OOPS PROJECT DOCUMENTATION

This project is about a car production factory where multiple cars are made on multiple assembly lines simultaneously. Each car in the assembly line is formed by basic parts. Different types of cars can be made by using different combinations of various basic parts. The time required to assemble different types of basic parts can vary.

DESCRIPTION OF CLASSES AND INTERFACES

1) Interface of each Basic part
   i) Frame interface
      - It has a display() method.
   ii) Wheel Interface
      - It has a display() method.
   iii) Engine Interface
      - It has a display() method.
   iv) Interior Interface
      - It has a display() method.
   v) Chasis Interface
      - It has a display() method.
   vi) Color Interface
      - It has a display() method.
2) Classes of each part
   1) Frame1, Frame2, Frame3, Frame4, Frame5 classes implements Frame Interface.
   2) Wheel1, wheel2, wheel3 classes implements wheel interface
   3) Engine1, Engine2, Engine3, Engine4, Engine5, Engine6, Engine7, Engine8 classes implements Engine Interface.
   4) Interior1, Interior2, Interior3 classes implement Interior Interface.
   5) Chasis1, Chasis2, Chasis3, Chasis4, Chasis5, Chasis6, Chasis7 classes implements Chasis Interface.
   6) Color1, Color2, Color3, Color4, Color5, Color6, Color7, Color8, Color9, Color10, Color11, Color12 classes implements Color Interface.

      - Total 38 classes has been used which contains an overridden display() method which makes the thread sleep for particular period(which is the time required for a particular basic part to get assemble) and after that , prints the required output with the time at which that part is fitted.

The display method takes the parameter int n which provides us the information of the assembly line's number where the basic part is fitted.

3) CarInterface Interface
    - This interface contains :- frame(),wheel(),engine(),interior(),chasis(),color() methods with required parameters. The parameter is int n and int m, where n represents the assembly line's Number and m represents the type of basic part fitted.

4) AssemblyLine Class

The class extends the thread class and implements the carInterface.

- Contains n as an instance variable of dataType int which provide the information of the AssemblyLine's number.
- Contains start1 variable which is static and its data type is long. It takes a random value of time when initialized. It is initialized with the function `System.nanoTime() which is in the package  java.time.Duration.`
- Contains num variable of dataType int which provides the no of assembly line running in the factory (which is a user input.).The constructor of AssemblyLine class will initialize the num variable as soon as the object is made.
- Contains a static array arr1 of data type int which helps to store the type of basic parts (like 1 for Frame1, 2 for Wheel2 etc.) that the user has inputted.
- It contains overridden methods of frame(), wheel(), engine(), interior(), chasis(), color() with the parameters same as the ones used in carInterface interface.

Frame() method
This methods have various (if-else if) statements, and depending upon the value of the variable m, the reference of interface Frame is made and the display method is called.
Wheel() method
This methods have various (if-else if) statements, and depending upon the value of the variable m, the reference of interface Wheel is made and the display method is called.
Engine() method
This methods have various (if-else if) statements ,and depending upon the value of the variable m, the reference of interface Engine is made and the display method is called.
Interior() method
This methods have various (if-else if) statements ,and depending upon the value of the variable m, the reference of interface Interior is made and the display method is called.
Chasis() method

This methods have various (if-else if) statements, and depending upon the value of the variable m, the reference of interface Chasis is made and the display method is called.

