```python
import paramiko
import requests
import json

from interface import Interface

requests.packages.urllib3.disable_warnings()

class Router:
    def __init__(self, host, username, password):
        self.host = host
        self.username = username
        self.password = password
        self.interfaces = self._set_interfaces()

    def get_host(self):
        return self.host

    def _set_interfaces(self):
        interface_names = self.get_interface_names()

        interfaces = []
        for interface_name in interface_names:
            txload = self.get_interface_txload(interface_name)
            interfaces.append(Interface(interface_name, txload))

        return interfaces
```

Зм.	Аркуш	№ дакум.	Подпіс	Дата	БДАС.460641.011 ТЗ			
Распрац.	Буцько					Літ.	Аркуш	Аркушау
Правяр.	Рудинская				Зыходны код		1	4
Рэцэнз.					класа Router			
Н. Кантр.	Боженков					БДАС СП–741		
Зацвярдж.	Горбадей							

```python
def get_interfaces(self):
    return self.interfaces


def get_interface_names(self):
    url = 'https://{h}/restconf/data/ietf-interfaces:
        interfaces/interface?fields=name'.format(h=self.host)

    headers = {'Content-Type': 'application/yang-data+json',
        'Accept': 'application/yang-data+json'}

    response = requests.get(url=url,
                    auth=(self.username, self.password),
                    headers=headers,
                    verify=False).json()

    return [ interface['name'] for interface in response['ietf-interfaces:interface'] ]


def get_interface_txload(self, interface_name):
    ssh = paramiko.SSHClient()
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    ssh.connect(self.host, 22, self.username, self.password, look_for_keys=False)

    stdin, stdout, stderr = ssh.exec_command(
        'show interfaces {interface_name} | include
load'.format(interface_name=interface_name))
    output = stdout.readlines()

    ssh.close()

    return int('\r\n'.join(output).strip().split(', ')[1].split()[1].split('/')[0])
```

```python
def restart_eigrp(self):
    self._disable_eigrp()
    self._enable_eigrp()


def _enable_eigrp(self):
    url = "https://{host}/restconf/data/Cisco-IOS-XE-native:
        native/router/router-eigrp/eigrp/classic-mode=1?fields=shutdown"
        .format(host=self.host)

    headers = {'Content-Type': 'application/yang-data+json'}

    patch_data = '''
        {
          "Cisco-IOS-XE-eigrp:classic-mode": {
            "shutdown": false
          }
        }
    '''

    requests.patch(url=url,
             data=patch_data,
             auth=(self.username, self.password),
             headers=headers,
             verify=False)
```

Зм.	Аркуш	№ докум.	Подпіс	Дата			
					БДАС.460641.011 ТЗ	Аркуш	
							3

```python
def _disable_eigrp(self):
    url = "https://{host}/restconf/data/Cisco-IOS-XE-native:
        native/router/router-eigrp/eigrp/classic-mode=1?fields=shutdown"
        .format(host=self.host)


    headers = {'Content-Type': 'application/yang-data+json'}


    patch_data = '''
        {
          "Cisco-IOS-XE-eigrp:classic-mode": {
            "shutdown": true
          }
        }
        '''


    requests.patch(url=url,
            data=patch_data,
            auth=(self.username, self.password),
            headers=headers,
            verify=False)
```

Зм.	Аркуш	№ докум.	Подпіс	Дата	БДАС.460641.011 Т3	Аркуш
						4