НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський політехнічний інститут імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра програмного забезпечення комп’ютерних систем

**КУРСОВА РОБОТА**

***з дисципліни "Компоненти програмної інженерії"***

Виконав: Щербатюк Ярослав Сергійович

Група: КП-02

Перевірив: Погорєлов Володимир

Володимирович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1 семестр 2022/2023

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський політехнічний інститут імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра програмного забезпечення комп’ютерних систем

|  |  |
| --- | --- |
| Узгоджено  Керівник роботи  \_\_\_\_\_\_\_\_\_\_/Погорєлов.В.В./ | ЗАХИЩЕНА "\_\_"\_\_\_\_\_\_\_\_\_2022р.  з оцінкою\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_/ Погорєлов.В.В./ |

**Програмний додаток серверу що міститься у Docker-контейнері, та тести написані з використанням Robot framework**

Виконавець роботи

Щербатюк Ярослав Сергійович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023р.

**Код**

|  |
| --- |
| api.py(сервер) |
| **from** flask **import** Flask, request, jsonify  **from** directory **import** Directory      app = Flask(\_\_name\_\_)    root = Directory('root', 100)  deleted\_list = []    @app.route('/directory', methods=['POST', 'GET', 'PATCH', 'DELETE'])  **def** directory():  **if** request.method == 'POST':  **if** any(x.name == request.args.get('name') **for** x **in** root.list) **or** request.args.get('name') == 'root':  **return** jsonify({  "message": "Directory already exists.",  }), 400  dir = Directory(request.args.get('name'), request.args.get('max\_elems'), root)  **return** jsonify({  "message": "Directory created successfully.",  "directory": {  "parent": str(dir.parent),  "name": str(dir.name),  "DIR\_MAX\_ELEMS": int(dir.DIR\_MAX\_ELEMS),  "count\_elems": int(dir.count\_elems),  "list": str(dir.list)  }  }), 201    **elif** request.method == 'GET':  **if** any(dir.name == request.args.get('name') **for** dir **in** root.list) **or** request.args.get('name') == 'root':  **if** request.args.get('name') == 'root':  dir = root  **else**:  dir = next(x **for** x **in** root.list **if** x.name == request.args.get('name'))  **return** jsonify({  "message": "Directory was read successfully.",  "directory": {  "parent": str(dir.parent),  "name": str(dir.name),  "DIR\_MAX\_ELEMS": int(dir.DIR\_MAX\_ELEMS),  "count\_elems": int(dir.count\_elems),  "list": str(dir.list)  }  }), 200  **return** jsonify({  "message": "Directory doesn't exist.",  }), 400    **elif** request.method == 'PATCH':  **if** any(dir.name == request.args.get('name') **for** dir **in** root.list):  dir = next(x **for** x **in** root.list **if** x.name == request.args.get('name'))  dir.move(root)  **return** jsonify({  "message": "Directory moved successfully.",  "directory": {  "parent": str(dir.parent.name),  "name": str(dir.name),  "DIR\_MAX\_ELEMS": int(dir.DIR\_MAX\_ELEMS),  "count\_elems": int(dir.count\_elems),  "list": str(dir.list)  }  }), 200  **return** jsonify({  "message": "Directory doesn't exist.",  }), 400    **else**:  **if** request.args.get('name') **not** **in** deleted\_list **and** any(dir.name == request.args.get('name') **for** dir **in** root.list):  dir = next(x **for** x **in** root.list **if** x.name == request.args.get('name'))  **del** dir  deleted\_list.append(request.args.get('name'))  **return** jsonify({  "message": "Directory deleted successfully.",  }), 200  **return** jsonify({  "message": "Directory was not deleted.",  }), 400    **if** \_\_name\_\_ == '\_\_main\_\_':  app.run(host='0.0.0.0') |

|  |
| --- |
| main.py (cli) |
| **import** sys  **import** pip.\_vendor.requests **as** requests    *# total arguments*  n = len(sys.argv)    Request = "http://localhost:5000/" + sys.argv[2] + "?"    **for** i **in** range(3, n):  Request += sys.argv[i]  **if** i != n-1:  Request += "&"    response = ""  **if** sys.argv[1] == "get":  response = requests.get(Request)  **if** sys.argv[1] == "post":  response = requests.post(Request)  **if** sys.argv[1] == "patch":  response = requests.patch(Request)  **if** sys.argv[1] == "delete":  response = requests.delete(Request)      **print**("Status code:", response.status\_code)  **print**("Response:", response.json()) |

|  |
| --- |
| Robot.robot |
| \*\*\* Settings \*\*\*  Library Process  Library OperatingSystem  \*\*\* Variables \*\*\*  ${cli} /home/git/qa\_kp01\_Pidruchnyy/lab3/main.py    \*\*\* Test Cases \*\*\*  Dir root create  ${result} = Run Process python3 ${cli} get directory name\=root  Should Contain ${result.stdout} Status code: 200    Dir inner create  ${result} = Run Process python3 ${cli} post directory parent\=root name\=child max\_elems\=8  Should Contain ${result.stdout} Status code: 201    Dir create **not** exists  ${result} = Run Process python3 ${cli} post directory parent\=root name\=child max\_elems\=8  Should Contain ${result.stdout} Status code: 400    Dir move  ${result} = Run Process python3 ${cli} patch directory name\=child parent\=root  Should Contain ${result.stdout} Status code: 200    Dir delete  ${result} = Run Process python3 ${cli} delete directory name\=child  Should Contain ${result.stdout} Status code: 200    Dir delete **not** exists  ${result} = Run Process python3 ${cli} delete directory name\=child  Should Contain ${result.stdout} Status code: 400 |

**Висновок**

У ході виконання даної курсової роботи було розроблено сервер на мові програмування python, розташувано його у докер контейнері, а також розроблено інтерфейс командного рядка для створення запитів до серверу та тестування за допомогою Robot framework.