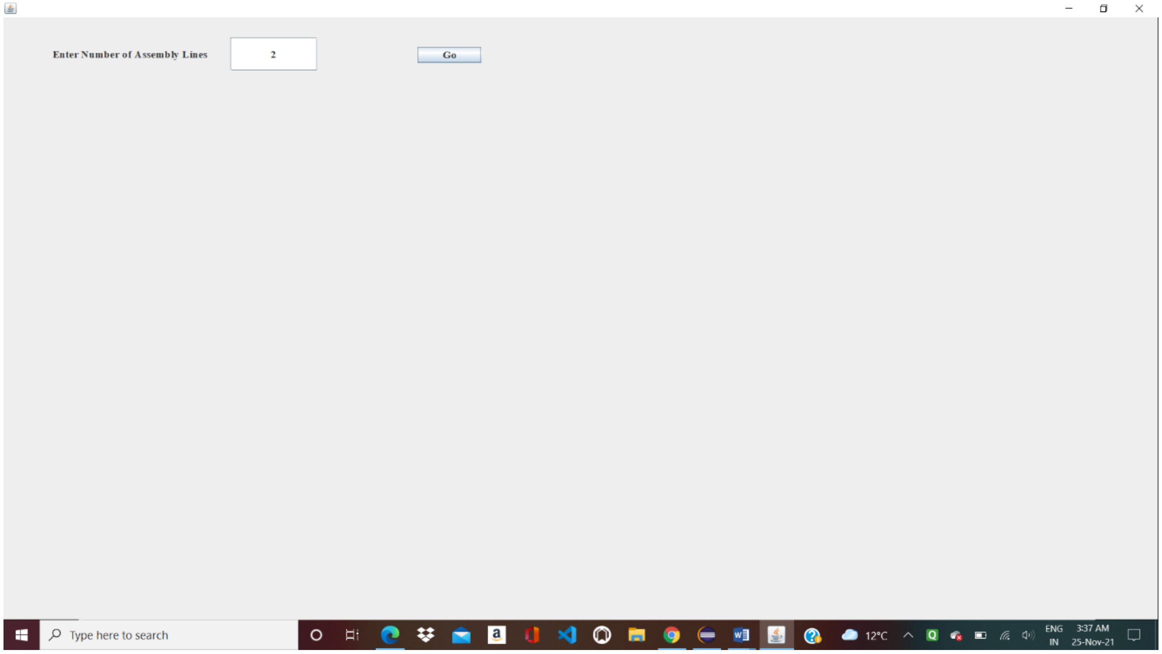
Color() method

This methods have various (if-else if) statement, and depending upon the value of the variable m, the reference of interface Color is made and the display method is called.

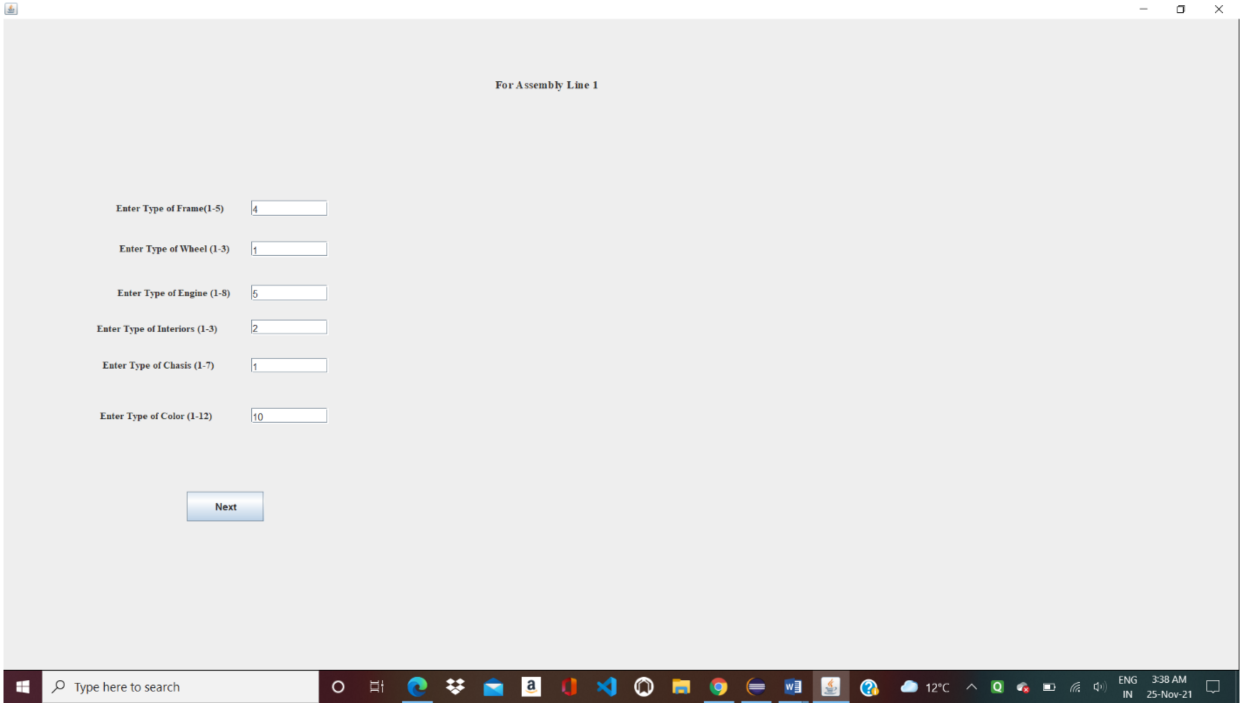
- Contains an overridden run() method from the thread class .It contains a for loop in it and in that for loop, it has an if ladder. It basically calls the function mentioned in the previous point depending on the value of i.
- The main function is also in the AssemblyLine class. It has the object of Car_production_assignment where we have implemented the gui framework.

