Java Input and Output
الدخل و الخرج في لغة الجافا

Dr. REEMA AL-KAMHA

مقدمة

- تمتاز لغة الجافا بتعريفها إطار عمل لتسهيل الدخل و الخرج.

كتابة نص إلى ملف

- **استدعاء الحزمة** **import java.io.*;**
- **إنشاء متغير مرجعي من نوع File يؤشر على غرض من نوع File يتضمن هذا الغرض اسم الملف الذي سيتم التعامل معه و مساره الفعلي (تحديد المسار اختياري) من خلال العبارة البرمجية التالية:**

File file=new File("path\filename.txt")

ملاحظة: لا يعني نجاح إنشاء الغرض أن الملف المحدد بالاسم filename موجود على القرص، إنما الهدف هنا تحديد الملف الذي سيتم التعامل معه.

- **استخدام الصف FileWriter لكتابة البيانات إلى الملف المحدد بالاسم filename:**

FileWriter fr=new FileWriter(file)

إذا كان الملف المحدد بالاسم filename غيرموجود فيتم إنشاؤه، ويتم إنشاء قناة اتصال معه لتأمين عملية الكتابة عليه.

- **استخدام الصف BufferedWriter:** مخازن مؤقتة تعمل على التخزين المؤقت للبيانات قبل ارسالها إلى الملف المحدد بالاسم filename حيث تبقى حتى يصل حجمها إلى مستوى معين، وتستخدم لتقليل عمليات الولوج المكلفة زمنيا إلى القرص.

BufferedWriter bw=new BufferedWriter(fr)

- **الكتابة على الملف:**

bw.write(العبارة المطلوب كتابتها);

- **اغلاق الملف:** **bw.close();**

قراءة نص من ملف

- عملية القراءة من ملف تخضع للقواعد نفسها تقريبا.
- تعريف غرض من النوع File
- تعريف غرض من نوع FileReader (بدلا من FileWriter).
- بهدف زيادة أداء عملية القراءة نستخدم المخزن المؤقت BufferedReader
- قراءة أسطر الملف
- إغلاق الملف

مثال

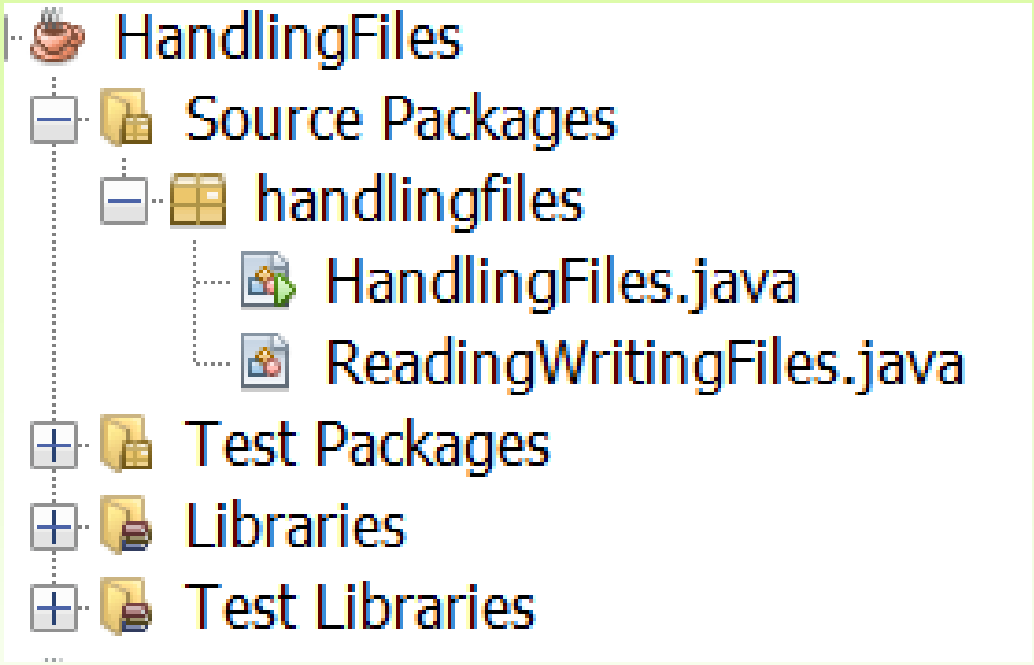
اكتب برنامجا بلغة الجافا للتوصيف التالي:

- **تعريف صف اسمه ReadingWritingFiles**، يحوي:
  - **الطريقة WritingFile** لإنشاء ملف يحدد المستخدم اسمه و مساره، ويكتب فيه الأسطر التالية:

Hello there
I am here
I am learing how to write a file

  - **الطريقة ReadingFile** للقراءة من ملف يحدد المستخدم اسمه ومساره.
- **الصف الرئيسي HandlingFiles** يقوم بإنشاء غرض من الصف السابق و استدعاء الطرق الموجودة فيه.

