

## ДАДАТАК А

### Зыходны код класа Router

```
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
```

					БДАС.450102.002 Д1			
Зм.	Аркуш	№ дакум.	Подпіс	Дата	Праграмае мадэліраванне размеркавання патокаў трафіка у сетках SDN			
Распрац.	Буцько							
Правяр.	Рудинская							
Рэцэнз.								
Н. Кантр.	Боженков							
Зацвярдж.	Горбадзей				БДАС СП-741			
					Літ.	Аркуш	Аркушай	
						1	4	

## Працяг дадатку А

```
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])
```

					БДАС.450102.002 Д1	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		2

## Працяг дадатку А

```
def restart_egrp(self):
    self._disable_egrp()
    self._enable_egrp()

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

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

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

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

					БДАС.450102.002 Д1	Аркуш
						3
Зм.	Аркуш	№ дакум.	Подпіс	Дата		

## Працяг дадатку А

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

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

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

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

					БДАС.450102.002 Д1	Аркуш
Зм.	Аркуш	№ дакум.	Подпіс	Дата		4