

Files and input/output streams

File: تعد الملفات أهم آلية لتخزين البيانات بشكل دائم على أجهزة تخزين كبيرة السعة. يعني دائماً عدم فقدان البيانات عند إيقاف تشغيل الجهاز. يمكن أن تحتوي الملفات على البيانات بتنسيق يمكن تفسيره بواسطة البرامج ، على سبيل المثال ، برنامج Java مخزن في ملف على القرص الصلب. في هذه الوحدة سنتعامل فقط مع الملفات النصية (Text File).

يتميز كل ملف باسم ودليل يتم وضع الملف فيه (يمكن للمرء أن يأخذ في الاعتبار المسار الكامل الذي يسمح له بالعثور على الملف على القرص الصلب كجزء من اسم الملف). أهم العمليات على الملفات هي: الإنشاء ، القراءة من ، الكتابة إلى.

Object Output/Input: يقوم بقراءة /كتابة أنواع البيانات الأولية وكائنات Java، يكون سهل الاستخدام إذا كنت تريد إرسال بيانات معقدة، و يعتبر Object اسرع مرتين تقريبا من Data لقراءة/كتابة مجموعة ضخمة من بيانات (huge array of double).

العمليات على الملفات

لتنفيذ عمليتي القراءة والكتابة على ملف ، يجب فتح الملف قبل القيام بالعمليات ، والإغلاق بعد أن ننتهي من تشغيله.

*فتح ملف يعني الإشارة إلى نظام التشغيل الذي نريد تشغيله على ملف من داخل برنامج Java ، ويقوم نظام التشغيل بالتحقق مما إذا كانت هذه العمليات ممكنة ومسموح بها. هناك طريقتان لفتح الملف: الفتح للقراءة والفتح للكتابة.

*إغلاق ملف يعني الإشارة إلى نظام التشغيل أن الملف الذي تم فتحه مسبقاً لم يعد مستخدماً من قبل البرنامج. يؤدي إغلاق الملف أيضاً إلى ضمان نقل البيانات المكتوبة في الملف بشكل فعال إلى القرص الصلب.

Writing text files

لكتابة سلسلة نصية في ملف ، يتعين علينا القيام بما يلي:

