**Part 1**

**Question 1**

import java.io.FileInputStream;

import java.io.IOException;

public class LectureFichierBinaire {

public static void main(String[] args) {

String cheminFichier = "fichier\_binaire.dat"; // Remplacez par le chemin de votre fichier binaire

try (FileInputStream fis = new FileInputStream(cheminFichier)) {

int octet;

System.out.println("Contenu du fichier binaire (en octets) :");

while ((octet = fis.read()) != -1) {

System.out.print(octet + " "); // Affiche chaque octet en tant que nombre

}

} catch (IOException e) {

System.err.println("Erreur lors de la lecture du fichier : " + e.getMessage());

}

}

}

**Qestion 2**

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

public class CopierFichierBinaire {

public static void main(String[] args) {

String sourceFichier = "source\_image.jpg"; // Chemin du fichier source

String destinationFichier = "copie\_image.jpg"; // Chemin du fichier de destination

try (

FileInputStream fis = new FileInputStream(sourceFichier);

FileOutputStream fos = new FileOutputStream(destinationFichier)

) {

byte[] buffer = new byte[1024]; // Tampon de 1 Ko pour optimiser la lecture/écriture

int longueur;

System.out.println("Copie du fichier en cours...");

while ((longueur = fis.read(buffer)) != -1) {

fos.write(buffer, 0, longueur); // Écrit uniquement les octets lus

}

System.out.println("Copie terminée. Fichier copié : " + destinationFichier);

} catch (IOException e) {

System.err.println("Erreur lors de la copie du fichier : " + e.getMessage());

}

}

}

**Partie 2**

**Question 3**

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

public class LectureTexte {

public static void main(String[] args) {

String cheminFichier = "livres.txt"; // Remplacez par le chemin de votre fichier texte

try (BufferedReader br = new BufferedReader(new FileReader(cheminFichier))) {

String ligne;

System.out.println("Contenu du fichier texte :");

while ((ligne = br.readLine()) != null) {

System.out.println(ligne); // Affiche chaque ligne du fichier

}

} catch (IOException e) {

System.err.println("Erreur lors de la lecture du fichier : " + e.getMessage());

}

}

}

**Qestion 4**

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

public class LectureEtEcritureTexte {

public static void main(String[] args) {

String fichierSource = "livres.txt"; // Chemin du fichier source

String fichierDestination = "sortie.txt"; // Chemin du fichier de destination

try (

BufferedReader br = new BufferedReader(new FileReader(fichierSource));

BufferedWriter bw = new BufferedWriter(new FileWriter(fichierDestination))

) {

String ligne;

System.out.println("Lecture et écriture en cours...");

while ((ligne = br.readLine()) != null) {

bw.write(ligne); // Écrit la ligne dans le fichier de destination

bw.newLine(); // Ajoute une nouvelle ligne

}

System.out.println("Écriture terminée. Résultat enregistré dans : " + fichierDestination);

} catch (IOException e) {

System.err.println("Erreur lors du traitement des fichiers : " + e.getMessage());

}

}

}

**PARTIE 3/**

**Question 5**

import java.io.Serializable;

public class Livre implements Serializable {

private static final long serialVersionUID = 1L; // Version de la classe pour la sérialisation

private String titre;

private String auteur;

private int anneePublication;

// Constructeur

public Livre(String titre, String auteur, int anneePublication) {

this.titre = titre;

this.auteur = auteur;

this.anneePublication = anneePublication;

}

// Getters

public String getTitre() {

return titre;

}

public String getAuteur() {

return auteur;

}

public int getAnneePublication() {

return anneePublication;

}

// toString pour afficher un Livre

@Override

public String toString() {

return "Livre{" +

"titre='" + titre + '\'' +

", auteur='" + auteur + '\'' +

", anneePublication=" + anneePublication +

'}';

}

}

**Question 6**

import java.io.FileOutputStream;

import java.io.ObjectOutputStream;

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

public class SerialisationBibliotheque {

public static void main(String[] args) {

String cheminFichier = "bibliotheque.ser"; // Fichier de sauvegarde

// Liste de livres

List<Livre> bibliotheque = new ArrayList<>();

bibliotheque.add(new Livre("1984", "George Orwell", 1949));

bibliotheque.add(new Livre("Le Petit Prince", "Antoine de Saint-Exupéry", 1943));

bibliotheque.add(new Livre("Harry Potter", "J.K. Rowling", 1997));

// Sérialisation

try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(cheminFichier))) {

oos.writeObject(bibliotheque); // Écrit l'objet sérialisé dans le fichier

System.out.println("Liste de livres sérialisée dans " + cheminFichier);

} catch (IOException e) {

System.err.println("Erreur lors de la sérialisation : " + e.getMessage());

}

}

}

**Question 7**

import java.io.FileInputStream;

import java.io.ObjectInputStream;

import java.io.IOException;

import java.util.List;

public class DeserialisationBibliotheque {

public static void main(String[] args) {

String cheminFichier = "bibliotheque.ser"; // Fichier contenant les objets sérialisés

// Désérialisation

try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(cheminFichier))) {

List<Livre> bibliotheque = (List<Livre>) ois.readObject(); // Lire l'objet

System.out.println("Liste des livres désérialisée :");

for (Livre livre : bibliotheque) {

System.out.println(livre); // Affiche chaque livre

}

} catch (IOException | ClassNotFoundException e) {

System.err.println("Erreur lors de la désérialisation : " + e.getMessage());

}

}

}