INPUTS AND OUTPUTS

Input:



Enter Number of Assembly Lines 2



For Assembly Line 1

Enter Type of Frame(1-5) 4

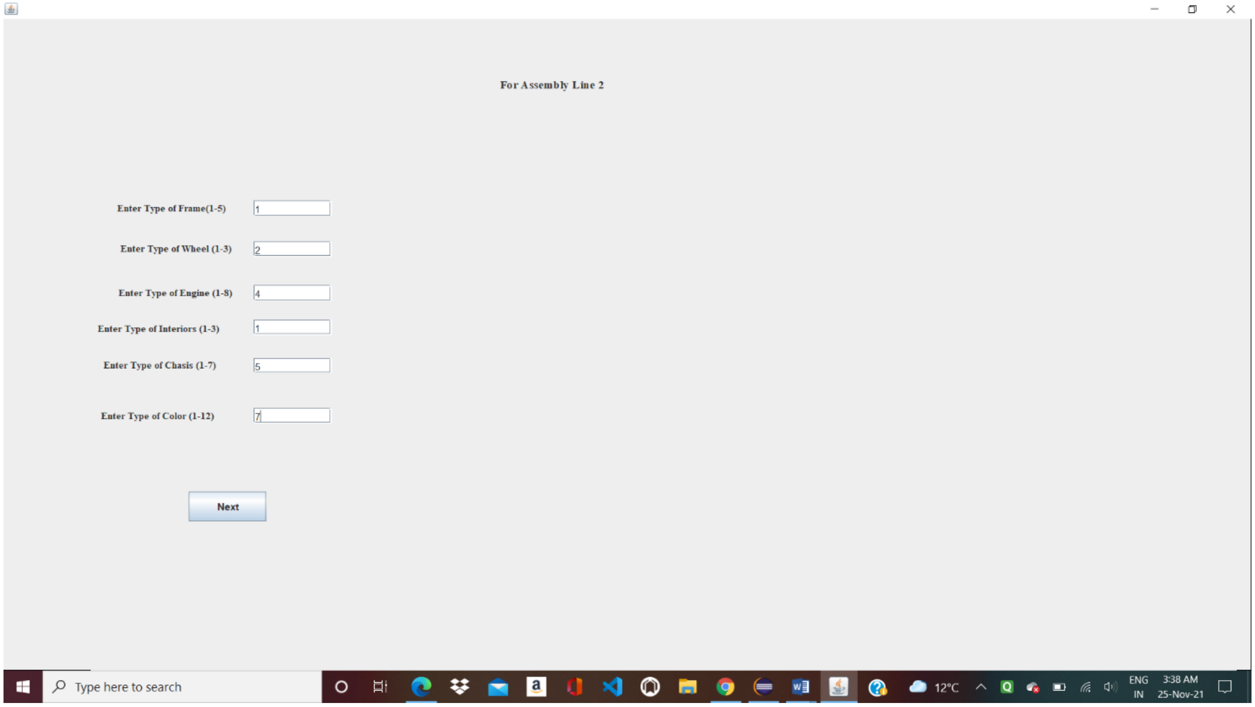
Enter Type of Wheel (1-3) 1

Enter Type of Engine (1-8) 5

Enter Type of Interiors (1-3) 2

Enter Type of Chasis (1-7) 1