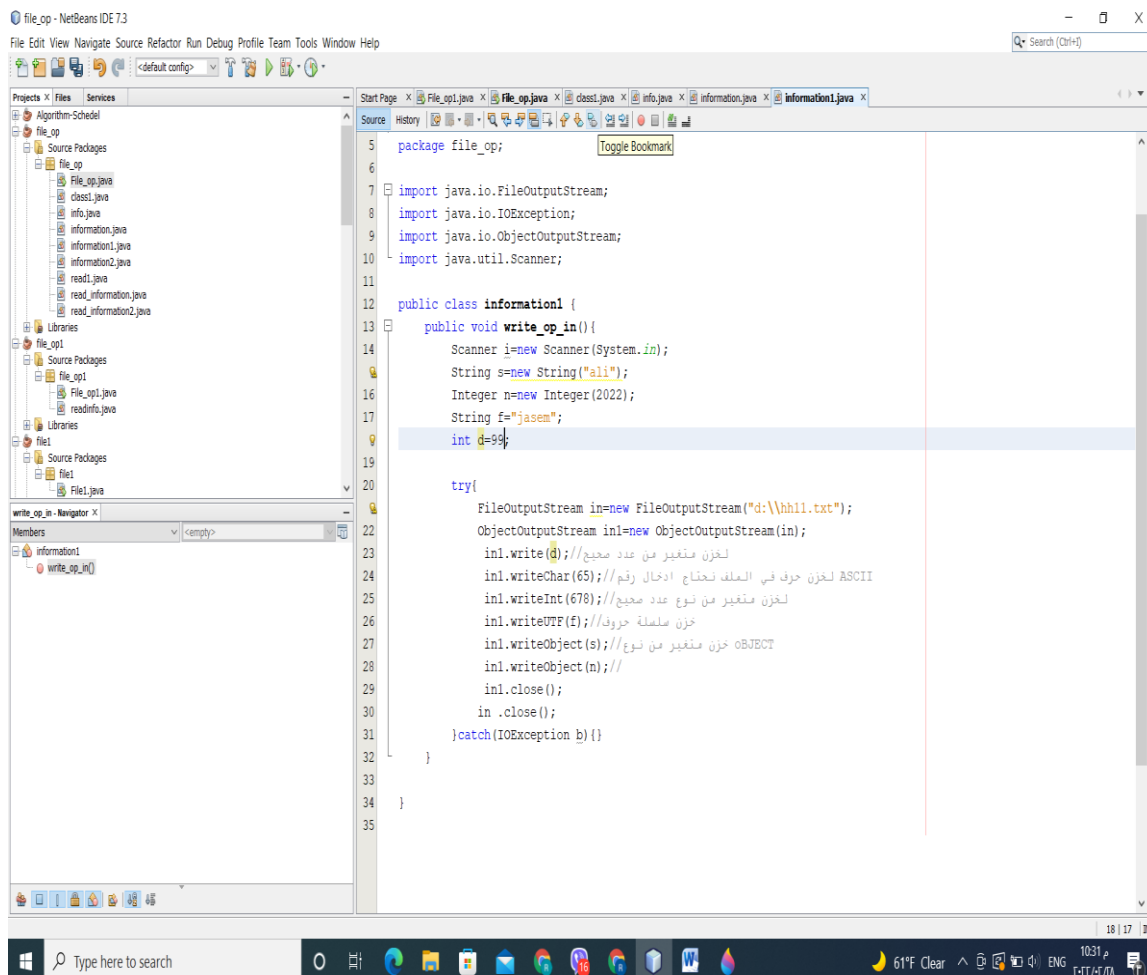
١. افتح الملف للكتابة عن طريق إنشاء كائن من الفئة `FileOutputStream` مرتبط باسم الملف . يتم استخدام فئة `ObjectOutputStream` لكتابة أنواع البيانات الأولية وكائنات Java (بيانات الأولية و بيانات نوع كائن) إلى `OutputStream`. يمكن فقط كتابة الكائنات التي تدعم واجهة `java.io.Serializable` في التدفقات.

٢. كتابة نص على الملف باستخدام الأدوات.

٣. أغلق الملف عند الانتهاء من الكتابة إليه.

Program for writing to a file

تمثل الصورة رقم (١) برنامج لفتح ملف نصي من نوع (Text Documents) على محرك الاقراص (D:\) وكتابة بيانات (البيانات الاعتيادية و بيانات من نوع Object) داخل ملف.



```
5 package file_op;
6
7 import java.io.FileOutputStream;
8 import java.io.IOException;
9 import java.io.ObjectOutputStream;
10 import java.util.Scanner;
11
12 public class information1 {
13     public void write_op_in(){
14         Scanner i=new Scanner(System.in);
15         String s=new String("ali");
16         Integer n=new Integer(2022);
17         String f="jaseem";
18         int d=99;
19
20         try{
21             FileOutputStream in=new FileOutputStream("d:\\hh11.txt");
22             ObjectOutputStream in1=new ObjectOutputStream(in);
23             in1.write(d); // لخصن متغير من عدد صحيح
24             in1.writeChar(65); // لخصن حرف في الملف نحتاج ادخال رقم
25             in1.writeInt(678); // لخصن متغير من نوع عدد صحيح
26             in1.writeUTF(f); // لخصن سلسلة حروف
27             in1.writeObject(s); // لخصن OBJECT من نوع
28             in1.writeObject(n); //
29             in1.close();
30             in.close();
31         }catch(IOException b){}
32     }
33 }
34 }
35 }
```

صورة (١)

