République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
ECOLE SUPÉRIEURE EN INFORMATIQUE
8 Mai 1945 - Sidi-Bel-Abbès

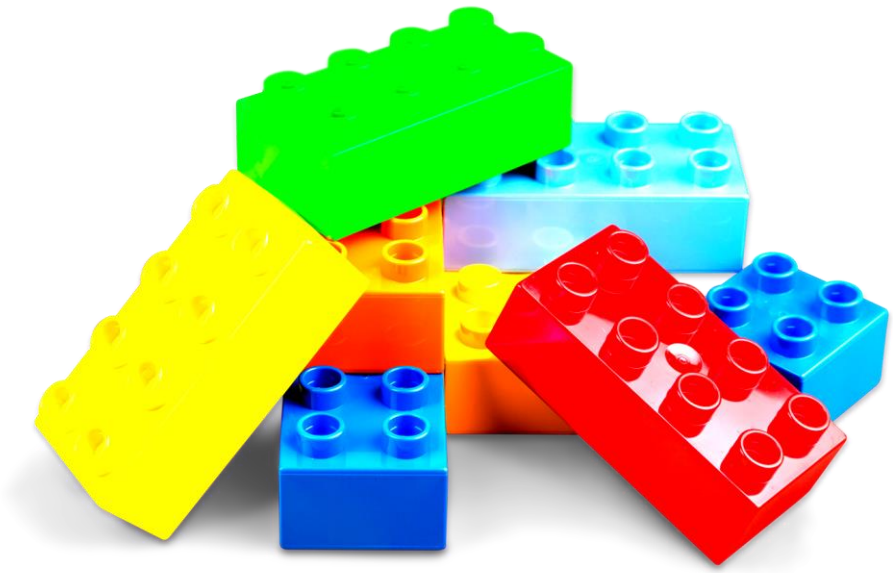
الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
المدرسة العليا للإعلام الآلي
8 ماي 1945 - سيدي بلعباس

Object-Oriented Programming (OOP)

2nd Year CPI

Mini Project Guide

Date: 03-30-2023



Output: see video demonstration ➔ https://youtu.be/WH8ps6V_avA

or make YouTube search for: OOP Mini project Demo

Deadline: before **08-06-2023** (First coming high mark)

Dr, SERHANE Oussama

- **Mini Project objectives :**
  1- Be familiar with creating/managing a java se graphical interface
  2- Applying some oriented object concepts
  3- Acquire best practices of desktop-based application
  4- Get experience with NetBeans IDE
  5- Debugging and error handling
  6- Build something …

