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: Робейко Роман Іванович

Группа: 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

Робейко Роман Іванович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 р.

**Код**

| api.py |
| --- |
| 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 |
| --- |
| import sys  import pip.\_vendor.requests as requests  # total arguments  n = len(sys.argv)  Request = "http://localhost:8888/" + 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 |
| --- |
| Поки не виконано |

**Висновок:** В процесі виконання роботи я розробив сервер на мові програмування python, помістив його у докер контейнер, розробив інтерфейс командного рядка для створення запитів до серверу та тестування за допомогою фреймворку Robot.