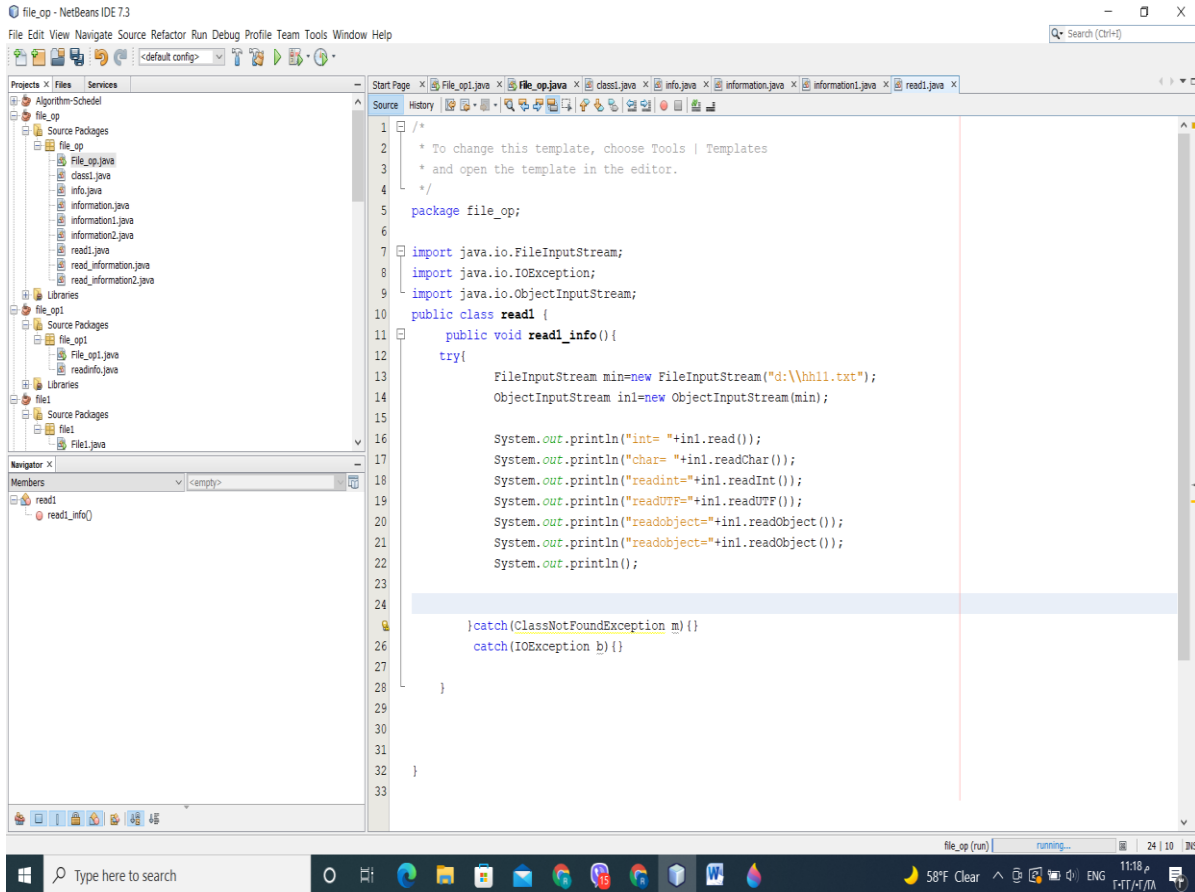
Reading from a text file

لقراءة سلاسل نصية من ملف ، يتعين علينا:

١. افتح الملف للقراءة عن طريق إنشاء كائن من فئة `FileInputStream`. يقوم `ObjectInputStream` بقراءة الكائنات والبيانات الأولية المكتوبة باستخدام `ObjectOutputStream`.
٢. قراءة سطور النص من الملف.
٣. أغلق الملف عند الانتهاء من القراءة منه.