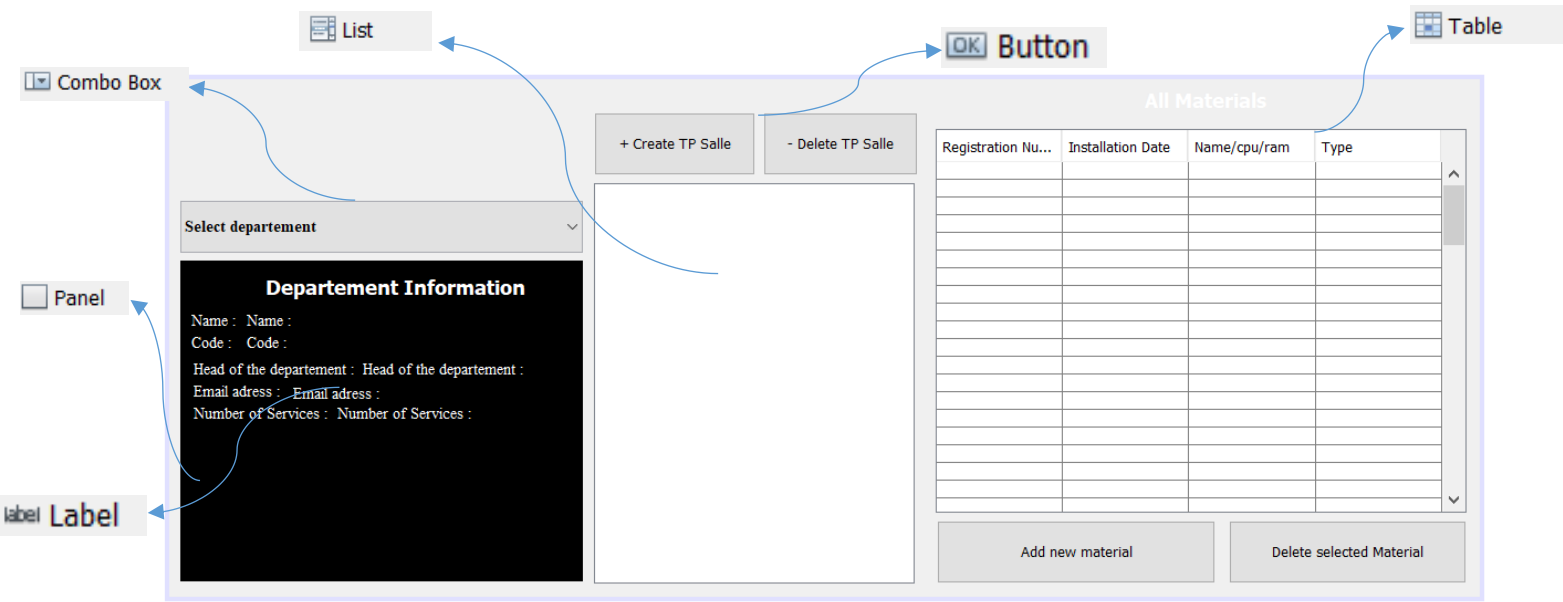
- **Recommandations :**
  - *Each student must have its own version of this mini-project*
  - *The consultation will be at the TP sessions*
  - *Students that won't accomplish this mini-project will get **0/20**. Which may affect their TP mark.*
  - *Students that accomplish all this mini-project sooner and get the full mark will have a bonus that might be +1 or +2 on the exam.*
  - *I will answer any kind of question-related to this during the lecture sessions.*

Part One: UI Design

Small Descriptions

Task 01: using NetBeans IDE, create a new Frame, lets call it "Main_Home", then clone the following design according the mentioned components.

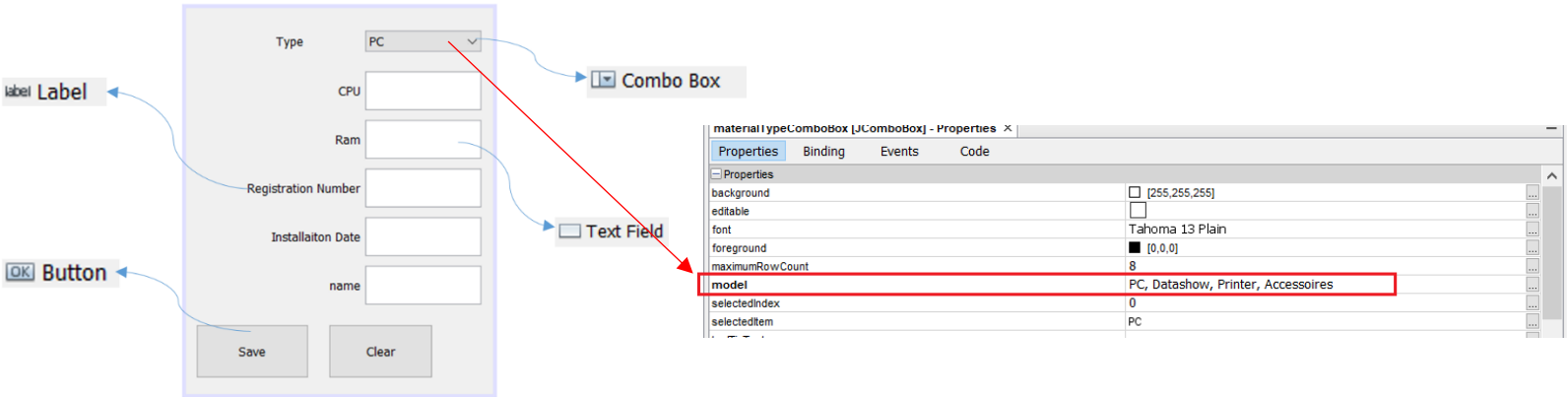
Hint: to simplify your design on the mainframe of your design write click and then set layout to absolute layout



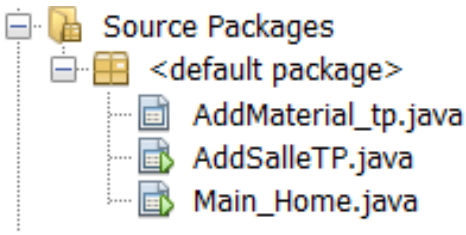
Task 02: Create an other Frame class call it "AddSalleTP" the will allow us to receive the information from a user and create a new TP Salle with their information.

