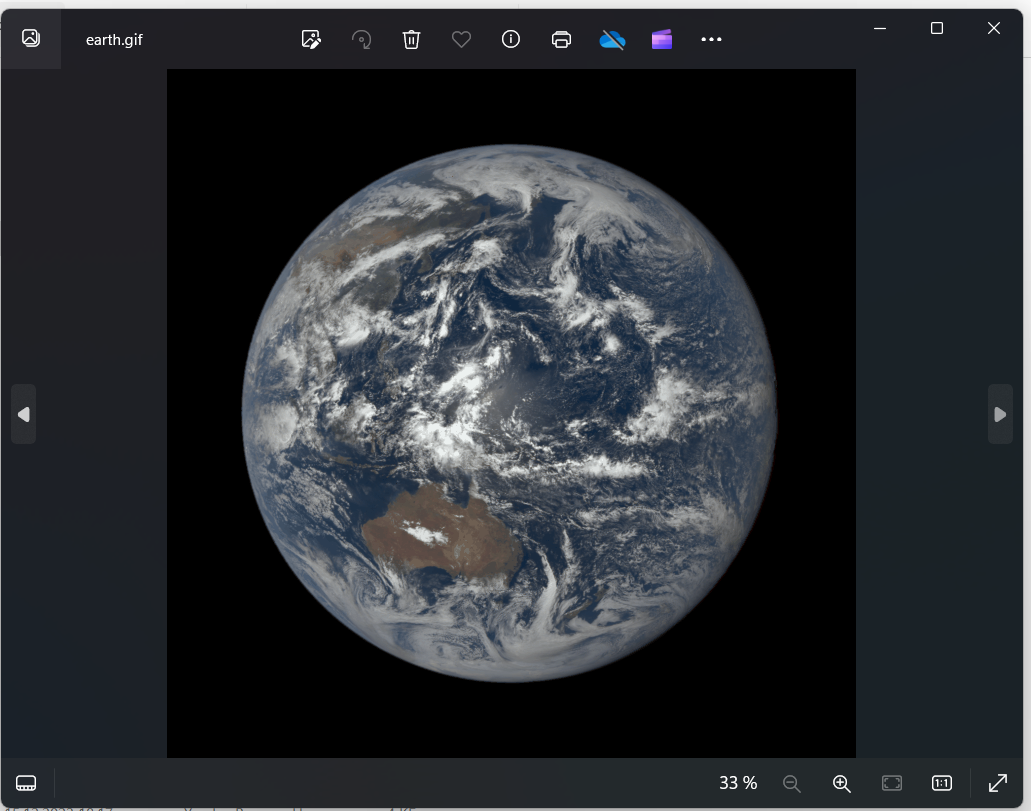
****

**Задание №3**

**Текст программы**

*import* requests  
*from* nasapy *import* Nasa  
*from* PIL *import* Image  
*import* random  
  
api\_key = "lIquZdBhc50ci12gfKiEHjW2KTR4Iw2ZJcevZsoC"  
nasa = Nasa(key=api\_key)  
  
today = "2023-09-09"  
fmt\_today = today.replace("-", "/")  
nasa\_date = nasa.epic(date=today)  
  
images\_counter = 0  
num\_data = len(nasa\_date)  
*for* index, data *in* enumerate(nasa\_date):  
 *try*:  
 response = requests.get(  
 f'https://api.nasa.gov/EPIC/archive/natural/{fmt\_today}/png/{data["image"]}.png?api\_key={api\_key}'  
 )  
 *with* open(f"image{str(index)}.png", "wb") *as* file:  
 file.write(response.content)  
 images\_counter += 1  
 print(f"proccessed: {images\_counter} out of {num\_data}")  
 *except* Exception *as* exc:  
 print(f"Ошибка при скачивании или сохранении изображения {index + 1}: ", exc)  
  
photos = []  
*for* i *in* range(images\_counter):  
 photo = Image.open(f"image{i}.png")  
 photos.append(photo)  
  
photos[0].save(  
 "earth.gif",  
 save\_all=*True*,  
 append\_images=photos[1:],  
 optimize=*True*,  
 duration=[random.randint(80, 100) *for* i *in* range(len(photos))],  
 loop=0,  
)

**Результат**

****