Program reading to a file

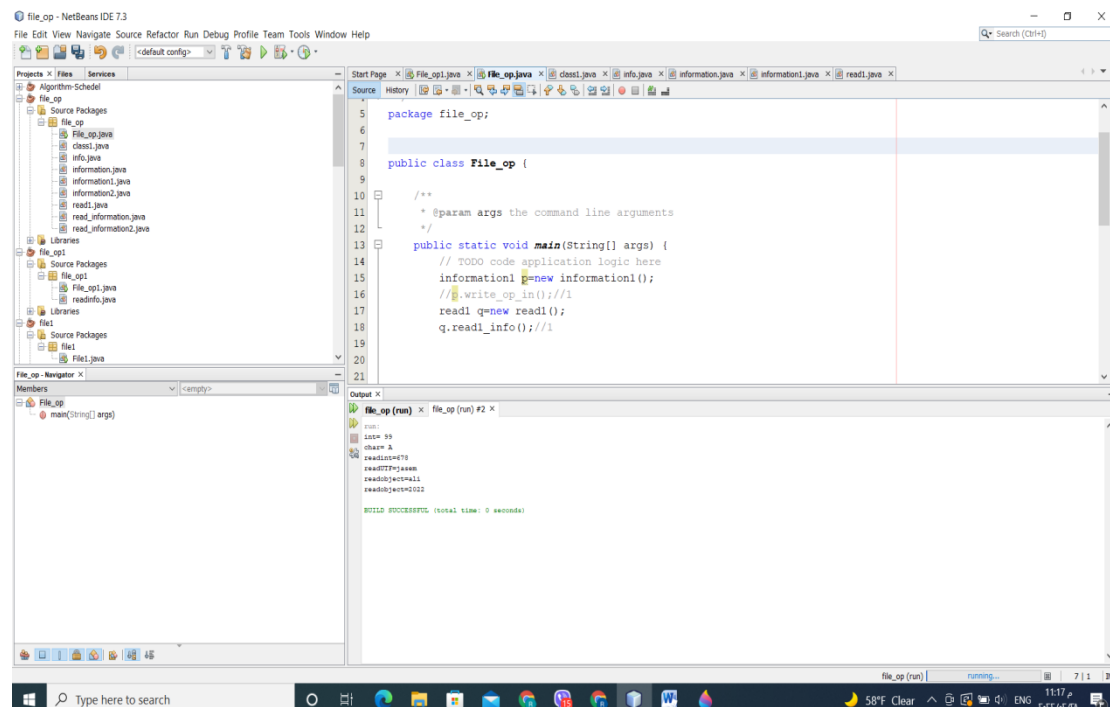
تمثل الصورة رقم (2) برنامج لفتح ملف من نوع (Text Documents) على محرك الاقراص (D:\) وقراءة البيانات من داخل ملف.



```
1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package file_op;
6
7  import java.io.FileInputStream;
8  import java.io.IOException;
9  import java.io.ObjectInputStream;
10 public class read1 {
11     public void read1_info() {
12         try{
13             FileInputStream min=new FileInputStream("d:\\hh11.txt");
14             ObjectInputStream inl=new ObjectInputStream(min);
15
16             System.out.println("int= "+inl.read());
17             System.out.println("char= "+inl.readChar());
18             System.out.println("readint="+inl.readInt());
19             System.out.println("readUTF="+inl.readUTF());
20             System.out.println("readObject="+inl.readObject());
21             System.out.println("readObject="+inl.readObject());
22             System.out.println();
23
24
25         }catch(ClassNotFoundException m){}
26         catch(IOException b){}
27     }
28
29
30
31
32 }
33
```

صورة (٢)

تمثل الصورة رقم (٣) class main حيث فقط تم استدعاء الدوال فيه و كذلك شاشة التنفيذ حيث تم طباعة البيانات التي تم سحبها من الملف على شاشة تنفيذ جافا.



```
package file_op;

public class File_op {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        information1 = new information1();
        // write_op_in(); //1
        readi q=new readl();
        q.readl_info();//1
    }
}
```

Output:

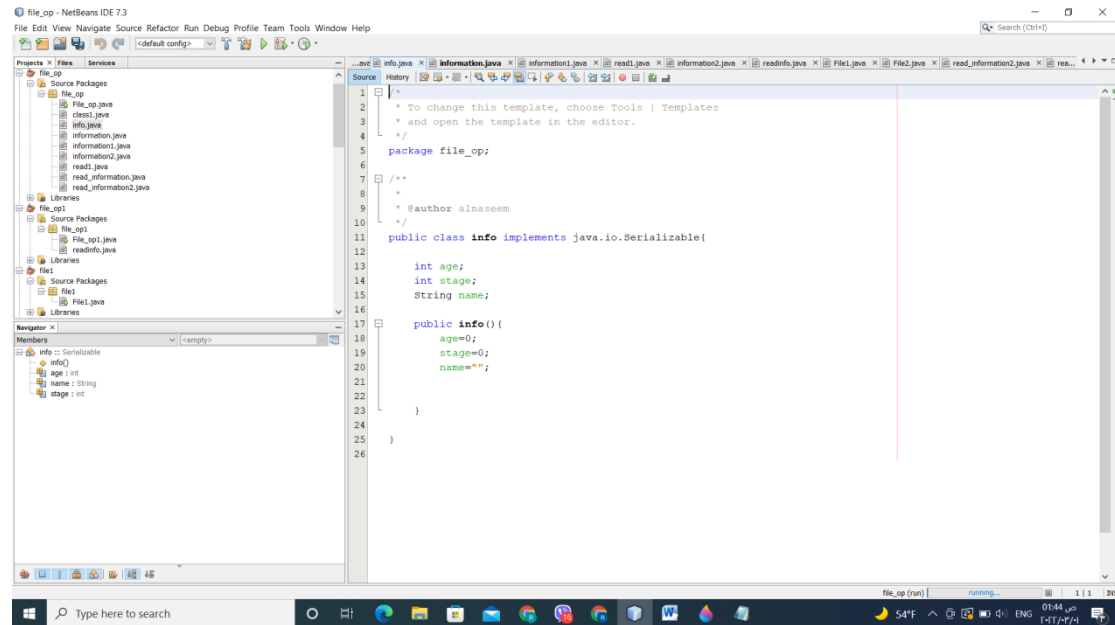
```
file_op (run) x file_op (run) #2 x
run:
INFO: 59
main: 2
readInfo=679
readInfo1=4444
readInfo2=1111
readInfo3=2022
BUILD SUCCESSFUL (total time: 0 seconds)
```

صورة (٣)

Example:-

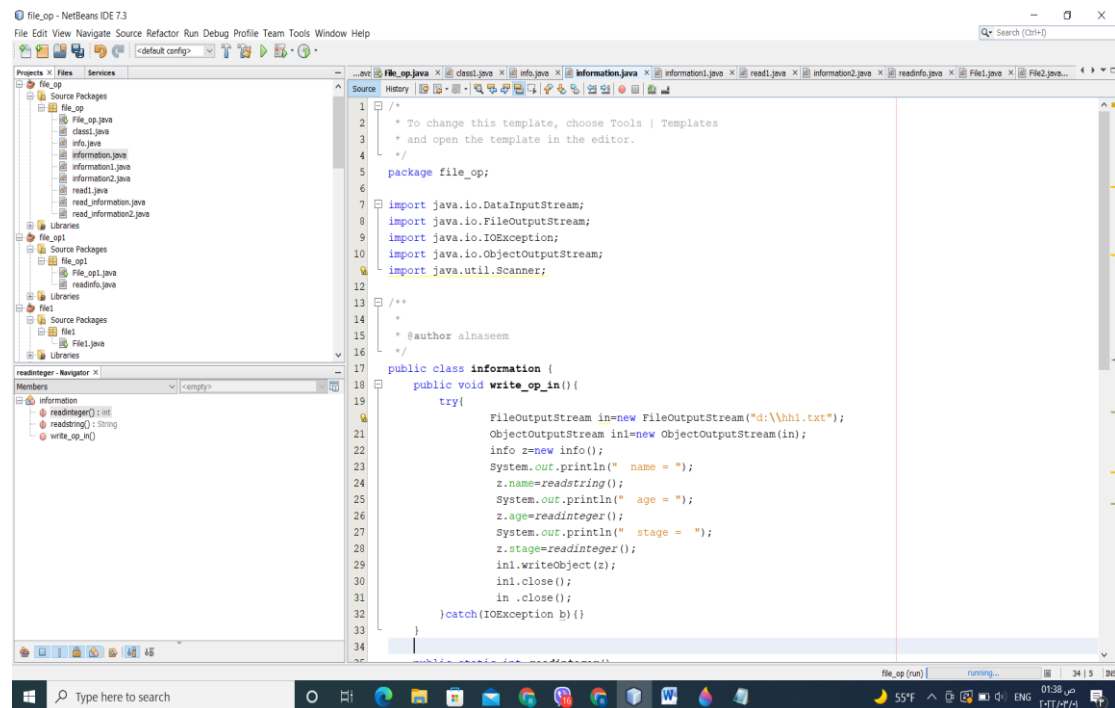
في هذا المثال نقوم بخرن بيانات طالب (اسم طالب ، عمر طالب، مرحلة) حيث يتم خزن هذه البيانات داخل فايل (Text Document) و ذلك باستخدام متغير واحد من نوع Object of class حيث يتم خلق sub class يتم تعريف بيانات طالب داخل sub class و يتم تضمينه في بيئة implements java.io.Serializable بعد اسم sub class.

