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 Якобчук Роман Валерійович

Студент III курсу групи КП-02

Перевірив: Погорєлов Володимир Володимирович

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

Програмний додаток серверу, що міститься у Docker контейнері, та тести

Написані з використанням Robot framework

Київ 2023

Код

|  |
| --- |
| Server – api.py |
| **from flask import Flask, request, jsonify**  **from flask\_restful import Resource, Api**  **import json**  **from directory import Directory**  **from binaryFile import BinaryFile**  **from logTextFile import LogTextFile**  **from bufferFile import BufferFile**  **app = Flask(\_\_name\_\_)**  **api = Api(app)**  **parentDirectory = Directory('root', 100)**  **# directory**  **directoryName = 'item\_1'**  **maxElements = 20**  **directory = Directory(directoryName, maxElements)**  **# binary**  **fileName = 'binary file'**  **information = 'binary information'**  **binary = BinaryFile(fileName, information, parentDirectory)**  **# logTextFile**  **fileName = 'item\_1'**  **log = LogTextFile(fileName, parentDirectory)**  **# buffer**  **fileName = 'buffer\_1'**  **maxFileSize = 20**  **buffer = BufferFile(fileName, maxFileSize)**  **class ApiServerWork(Resource):**  **@staticmethod**  **def get():**  **return "Server work"**  **class ApiDirectory(Resource):**  **def \_\_init\_\_(self):**  **self.directory = directory**  **@property**  **def post(self):**  **params = request.get\_json()**  **self.directory = Directory(params["name"], params["maxElements"])**  **return jsonify({**  **'Message': 'The Directory was created successfully'**  **})**  **def put(self):**  **data = request.get\_json()**  **parent\_directory = Directory(data["parent"])**  **return self.directory.\_\_move\_\_(parent\_directory)**  **def delete(self):**  **return self.directory.\_\_delete\_\_()**  **class ApiBinary(Resource):**  **def \_\_init\_\_(self):**  **self.binary = binary**  **def get(self):**  **return self.binary.\_\_read\_\_()**  **def post(self):**  **params = request.get\_json()**  **parent\_directory = Directory(params["parent"])**  **self.binary = BinaryFile(params["name"], params["information"], parent\_directory)**  **return jsonify({**  **'Message': 'The BinaryFile was created successfully'**  **})**  **def put(self):**  **params = request.get\_json()**  **parent\_directory = Directory(params["parent"])**  **return self.binary.\_\_move\_\_(parent\_directory)**  **def delete(self):**  **return self.binary.\_\_delete\_\_()**  **class ApiLogTextFile(Resource):**  **def \_\_init\_\_(self):**  **self.logText = log**  **def get(self):**  **return self.logText.\_\_read\_\_()**  **def post(self):**  **params = request.get\_json()**  **parent\_directory = Directory(params["parent"])**  **self.logText = LogTextFile(params["name"], parent\_directory)**  **return jsonify({**  **'Message': 'The LogTextFile was created successfully'**  **})**  **def put(self):**  **params = request.get\_json()**  **parent\_directory = Directory(params["parent"])**  **return self.logText.\_\_move\_\_(parent\_directory)**  **def patch(self):**  **params = request.get\_json()**  **return self.logText.\_\_log\_\_(params["line"])**  **def delete(self):**  **return self.logText.\_\_delete\_\_()**  **class ApiBuffer(Resource):**  **def \_\_init\_\_(self):**  **self.buffer = buffer**  **def get(self):**  **return self.buffer.\_\_consume\_\_()**  **def post(self):**  **params = request.get\_json()**  **parent\_directory = Directory(params["parent"])**  **self.buffer = BufferFile(params["name"], params["maxFileSize"], parent\_directory)**  **return jsonify({**  **'Message': 'The BufferFile was created successfully'**  **})**  **def put(self):**  **params = request.get\_json()**  **parent\_directory = Directory(params["parent"])**  **return self.buffer.\_\_move\_\_(parent\_directory)**  **def patch(self):**  **params = json.loads(request.data)**  **return self.buffer.\_\_push\_\_(params["element"])**  **def delete(self):**  **return self.buffer.\_\_delete\_\_()**  **api.add\_resource(ApiServerWork, '/')**  **api.add\_resource(ApiDirectory, '/directory')**  **api.add\_resource(ApiBinary, '/binary')**  **api.add\_resource(ApiLogTextFile, '/log')**  **api.add\_resource(ApiBuffer, '/buffer')**  **if \_\_name\_\_ == '\_\_main\_\_':**  **app.run(debug=True)** |

|  |
| --- |
| CLI – main.py |
| **import sys**  **import requests**  **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] == "put":**  **response = requests.put(Request)**  **if sys.argv[1] == "delete":**  **response = requests.delete(Request)**  **print("Status code:", response.status\_code)**  **print("Response:", response.json())** |

|  |
| --- |
| Tests – Robot.robot |
| \*\*\* Settings \*\*\*  Library OperatingSystem  Library Process  \*\*\* Variables \*\*\*  ${main} /home/roman/qa-kp02-yakobchuk/lab3/main.py  \*\*\* Test Cases \*\*\*  #Directory  Dir create  ${result} = Run Process python3 ${main} post directory name\=root maxElements\=100  Dir move  ${result} = Run Process python3 ${main} patch directory name\=name parent\=root  Dir delete  ${result} = Run Process python3 ${main} delete directory name\=name  Dir read  ${result} = Run Process python3 ${main} get directory name\=name parent\=root  #Binary  Bin create  ${result} = Run Process python3 ${main} post binaryFile name\=bin parent\=root information\=hello  Bin move  ${result} = Run Process python3 ${main} patch binaryFile name\=bin parent\=root  Bin delete  ${result} = Run Process python3 ${main} delete binaryFile name\=bin  Bin read  ${result} = Run Process python3 ${main} get binaryFile name\=bin parent\=root  #LogTextFile  Log create  ${result} = Run Process python3 ${main} post logTextFile name\=log parent\=root information\=hello  Log move  ${result} = Run Process python3 ${main} patch logTextFile name\=log parent\=root  Log delete  ${result} = Run Process python3 ${main} delete logTextFile name\=log  Log read  ${result} = Run Process python3 ${main} get logTextFile name\=log parent\=root  #Buffer  Buf create  ${result} = Run Process python3 ${main} post bufferFile name\=buf parent\=root information\=hello  Buf move  ${result} = Run Process python3 ${main} patch bufferFile name\=buf parent\=root  Buf delete  ${result} = Run Process python3 ${main} delete bufferFile name\=buf  Buf read  ${result} = Run Process python3 ${main} get bufferFile name\=buf parent\=root |

**Висновок**

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