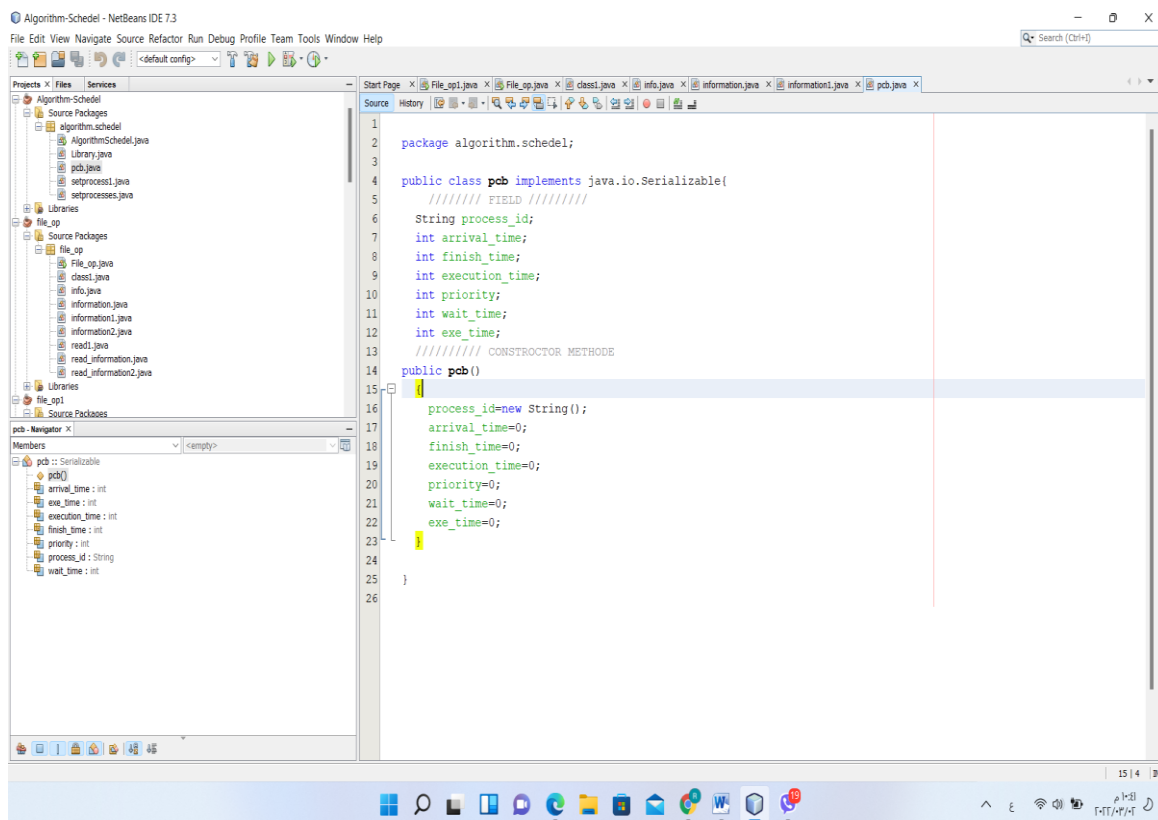


## Scheduling

تتناول هذه المحاضرة مقدمة لمشروع جدولة العمليات حيث يتم من خلالها تعريف المتغيرات الخاصة بكل عملية وادخال قيم المتغيرات للعملية وتخزينها في الفايل ومن ثم سحب هذه المتغيرات من الفايل من اجل تطبيق احد عمليات الجدولة عليها.

في هذه المحاضرة سوف نقوم بتعريف المتغيرات التي نحتاجها لكل عملية في (PCB sub class1) و اعطاء هذه المتغيرات قيم ابتدائية في كلاس (كلاس التعاريف). ثم نقوم بخلق () sub class2 من اجل قراءة و خزن قيم متغيرات لكل عملية، ومن ثم نخلق () sub class3 من اجل سحب البيانات من الفايل و تطبيق عمليات الجدولة عليها.



```

1 package algorithm.schedel;
2
3
4 public class pcb implements java.io.Serializable{
5     ////////// FIELD //////////
6     String process_id;
7     int arrival_time;
8     int finish_time;
9     int execution_time;
10    int priority;
11    int wait_time;
12    int exe_time;
13    ////////// CONSTRUCTOR METHODE
14    public pcb()
15    {
16        process_id=new String();
17        arrival_time=0;
18        finish_time=0;
19        execution_time=0;
20        priority=0;
21        wait_time=0;
22        exe_time=0;
23    }
24
25
26

```

صورة (1)

تمثل صورة رقم (1) كلاس pcb حيث تم من خلاله التصريح عن المتغيرات واعطاءها قيم ابتدائية.

```

1 package algorithm.schedel;
2 import java.io.DataInputStream;
3 import java.io.FileOutputStream;
4 import java.io.IOException;
5 import java.io.ObjectOutputStream;
6 public class setprocesses {
7     public static final int n=4;////////// input process into file //////////
8     public void read()
9     {
10        System.out.println("input " +n+ " process ");
11    }
12    FileOutputStream fos=new FileOutputStream ("E:\\java\\in.txt");
13    ObjectOutputStream oos=new ObjectOutputStream (fos);
14    for(int i=0;i<n;i++)
15    {
16        pcb a=new pcb();
17        System.out.println("process id ");
18        a.process_id = readString();
19        System.out.println("arrival time");
20        a.arrival_time = readInteger();
21        System.out.println("execution time");
22        a.execution_time = readInteger();
23        System.out.println("priority");
24        a.priority = readInteger();
25        a.new_time = execution_time;
26        oos.writeObject(a);
27    }
28    fos.close();
29    fos.close();
30    }catch (IOException ex) {ex.getMessage();}
31    }
32    public static int readInteger()
33    {
34        int r = 0;
35        DataInputStream f = new DataInputStream(System.in);
36        try
37        {
38            Integer n = new Integer(f.readLine());
39            r = n.intValue();
40        }
41        catch (IOException ex) {ex.getMessage();}
42        return r;
43    }
44    public static String readString()
45    {
46        String s=new String("");
47        DataInputStream f = new DataInputStream(System.in);
48        try
49        {
50            s=f.readLine();
51        }
52        catch (IOException ex) {ex.getMessage();}
53        return s;
54    }
55 }