تمثل صورة رقم (٤) خلق كلاس ثانوي sub class يتم تعريف البيانات المطلوبة داخل هذا الكلاس فقط (كلاس خاص بالتعريفات التي تمثل نوع الكائن) لتمثيل حالة طالب و هي اسم و عمر طالب و مرحلة. و من ثم كتابة دالة (implements java.io.Serializable) يكون بين اسم الكلاس الثانوي و ({}).



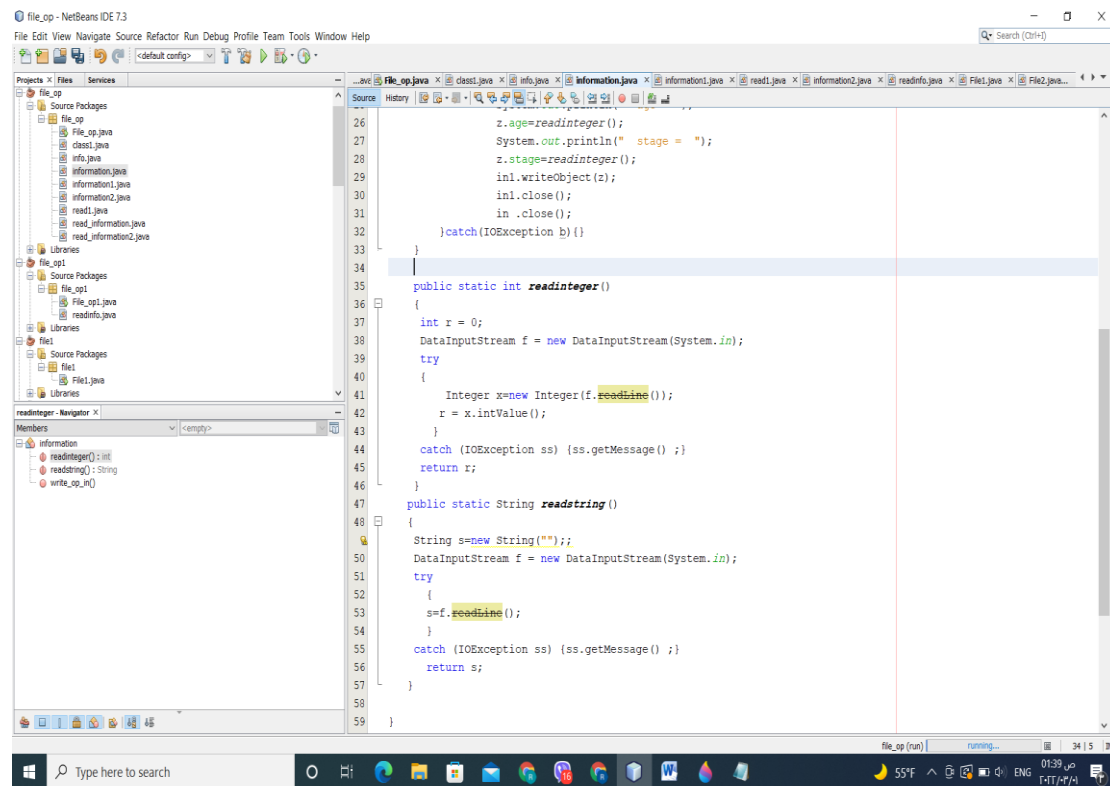
صورة (٤)

صورة رقم (٥) تمثل خلق sub class اخر لتضمين دالة الكتابة في الفايل باستخدام
FileOutputStream



صورة (٥)