البرنامج HandlingFiles



الصف ReadingWritingFiles

```
package handlingfiles;
import java.io.*;
import java.util.*;
public class ReadingWritingFiles   {
    public void writingFile() {...17 lines }
    public void ReadingFile() {...19 lines }
}
```

الطريقة writingFile لإنشاء ملف و الكتابة ضمنه

```
public void writingFile(){
  System.out.println("Enter the path and the name of the file: ");
  Scanner sr=new Scanner(System.in);
  String name=sr.nextLine();
  File f=new File(name);
  try{
    FileWriter fw=new FileWriter(f);
    BufferedWriter bw=new BufferedWriter(fw);
    bw.write("Hello There");
    bw.write("\nI am here");
    bw.write("\nI am learning how to write a file");
    bw.close();
  }
  catch(IOException e){
    e.getMessage();
  }
}
```

لإغلاق الملف

الطريقة ReadingFile للقراءة من ملف

```
public void ReadingFile(){
   System.out.println("Enter the path and the name of the file to read from:");
   Scanner sr=new Scanner(System.in);
   String name=sr.nextLine();
   File f=new File(name);
   try{
     FileReader fr=new FileReader(f);
     BufferedReader br=new BufferedReader(fr);
     String st=br.readLine();
     while(st!=null){
       System.out.println(st);
       st=br.readLine();
     }
     br.close();
   }
   catch(IOException e){
     e.getMessage();
   }
}
```

تعريف غرض من نوع الصف File

استخدام الصف FileReader

الصف BufferedReader لزيادة الأداء.

لقراءة أسطر الملف

لإغلاق الملف

الصف الرئيسي HandlingFiles

```
package handlingfiles;
public class HandlingFiles {
  public static void main(String[] args) {
    ReadingWritingFiles frw=new ReadingWritingFiles();
    frw.writingFile();
    frw.ReadingFile();
  }
}
```

الملفات و الأغراض Object Input/Output

- يمكن للملفات أن تتعامل مع بيانات من أنواع أكثر تعقيدا مثل الأغراض.
- تتطلب عملية تخزين الأغراض في الملفات تحويل هذه الأغراض المطلوب تخزينها إلى متتالية من البايتات، تدعى هذه العملية Serialization.
- عند تحويل الأغراض إلى متتالية بايتات فإنه يمكن تخزينها إلى الملف، أو حتى إرسالها عبر الشبكات الحاسوبية إلى وجهة معينة.
- يمكن لاحقا قراءة متتالية البايتات من الملف أو من الشبكة الحاسوبية و استعادة الغرض الأصلي منها.
- تشترط لغة الجافا أن يتبنى صف معين الواجهة Serializable لكي يصبح ممكنا تحويل الأغراض المنشأة منه إلى متتالية من البايتات.
- لا تملك الواجهة Serializable أي طرق أو حقول و لكنها وسيلة لوضع إشارة على الغرض تبين أنه قابل للتحويل إلى متتالية من البايتات.

الملفات و الأغراض Object Input/Output

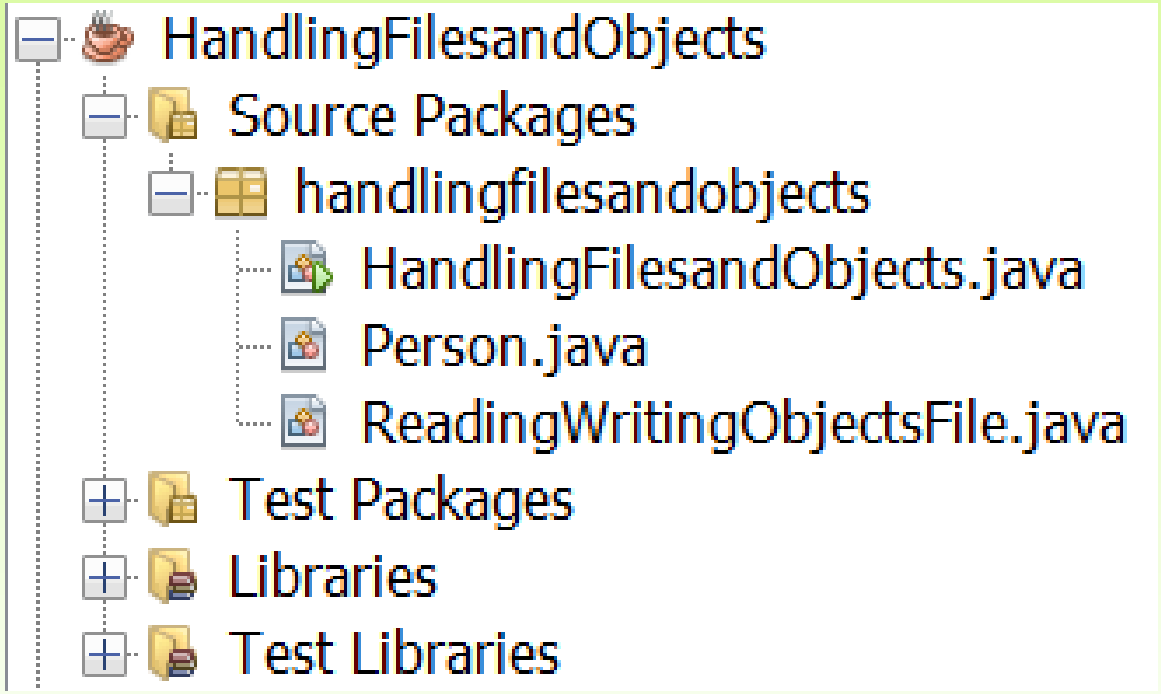
- خطوات تخزين الأغراض إلى الملفات أو قراءتها من هذه الملفات مشابهة لتلك الخاصة بالتعامل مع ملفات الأنواع البسيطة. حيث يتم تعريف متغير مرجعي من نوع File يشير إلى مسار معين في القرص. يتم بعدها استخدام الصف FileOutputStreamلإنشاء قناة لنقل سيل البيانات إلى القرص في حالة الكتابة، و استخدام الصف FileInputStream لإنشاء قناة لنقل سيل البيانات من القرص في حالة القراءة.
- وأخيرا **بهدف زيادة أداء عملية الكتابة على الملف نستخدم المخزن المؤقت** ObjectOutputStream في حال الكتابة على الملف، و نستخدم المخزن المؤقت ObjectInputStream في حال القراءة من الملف.

