**MINISTERUL EDUСAŢIEI ŞI СERСETĂRII AL REPUBLIСII MOLDOVA**

**UNIVERSITATEA DE STAT „ALEСU RUSSO” DIN BĂLŢI**

**FAСULTATEA DE ŞTIINŢE REALE, EСONOMIСE ȘI ALE MEDIULUI**

**СATEDRA DE MATEMATIСĂ ȘI INFORMATIСĂ**

**LUCRARE DE LABORATOR NR. 2**

**„JSON”**

**Curs „Platforma JAVA Enterprise II”**

**Coordonator:**

**Olga CERBU**

dr.,conf.univ

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

(semnătura)

**Autor:**

Studentul grupei AW21M

**Dumitru BAZAOСHI**

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

(semnătura)

**BĂLȚI, 2023**

**CUPRINS**

[Scop. Condiția problemei 2](#_Toc147397881)

[Ce este JSON? 2](#_Toc147397882)

[Caracteristicile cheie ale JSON 2](#_Toc147397883)

[Exemplu de JSON file 2](#_Toc147397884)

[Maven dependența pentru JSON.simple 3](#_Toc147397885)

[Cod Java fișier WriteJSON 3](#_Toc147397886)

[Cod Java fișier ReadJSON 6](#_Toc147397887)

[Rezultate Consolă 9](#_Toc147397888)

[Concluzii 10](#_Toc147397889)

[Referințe bibliografice 10](#_Toc147397890)

Scop. Condiția problemei

Scopul lucrării de laborator este de a învăța și a implementa citirea și procesarea datelor dintr-un fișier JSON folosind limbajul JAVA. Astfel întelegerea structurii JSON, studierea tehnologiilor disponibile pentru procesarea unui JSON, interpretarea rezultatelor și formularea concluziilor.

Ce este JSON?

JSON (JavaScript Object Notation) este un format ușor de citit și de scris pentru transmiterea datelor între un client și un server. Este un format de date text bazat pe JavaScript, dar poate fi folosit și în alte limbaje de programare. JSON este popular pentru că este simplu și ușor de înțeles, fiind utilizat în dezvoltarea aplicațiilor web și a serviciilor web.

Caracteristicile cheie ale JSON

* **Structura de date ușor de înțeles**: JSON utilizează o sintaxă simplă și concisă, care este ușor de citit și de înțeles pentru oameni.
* **Structura ierarhică**: JSON permite încapsularea datelor într-o structură ierarhică formată din obiecte și matrice (liste).
* **Perechi cheie-valoare**: Datele în JSON sunt reprezentate sub forma de perechi cheie-valoare, ceea ce facilitează accesul și manipularea acestora.
* **Portabilitate**: JSON este independent de limbaj și poate fi folosit într-o varietate de limbaje de programare.

Exemplu de JSON file

{

"artists" : [

{

"artistname" : "Deep Purple",

"albums" : [

{

"albumname" : "Machine Head",

"year" : "1972",

"genre" : "Rock"

},

{

"albumname" : "Stormbringer",

"year" : "1974",

"genre" : "Rock"

}

]

}

]

Maven dependența pentru JSON.simple

<dependencies>

<dependency>

<groupId>com.googlecode.json-simple</groupId>

<artifactId>json-simple</artifactId>

<version>1.1.1</version>

</dependency>

</dependencies>

Cod Java fișier WriteJSON

package org.example;

import java.io.FileWriter;

import java.io.IOException;

import org.json.simple.JSONArray;

import org.json.simple.JSONObject;

public class WriteJSON {

@SuppressWarnings("unchecked")

public static void main(String[] args)

{

// Studentul 1

// Crearea unui obiect nou JSON numit studentDetails

JSONObject studentDetails = new JSONObject();

studentDetails.put("id", "101");

studentDetails.put("firstname", "Olga");

studentDetails.put("lastname", "Criclivaia");

studentDetails.put("subject", "Math");

studentDetails.put("marks", "9");

// Crearea unui obiect nou JSON numit studentObject

JSONObject studentObject = new JSONObject();

studentObject.put("student", studentDetails);

// Studentul 2

JSONObject studentDetails2 = new JSONObject();

studentDetails2.put("id", "102");

studentDetails2.put("firstname", "Valentina");

studentDetails2.put("lastname", "Advahova");

studentDetails2.put("subject", "Chemistry");

studentDetails2.put("marks", "8");

JSONObject studentObject2 = new JSONObject();

studentObject2.put("student", studentDetails2);

// Studentul 3

JSONObject studentDetails3 = new JSONObject();

studentDetails3.put("id", "103");

studentDetails3.put("firstname", "Alina");

studentDetails3.put("lastname", "Gutul");

studentDetails3.put("subject", "English");

studentDetails3.put("marks", "10");

JSONObject studentObject3 = new JSONObject();

studentObject3.put("student", studentDetails3);

// Studentul 4

JSONObject studentDetails4 = new JSONObject();

studentDetails4.put("id", "104");

studentDetails4.put("firstname", "Nadejda");

studentDetails4.put("lastname", "Ianciuc");

studentDetails4.put("subject", "Physics");

studentDetails4.put("marks", "7");