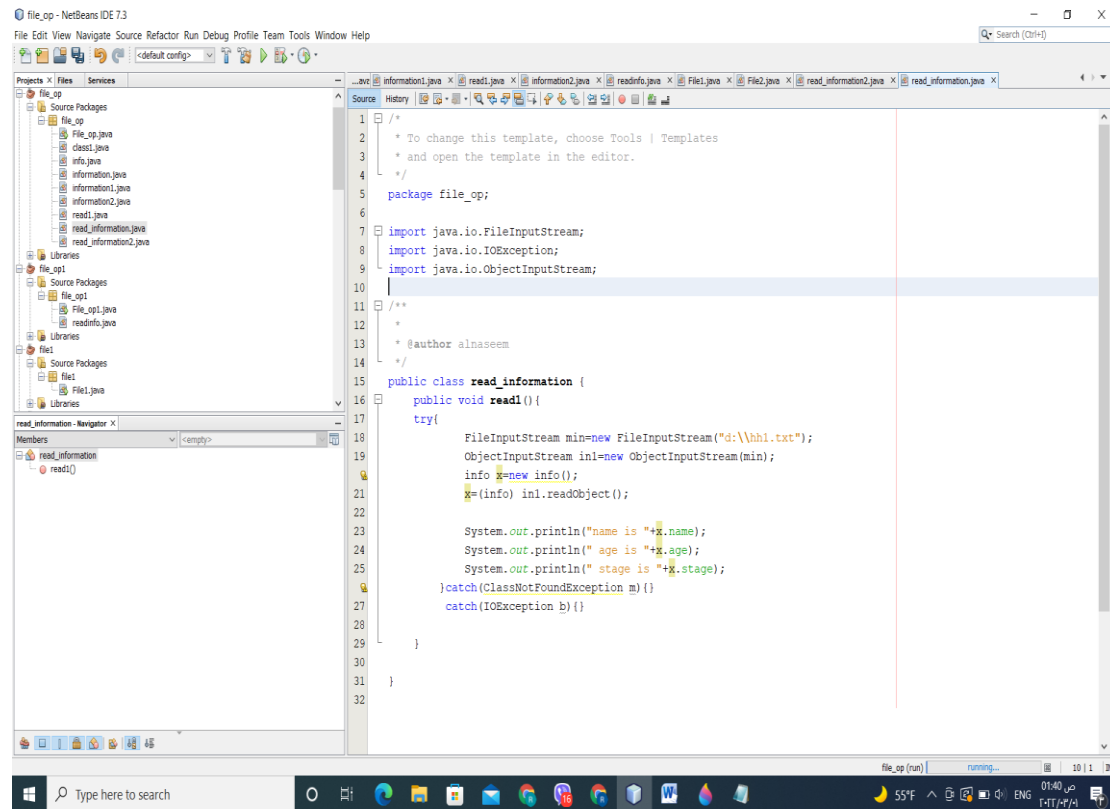
في هذا المثال صورة رقم (٦) تمثل دوال قراءة المتغيرات (اسم، عمر، مرحلة) من keyboard باستخدام دالة (readinteger) لقراءة متغير يمثل عدد صحيح، و دالة (readstring) لقراءة متغير يمثل سلسلة حروف.



```
26         z.age=readinteger();
27         System.out.println(" stage = ");
28         z.stage=readinteger();
29         in1.writeObject(z);
30         in1.close();
31         in.close();
32     }catch(IOException b){
33     }
34
35     public static int readinteger()
36     {
37         int r = 0;
38         DataInputStream f = new DataInputStream(System.in);
39         try
40         {
41             Integer x=new Integer(f.readLine());
42             r = x.intValue();
43         }
44         catch (IOException ss) {ss.getMessage(); }
45         return r;
46     }
47     public static String readstring ()
48     {
49         String s=new String("");
50         DataInputStream f = new DataInputStream(System.in);
51         try
52         {
53             s=f.readLine();
54         }
55         catch (IOException ss) {ss.getMessage(); }
56         return s;
57     }
58
59 }
```

صورة (٦)