مثال على كتابة و قراءة ملف يحوي أغراض بلغة الجافا

اكتب برنامجا بلغة الجافا للتوصيف التالي:

- **تعريف صف اسمه Person**، يحوي اسم الشخص و عمره.
- **ReadingWritingObjectsFile**، يحوي:
  - الطريقة **WritingObjectsFile** لإنشاء ملف يحدد المستخدم اسمه و مساره، و يخزن فيه أغراض من الصف Person
  - الطريقة **ReadingObjectsFile** لقراءة أغراض من ملف يحدد المستخدم اسمه ومساره.
- **الصف الرئيسي HandlingFilesandObjects** يقوم بإنشاء غرض من الصف ReadingWritingObjectsFile واستدعاء الطرق الموجودة فيه.

البرنامج HandlingFilesandObjects



الصف Person

```
package handlingfilesandobjects;
import java.io.Serializable;
public class Person implements Serializable {
  private String name;
  private int age;
  Person(String name, int age){
    this.name=name;
    this.age=age;
  }
  String get_name(){
    return name;
  }
  int get_age(){
   return age;
  }
}
```

يتبنى الصف Person الواجهة Serializable وذلك ليتم تحويل الأغراض المنشأة منه إلى سيل من البايتات و بالتالي تخزينها على ملف بشكل صحيح، و بالعكس.

الصف ReadingWritingObjectsFile

```
package handlingfilesandobjects;
import java.io.*;
import java.util.Scanner;
public class ReadingWritingObjectsFile {
  public void writingObjectsFile(){...27 lines }
  public void ReadingObjectsFile(){...19 lines }
}
```

الطريقة WritingObjectsFile

```
public void writingObjectsFile(){
  Scanner sc=new Scanner(System.in);
  String ans;
  System.out.print("Enter the path and the name of the file:");
  File f=new File(sc.nextLine());
  try{
   FileOutputStream fos=new FileOutputStream(f);
   ObjectOutputStream oos=new ObjectOutputStream(fos);
   do{
     System.out.println("Person Information");
     System.out.println("Enter the name of the person:");
     String name=sc.next();
     System.out.println("Enter the age of the person:");
     int age=sc.nextInt();
     Person p=new Person(name, age);
     oos.writeObject((Person)p);
     System.out.println("Do you want to continue Y/N");
     ans=sc.next();
   }
   while(ans.equalsIgnoreCase("Y"));
   oos.close();
  }
  catch(Exception e){
   System.out.println(e.getMessage());
  }
}
```

الطريقة writeObject() و ذلك لكتابة الغرض بشكل صحيح في الملف.

الطريقة ReadingObjectsFile

```
public void ReadingObjectsFile(){
  System.out.println("Enter the path and the name of the file to read from:");
  Scanner sr=new Scanner(System.in);
  String name=sr.nextLine();
  File f=new File(name);
  try{
    FileInputStream fos=new FileInputStream(f);
    ObjectInputStream ois=new ObjectInputStream(fos);
    Person p=(Person)ois.readObject();
    while(p!=null){
     System.out.println(p.get_name()+" "+p.get_age());
     p=(Person)ois.readObject();
    }
    ois.close();
  }
  catch(Exception e){
    System.out.println(e.getMessage());
  }
 }
```

ليتم استعادة الغرض بشكل صحيح نستخدم الطريقة ReadObject()

الصف الرئيسي ReadingObjectsFile

```
package handlingfilesandobjects;
public class HandlingFilesandObjects {
  public static void main(String[] args) {
    ReadingWritingObjectsFile f=new ReadingWritingObjectsFile();
    f.writingObjectsFile();
    f.ReadingObjectsFile();
  }
}
```

البرامج المطلوب قراءتها و تنفيذها و فهمها

- **HandlingFiles**
- **HandlingFilesandObjects**
- **LibrarySystem**