Task 03: Similarly, create a new Frame class call it "AddMaterial_tp" the will allow us to receive the information from a user and create a new materials, those materials could be either pc, datashow, printers, or accessories.

Note: Edit the properties of the Combo Box set the default values for the model as following

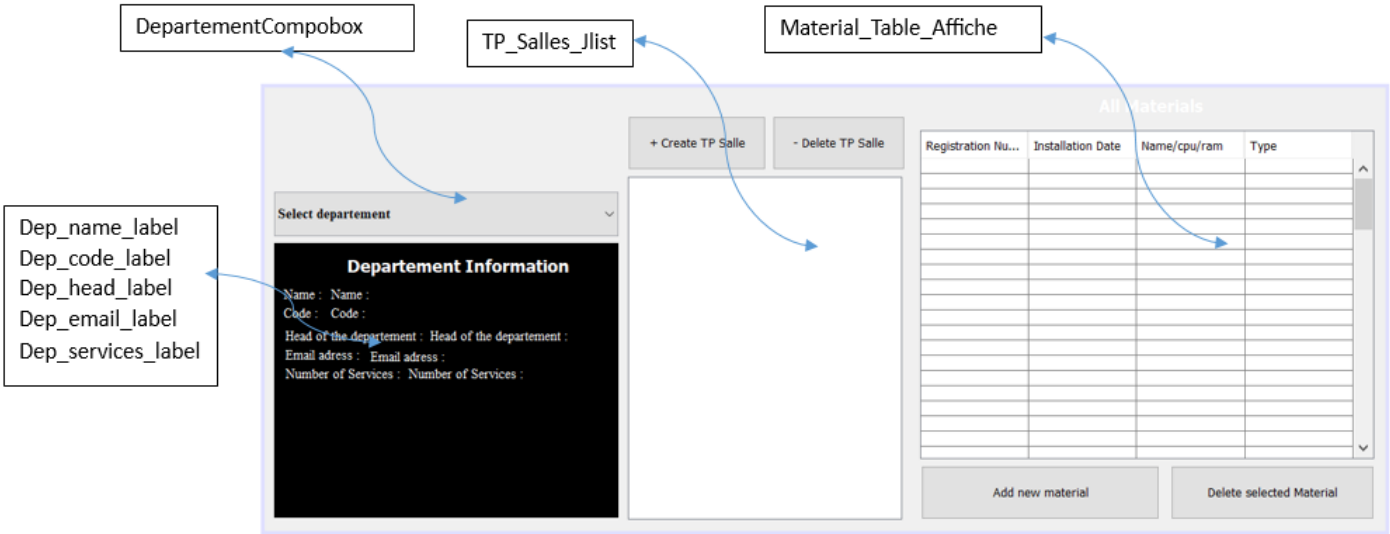


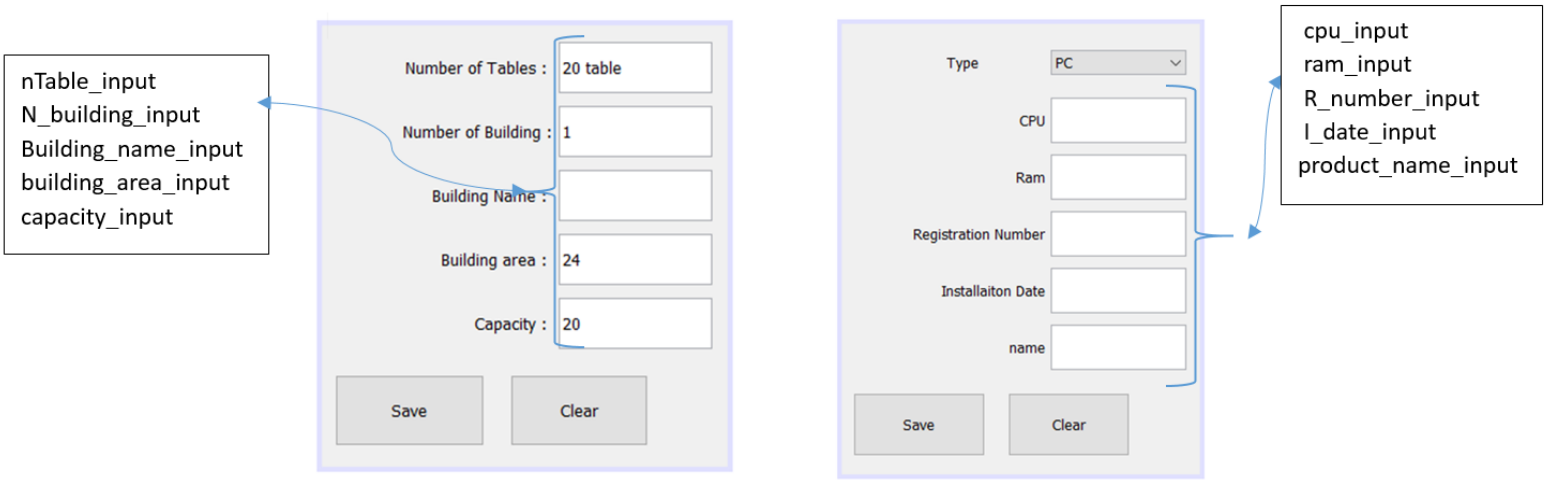
Final result UI files: The final output files of this part will be the following three classes



The next step, after UI design, is to set a variable name for each component to ease the manipulation. The following is the list of variable assignments for each component in our UI. Components that are not mentioned are set to the default variable names choose by IDE, for instance, jButton1, jButton2 ...etc.

Note: This is a bonus that will ease the code integration later.




nTable_input
N_building_input
Building_name_input
building_area_input
capacity_input
Number of Tables :
20 table
Number of Building :
1
Building Name :
Building area :
24
Capacity :
20
Save
Clear
Type
PC
CPU
Ram
Registration Number
Installaiton Date
name
Save
Clear
cpu_input
ram_input
R_number_input
I_date_input
product_name_input

Part Two: Coding (Read and build)

In this part, we will integrate the logic functionality to each components. We will reuse the code source of already implemented classes of TP 05 and TP06 part two.

Main_Home.java

The following capture shows the detail code structure of class Main_home and their methods including its constructor (line 89).

```
6   public class Main_Home extends javax.swing.JFrame {
7
8       public static Institution Esi = null;      // variable of Type Institution to store the hard coded Institution
9       public static ArrayList<TP_Salle> salles = new ArrayList<>();  // List of salles to store the hard coded TP salles
10      int Current_Salle_TP_Index;  // an integer variable to store selected TP salle index from ui
11
12      public static void Institution_init() {...25 lines }
37
38      public static void Tp_salles_init() {...18 lines }
56
57      public void load_Departements() {...5 lines }
62
63      public static void load_Salles() {...8 lines }
71
72      public static void print_salle_materials(ArrayList<Materials> tp_materials) {...16 lines }
88
89      public Main_Home() {...8 lines }
```

Task 01 : Read the following methods description then complete the code integration methods by methods, whenever adding a new method code it is highly recommended to perform a partial test.