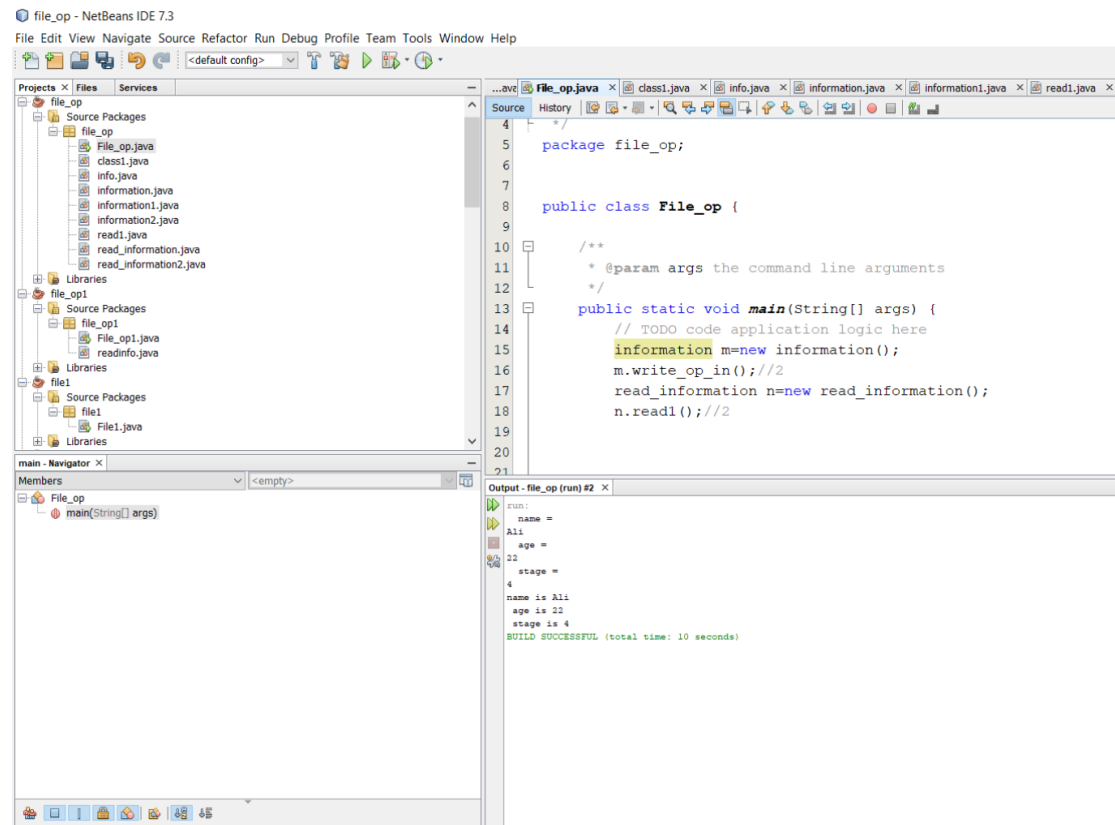
صورة رقم (٧) تمثل خلق sub class اخر لتضمين دالة قراءة بيانات من فايل.



```
1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package file_op;
6
7  import java.io.FileInputSteam;
8  import java.io.IOException;
9  import java.io.ObjectInputSteam;
10
11  /**
12   *
13   * @author alnaseem
14   */
15  public class read_information {
16      public void read() {
17          try{
18              FileInputSteam min=new FileInputSteam("d:\\hh1.txt");
19              ObjectInputSteam inl=new ObjectInputSteam(min);
20              info x=new info();
21              x=(info) inl.readObject();
22
23              System.out.println("name is "+x.name);
24              System.out.println("age is "+x.age);
25              System.out.println(" stage is "+x.stage);
26          }catch(ClassNotFoundException m) {}
27          catch(IOException b) {}
28
29      }
30
31  }
32
```

صورة (٧)

صورة رقم (٨) تمثل main class حيث يتم فيه فقط استدعاء دالة كتابة بيانات في فايل و دالة قراءة بيانات من فايل، و شاشة تنفيذ للاسفل توضح حالة ادخال بيانات في حالة كتابة و طباعة بيانات بشاشة تنفيذ جافا عند قراءة.



صورة (٨)