JSONObject studentObject4 = new JSONObject();

studentObject4.put("student", studentDetails4);

// Adaugarea studentilor in lista

JSONArray studentList = new JSONArray();

studentList.add(studentObject);

studentList.add(studentObject2);

studentList.add(studentObject3);

studentList.add(studentObject4);

// Inscriere in Json file

// Crearea unui obiect Filewriter pentru deschiderea fisierului students.json

try (FileWriter file = new FileWriter("students.json")) {

// Metoda write din FileWriter pentru a scrie instanta JSONArray sau JSONObject într-un fișier

file.write(studentList.toJSONString());

file.flush();

} catch (IOException e) {

e.printStackTrace();

}

}

}

Cod Java fișier ReadJSON

package org.example;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.IOException;

import org.json.simple.JSONArray;

import org.json.simple.JSONObject;

import org.json.simple.parser.JSONParser;

import org.json.simple.parser.ParseException;

public class ReadJSON {

@SuppressWarnings("unchecked")

public static void main(String[] args)

{

// Crearea unui obiect de analiză JSON pentru a analiza fișierul citit

JSONParser jsonParser = new JSONParser();

try (FileReader reader = new FileReader("students.json"))

{

// Citirea JSON file

// Parsarea JSON din fișier

Object obj = jsonParser.parse(reader);

// Conversia obiectului la JSONArray

JSONArray studentList = (JSONArray) obj;

System.out.println(studentList); // Afișarea listei de studenți

// Iterarea prin lista de studenți și parsarea fiecărui obiect student

studentList.forEach(st -> parseStudentObject((JSONObject) st));

// Tratarea excepției

} catch (FileNotFoundException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

} catch (ParseException e) {

e.printStackTrace();

}

}

private static void parseStudentObject(JSONObject student)

{

// Obținerea obiectul elevului în listă

JSONObject studentObject = (JSONObject) student.get("student");

// Obținerea id studentului

String id = (String) studentObject.get("id");

System.out.println("\nStudent id: " + id);

// Obținerea numele studentului

String firstname = (String) studentObject.get("firstname");

System.out.println("First Name: " + firstname);

// Obținerea prenumele studentului

String lastname = (String) studentObject.get("lastname");

System.out.println("Last Name: " + lastname);

// Obținerea obiectul studentului

String subject = (String) studentObject.get("subject");

System.out.println("Subject: " + subject);

// Obținerea notei studentului

String marks = (String) studentObject.get("marks");

System.out.println("Marks: " + marks);

}

}

**JSON file creat**

[

{

"student": {

"firstname": "Olga",

"subject": "Math",

"id": "101",

"marks": "9",

"lastname": "Criclivaia"

}

},

{

"student": {

"firstname": "Valentina",

"subject": "Chemistry",

"id": "102",

"marks": "8",

"lastname": "Advahova"

}

},

{

"student": {

"firstname": "Alina",

"subject": "English",

"id": "103",

"marks": "10",

"lastname": "Gutul"

}

},

{

"student": {

"firstname": "Nadejda",

"subject": "Physics",

"id": "104",

"marks": "7",

"lastname": "Ianciuc"

}

}

]

Rezultate Consolă

*Rezultatele obținute în Consolă sunt:*

[{"student":{"firstname":"Olga","subject":"Math","id":"101","marks":"9","lastname":"Criclivaia"}},{"student":{"firstname":"Valentina","subject":"Chemistry","id":"102","marks":"8","lastname":"Advahova"}},{"student":{"firstname":"Alina","subject":"English","id":"103","marks":"10","lastname":"Gutul"}},{"student":{"firstname":"Nadejda","subject":"Physics","id":"104","marks":"7","lastname":"Ianciuc"}}]

Student id: 101

First Name: Olga

Last Name: Criclivaia

Subject: Math

Marks: 9

Student id: 102

First Name: Valentina

Last Name: Advahova

Subject: Chemistry

Marks: 8

Student id: 103

First Name: Alina

Last Name: Gutul

Subject: English

Marks: 10

Student id: 104

First Name: Nadejda

Last Name: Ianciuc

Subject: Physics

Marks: 7

Concluzii

În concluzie, această lucrare de laborator a avut ca obiectiv explorarea și aplicarea tehnicilor de scriere și citire a datelor dintr-un fișier JSON în limbajul de programare Java. Am înțeles ce este un Json file, cum el poate fi creat, cum el poate fi citit, care sunt caracteristicile principale a unui JSON file.

Referințe bibliografice

1. *JSON* [online] [vizitat 05.10.2023]. Disponibil: <https://www.json.org/json-en.html>
2. *What is JSON?* [online] [vizitat 05.10.2023]. Disponibil: <https://www.w3schools.com/whatis/whatis_json.asp>
3. *JSON.*simple *– Read and Write JSON* [online] [vizitat 05.10.2023]. Disponibil: <https://howtodoinjava.com/java/library/json-simple-read-write-json-examples/>