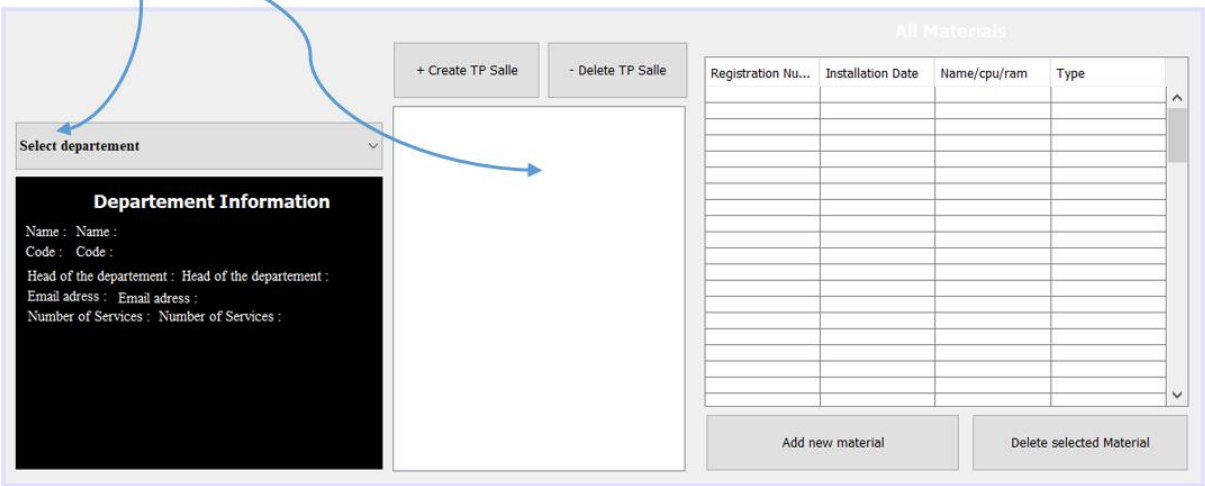
Line Number	Function	Description
12	**Institution_init()**	This method is used to create a dummy data institution with two departments, buildings and store all the result in already declared varible of type institution called Esi (line number 8). The objective of this code is to simplify the process.
38	**Tp_salles_init()**	This method is used to generate a dummy data of TP salles with their hard coded materials of different types. The objective of this code is to simplify the process of testing. By running this method it will fill in a ArrayList of tp salles (line 9).
57	**load_Departements()**	The aim of this method is to read department information stored in variable called Esi then add their name to their ComboBox.
63	**load_Salles()**	We used this method to iterate through the salles ArrayList and then print the names to JList UI.
72	**print_salle_materials**(List of tp_materials)	We used this method to iterate through pre passed a parameter list of materials and print them to out table on the Main_Home ui.
89	**Main_Home()**	This is the constructor, we invoke the required methods that we need to execute them during the first load of the UI.

Important: some methods are static. We declare them as static to invoke them from the other classes or UI.
It is worth mentioning that a static method should contain only a static variable including UI containers or pallets. The following video shows how to set a container to a static.

Youtube Title: Java-SE Netbeans set any container to static
Link: https://youtu.be/1FhMdu8oND4

Task 02: In the Main_Home constructor, we call methods to run during the first execution or our software. As following:

```
86  public Main_Home() {
87      initComponents(); // Auto generate by NetBeans IDE
        this.setLocationRelativeTo(null); // to show the frame on the middle of the screen
89      Institution_init(); // To initialize a hard-coded institution and department
        load_Departements(); // To display the departement names
91      Tp_salles_init(); // To initialize a dummy Tp salle 01 with its materials
92      load_Salles(); // To Display the existing (dummy salles) to their JList
93  }
```



AddSalleTP.java

Task 03: Clone the following steps:

To create a new TP salle we click on the create button. The button should implement an event Action type, or onMousClick. This process will call the constructor of AddSalleTP.java class. And method of setVisubale(True) as following.

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    new AddSalleTP().setVisible(true);
}
```

AddSalleTP frame will appear, the user will tape the salle TP information. Whenever he click Save Button three steps will be performed:

1- We frst collect the information of the inputs
2- Create a new object of TP salle and push it the global ArrayList called salles (line 8 in Main_Home.java
3- Invoke Main_Home.load_Salles(); method to refresh the salle list display

```
private void saveTPSalleButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // Step 01: collect information inputs
    String nTable = nTable_input.getText();
    int N_building = Integer.parseInt(N_building_input.getText());
    String Building_name = Building_name_input.getText();
    double building_area = Double.parseDouble(building_area_input.getText());
    int capacity =  Integer.parseInt(capacity_input.getText());
    // Step 02: create a new TP salle object
    TP_Salle Salle_TP  =  new TP_Salle(nTable, N_building, Building_name, building_area, capacity);
    Main_Home.salles.add(Salle_TP);
    // Step 01: invoke this method will reflresh the salle list display
    Main_Home.load_Salles();
}
```

Task 04: This task is split in two folds:

1- Before implementing create a new material button logic, we must first know the selected TP salle by the user. To do so, whenever a user click on salle (select a salle) from the list we will get its index value and store it in variable Current_Salle_TP_Index (Main_Home.java line 8) as following:

```
private void TP_Salles_JlistMouseClicked(java.awt.event.MouseEvent evt) {
    int selectedTp_salle = -1;
    selectedTp_salle = TP_Salles_Jlist.getSelectedIndex();
    this.Current_Salle_TP_Index = selectedTp_salle;
```

2- Display already saved materials of this TP Salle in the JTable of Main_Home, by calling the print_salle_materials() method and pass tp_materials list as paramaitre

```
    if (selectedTp_salle != -1) {
        // call printting function
        ArrayList<Materials> tp_materials = salles.get(selectedTp_salle).getTp_materials();
        print_salle_materials(tp_materials);
    }
}
```

AddMaterial.java

Task 05: To create a new materials we click on add material button. The button should implement an event Action type, or onMousClick. This process will invoke the constructor of AddMaterial_tp.java class. And method of setVisubale(True) as following.

new AddMaterial_tp(constructor parameters).setVisible(true);

However, to create a new material list we should know beforehand the selected salle that a user wants to add material too. Therefore, we pass Current_Salle_TP_Index as a parameter to the constructor of AddMaterial_tp class.

Todo:

1- **Edit AddMaterial_tp class by adding a new integer variable that will be initialized during the constructor call.**
2- **To save a material in Main_Home.salle list we need following code whenever save button clicked:**

```java
private void saveMaterials_BtnActionPerformed(java.awt.event.ActionEvent evt) {
    int index = materialTypeComboBox.getSelectedIndex();
    // collect information inputs
    String cpu = cpu_input.getText();
    String ram = ram_input.getText();
    String R_number = R_number_input.getText();
    String Instl_date = I_date_input.getText();
    String product_name = product_name_input.getText();
    Materials mat = null;
     switch (index) {
        case 0:
            // call constractor PC collect pc infos
            System.out.println("hello");
            PC pc = new PC(cpu, ram, R_number, Instl_date);
             Main_Home.salles.get(Current_Salle_TP_Index).add_material_to_TPSalle(pc);
            break;
        case 1:
            // call Datashow constractor
            Datashow datashow = new Datashow(product_name, R_number, Instl_date);
            Main_Home.salles.get(Current_Salle_TP_Index).add_material_to_TPSalle(datashow);
            break;
        case 2:
            // call Printer constractor
            Printer printer = new Printer(R_number, Instl_date);
            Main_Home.salles.get(Current_Salle_TP_Index).add_material_to_TPSalle(printer);
            break;
        case 3:
            // call accessoire constractor
            Accessoires accessoires = new Accessoires(product_name, R_number, Instl_date);
            Main_Home.salles.get(Current_Salle_TP_Index).add_material_to_TPSalle(accessoires);
            break;
    }
    Main_Home.print_salle_materials(Main_Home.salles.get(Current_Salle_TP_Index).getTp_materials());
}
```

Hint: remove the <u>main</u> function of AddMaterial_tp class to avoid the error that may occurs because of calling the past (no args) constructor.

Important: Since you have reached this page you already notice that this not a from scratch coding. It is a simple code assembling and even most of the source code has been given in the assets file. Therefore, student that succeeds to accomplish all the tasks with success before the deadline gets 15/20. The rest of 3/20 will be reserved for the additional improvement and 2 for software quality.

Deadlines/Marks details will distributed as following:

Deadline	Mark/20
30-04-2023 ➔ 04-05-2023	15/20
07-05-2023 ➔ 11-05-2023	14/20
14-05-2023 ➔ 18-05-2023	13/20
21-05-2023 ➔ 25-05-2023	12/20
28-05-2023 ➔ 01-06-2023	11/20
04-06-2023 ➔ 08-06-2023	10/20
After 08-06-2023	00/20

<u>Increase your mark:</u> the rest 5 notes will be split as following

2 point for the qualities/beauty of the software (eg. No error during execution, creative UI improvement …)

<u>The following must be done in the order:</u>

1 point for implementing both of delete TP salle from list and delete material from table

1 point for link between department selection and their own tp salles

1 point for transforming all the project to self executed (.jar file)