**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. 3**

**„Serializare/Deserializare”**

**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](#_Toc147754349)

[Ce este Serializarea și Deserializarea? 2](#_Toc147754350)

[Clasele ObjectInputStream și ObjectOutputStream 2](#_Toc147754351)

[Cod Java – Clasa Employee 2](#_Toc147754352)

[Cod Java – Clasa Serialization 3](#_Toc147754353)

[Cod Java – Clasa Deserialization 4](#_Toc147754354)

[Rezultate Consolă 5](#_Toc147754355)

[Concluzii 6](#_Toc147754356)

[Referințe bibliografice 6](#_Toc147754357)

Scop. Condiția problemei

Scopul acestei lucrării de laborator axate pe serializare și deserializare în Java este de a căpăta o înțelegere profundă a acestor concepte și de a învăța cum să utilizez acest mecanism puternic pentru a salva și restaura obiecte, ceea ce este crucial în dezvoltarea aplicațiilor robuste și eficiente. De asemenea studierea interfeti Serializable, implementarea serializării cât și implementarea deserializării, interpretarea rezultatelor și formularea concluziilor.

Ce este Serializarea și Deserializarea?

**Serializarea** este o metodă prin care se pot salva datele împreună cu signatura unui obiect. Folosind această operație se poate salva într-un fișier ca o secvență de octeți din care poate fi refăcut ulterior obiectul original.

Procesul invers, de citire a unui obiect serializat pentru a-i reface starea originală, se numește **deserializare**.

Clasele ObjectInputStream și ObjectOutputStream

Clasele **ObjectInputStream** și **ObjectOutputStream** sunt fluxuri de nivel înalt care conțin metodele de serializare și deserializare a unui obiect.

Clasa ObjectOutputStream conține multe metode de scriere pentru scrierea diferitelor tipuri de date, dar o metodă în special iese în evidență

* *public final void writeObject(Object x) throws IOException*

Metoda de mai sus serializează un obiect și îl trimite la fluxul de ieșire. În mod similar, clasa ObjectInputStream conține următoarea metodă de deserializare a unui obiect

* *public final Object readObject() throws IOException, ClassNotFoundException*

Cod Java – Clasa Employee

public class Employee implements java.io.Serializable {

public String name; // Numele angajatului

public String address; // Adresa angajatului

public transient int SSN; // Numărul de asigurare socială (nu va fi serializat)

public int number; // Numărul angajatului

public void mailCheck() {

System.out.println("Expedierea mesajului către " + name + " " + address);

}

}

Cod Java – Clasa Serialization

*Creare class Java pentru serializarea unui obiect:*

import java.io.\*;

public class Serialization {

public static void main(String [] args) {

// Crearea unui obiect de tip Employee și inițializarea datelor acestuia

Employee e = new Employee();

e.name = "Bazaochi Dumitru";

e.address = "Balti, Moldova";

e.SSN = 11122333;

e.number = 101;

String filename = "employee.ser";

try {

// Crearea unui flux de ieșire către fișierul employee.ser

FileOutputStream fileOut = new FileOutputStream(filename);

// Crearea unui obiect ObjectOutputStream pentru a scrie obiectul Employee în fluxul de ieșire

ObjectOutputStream out = new ObjectOutputStream(fileOut);

// Scrierea obiectului Employee în fluxul de ieșire

out.writeObject(e);

// Închiderea obiectului ObjectOutputStream și a fluxului de ieșire

out.close();

fileOut.close();

// Afișarea unui mesaj pentru a indica că datele serializate au fost salvate cu succes

System.out.printf("Datele serializate sunt salvate în employee.ser");

} catch (IOException i) {

// Tratarea excepțiilor de IO și afișarea stivei de apeluri

i.printStackTrace();

}

}

}

Cod Java – Clasa Deserialization

*Creare class Java pentru deserializarea unui obiect:*

import java.io.\*;

public class Deserialization {

public static void main(String [] args) {

// Inițializarea unei referințe de tip Employee cu null

Employee e = null;

String filename = "employee.ser";

try {

// Deschiderea unui flux de intrare către fișierul employee.ser

FileInputStream fileIn = new FileInputStream(filename);

// Crearea unui obiect ObjectInputStream pentru a citi obiectul serializat din fluxul de intrare

ObjectInputStream in = new ObjectInputStream(fileIn);

// Deserializarea obiectului și conversia acestuia la tipul Employee

e = (Employee) in.readObject();

// Închiderea obiectului ObjectInputStream și a fluxului de intrare

in.close();

fileIn.close();

} catch (IOException i) {

// Tratarea excepțiilor de IO și afișarea stivei de apeluri

i.printStackTrace();

return;

} catch (ClassNotFoundException c) {

// Tratarea excepției de clasă neencontrată (ClassNotFoundException) și afișarea stivei de apeluri

System.out.println("Clasa Employee nu este găsită");

c.printStackTrace();

return;

}

// Afișarea informațiilor despre Employeeul deserializat

System.out.println("Employee deserializat ...");

System.out.println("Nume: " + e.name);

System.out.println("Adresa: " + e.address);

System.out.println("SSN: " + e.SSN);

System.out.println("Număr: " + e.number);

}

}

Rezultate Consolă

*Rezultatele obținute în Consolă sunt:*

Employee deserializat ...

Nume: Bazaochi Dumitru

Adresa: Balti, Moldova

SSN: 0

Număr: 101

Concluzii

În concluzie, această lucrare de laborator a avut ca scop înțelegerea profundă și o experiență practică în utilizarea acestor procese fundamentale în manipularea obiectelor și a datelor în Java. Am înțeles conceptele de serializare și deserializare, cum putem salva și restaura starea obiectelor în urma proceselor efectuate.

Referințe bibliografice

1. *Serializarea obiectelor in Java* [online] [vizitat 09.10.2023]. Disponibil: <https://www.scritub.com/stiinta/informatica/java/Serializarea-obiectelor-in-Jav85778.php>
2. *Serizalizarea obiectelor* [online] [vizitat 09.10.2023]. Disponibil: <https://profs.info.uaic.ro/~acf/java/slides/extra/serializare_slide.pdf>
3. *Java - Serialization* [online] [vizitat 09.10.2023]. Disponibil: <https://www.tutorialspoint.com/java/java_serialization.htm>