```

صورة (٢)

تمثل الصورة رقم ( ٢ ) كلاس setprocess الذي يتضمن عملية ادخال اسماء العمليات و متعلقاتها وتخزينها في الفايل.

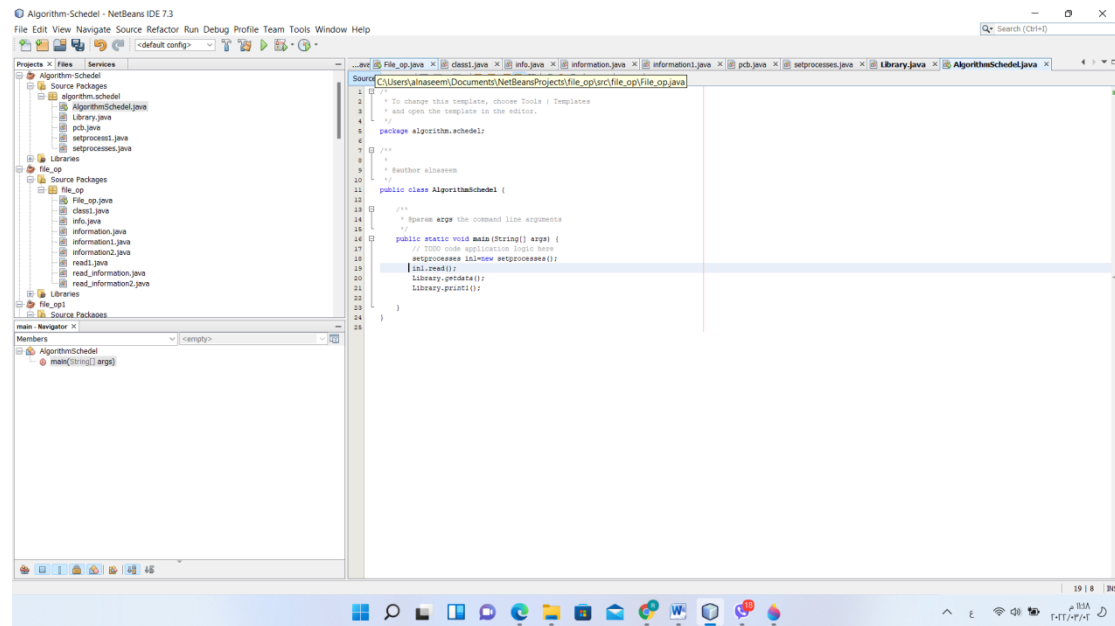
```

1 package algorithm.schedel;
2
3
4 import java.io.File;
5 import java.io.FileInputStream;
6 import java.io.IOException;
7 import java.io.ObjectInputStream;
8
9 public class Library {
10     static pcb Wait_Queue[] = new pcb[100];
11     static pcb Ready_Queue[] = new pcb[100];
12     static pcb Finish_Queue[] = new pcb[100];
13     public static int Wait_Length = setprocesses.n;
14     public static int Ready_Length = 0;
15     public static int Finish_Length=0;
16     public static int i=0;
17     //////////////// METHODES ////////////////
18     //////////////// GET PROCESS FROM FILE ////////////////
19     public static void getdata()
20     {
21         try {
22             File f = new File("E:\\java1.txt");
23             FileInputStream fis = new FileInputStream(f);
24             ObjectInputStream ois = new ObjectInputStream(fis);
25             for (int i = 0; i < setprocesses.n; i++) {
26                 pcb m = new pcb();
27                 x = (pcb) ois.readObject();
28                 Wait_Queue[i] = (pcb) x;
29             }
30             ois.close();
31             fis.close();
32         }
33         catch (ClassNotFoundException e) { e.printStackTrace(); }
34         catch (IOException ex) { ex.printStackTrace(); }
35     }
36     public static void print()
37     {
38         System.out.println("Process id      "+"Arrival time      "+"Execution time      "+"Finish time      "+"Wait time      ");
39         System.out.println("*****      *****      *****      *****");
40         for (int i = 0; i < setprocesses.n; i++)
41         {
42             System.out.println(Wait_Queue[i].process_id +
43                 "      "+"Wait_Queue[i].arrival_time+
44                 "      "+"Wait_Queue[i].priority+
45                 "      "+"      ");
46         }
47     }
48     //////////////// Print Finish Queue
49     public static void print()

```

### صورة (٣)

تمثل صورة رقم ( ٣ ) كلاس library الذي تم من خلاله سحب البيانات من الفايل لغرض المعالجه مستقبلا باحد عمليات الجدولة. ومن خلال هذا الكلاس سيتم تضمين كافة عمليات الجدولة. اما main class يمثل الكلاس الرئيسي و فيه يتم استدعاء الدوال لغرض تنفيذ (صورة رقم ٤).



صورة (٤)