Enter Type of Color (1-12) 10



For Assembly Line 2

Enter Type of Frame(1-5) 1

Enter Type of Wheel (1-3) 2

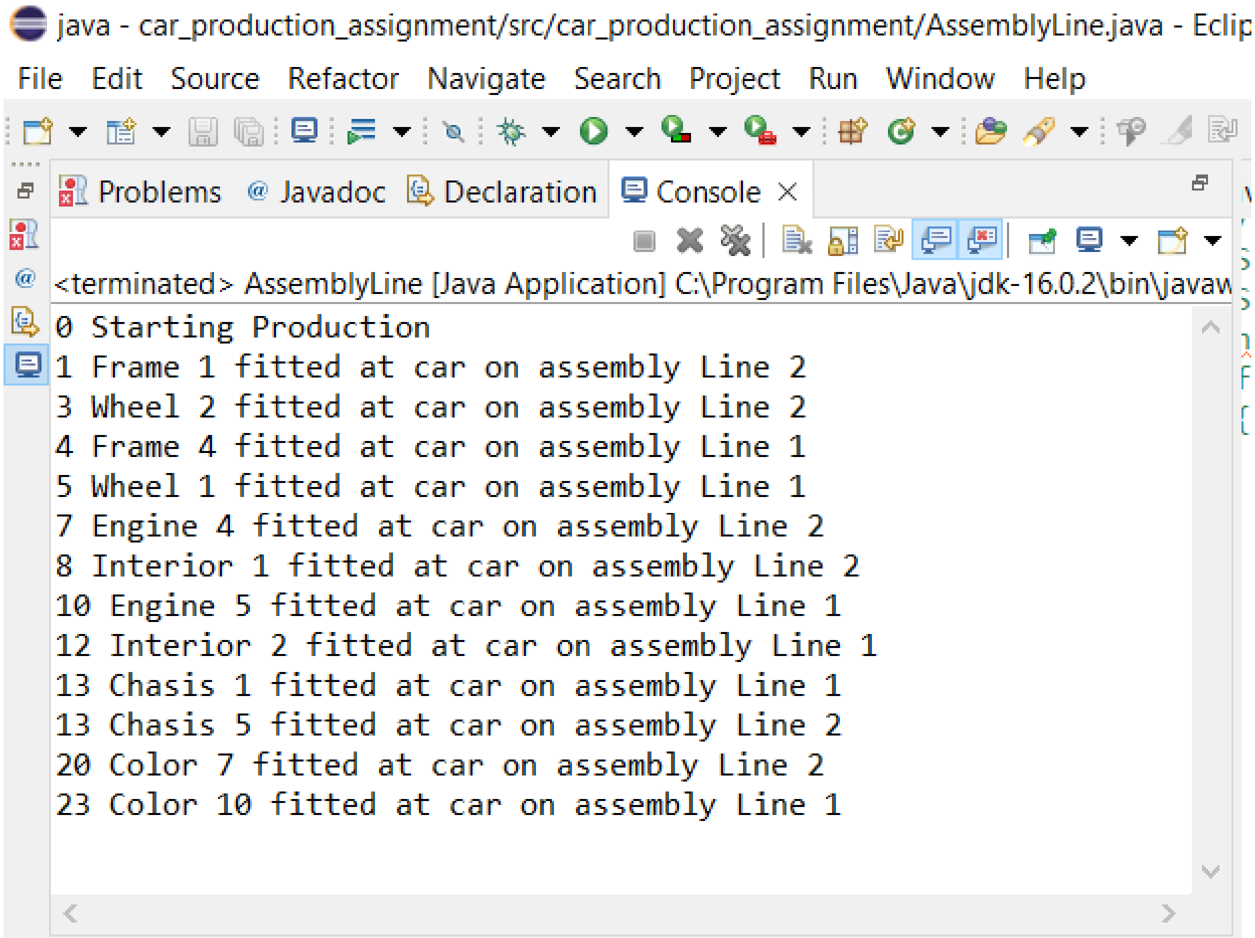
Enter Type of Engine (1-8) 4

Enter Type of Interiors (1-3) 1

Enter Type of Chasis (1-7) 5

Enter Type of Color (1-12) 7

Output:


java - car_production_assignment/src/car_production_assignment/AssemblyLine.java - Eclip
File Edit Source Refactor Navigate Search Project Run Window Help
Problems
Javadoc
Declaration
Console

```
<terminated> AssemblyLine [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw
0 Starting Production
1 Frame 1 fitted at car on assembly Line 2
3 Wheel 2 fitted at car on assembly Line 2
4 Frame 4 fitted at car on assembly Line 1
5 Wheel 1 fitted at car on assembly Line 1
7 Engine 4 fitted at car on assembly Line 2
8 Interior 1 fitted at car on assembly Line 2
10 Engine 5 fitted at car on assembly Line 1
12 Interior 2 fitted at car on assembly Line 1
13 Chasis 1 fitted at car on assembly Line 1
13 Chasis 5 fitted at car on assembly Line 2
20 Color 7 fitted at car on assembly Line 2
23 Color 10 fitted at car on assembly Line 1
```

Our Assumption

Time taken for Frame1 to be fit-1sec

Time taken for Frame4 to be fit-4sec

Time taken for Wheel1 to be fit-1sec

Time taken for Wheel2 to be fit-2sec

Time taken for Engine4 to be fit-4sec

Time taken for Engine5 to be fit-5sec

Time taken for Interior1 to be fit-1sec

Time taken for Interior2 to be fit-2sec

Time taken for Chasis1 to be fit-1sec

Time taken for Chasis5 to be fit-5sec

Time taken for Color7 to be fit-7sec

Time taken for Color10 to be fit-10sec

Explaining the Sample Output

Every basic component is given a specific time by us to get fitted in the car .Thus that statement will also take the same time to get printed.

(In each statement of the output ,the time taken is printed first and the statement is printed next)

The frame1 of assembly Line 2 takes 1 sec to get fitted and hence the 2st statement will be printed after 1 sec.

The wheel2 of assembly Line 2 takes 2 sec to get fitted and hence the 3rd statement in the output is printed after (2+1)=3sec.(1 is added because already the assembly line 2 has started and frame1 has been fitted which took 1 sec).

The frame4 of assembly Line 1 takes 4 sec to get fitted and hence the 4st statement will be printed after 4 sec.

The wheel 1 of assembly Line 1 takes 1 sec to get fitted and hence the 5th statement will be printed after (4+1) =5sec.

The Engine 4 of assembly Line 2 takes 4 sec to get fitted and hence the 6th statement will be printed after(2+1+4)=7sec.

The Interior1 of assemble Line 2 takes 1 sec to get fitted and hence the 7th statement will be printed after(2+1+4+1)=8sec.

The Engine5 of assembly Line 1 takes 5sec to get fitted and hence the 8th statement will be printed after (4+1+5)=10sec.

The Interior 2 of assembly Line 1 takes 2sec to get fitted and hence the 9th statement will be printed after(4+1+5+2)=12sec.

The Chasis 1 of assembly Line 1 takes 1sec to get fitted and hence the 10th statement will be printed after (4+1+5+2+1)=13sec.

The Chasis5 of assembly line 2 takes 5sec to get fitted and hence the 11th statement will be printed after(2+1+4+1+5)=13sec.

(note that both takes the same time to print ideally but there will be some difference in the values at their decimal places and hence due to this minute difference ,10th statement is printed 1st and then 11th is printed.).

The Color7 of assembly line 2 takes 7 sec to get fitted and hence the 12th statement will be printed after (2+1+4+1+5+7) =20sec.

The Color10 of assembly Line1 takes 10sec to get fitted and hene the 13th statement will be printed after (4+1+5+2+1+10)=23sec.

Thus the program will get terminate after 23secs.

EXPLAINING THE CODE AS A WHOLE

OUR GUI FRAMEWORK EXPLANATION

So the code starts with the main function which is in the AssemblyLine class. The window object of Car_production_assignment is formed which is meant for taking the input from the user using GUI Framework. The constructor of the Car_production_assignment calls the initialize() method. The initialize() method has the definitions of the icons and buttons that appear on the window when the program is running. It contains the JTextField(where the user input the no of assembly lines),JLabel(where `"Enter Number of Assembly Lines"` is written) and the JButton(named "Go").The JButton contains addActionListener method where the input from the user in the JTextField is stored in num(int type) variable. The num instance variable in assemblyLine class is initialized by num. (AssemblyLine.num=num).Then the frame is disposed by frame.dispose() function. Then the object of CarInfo is made and (1,num) parameter is passed in the constructor.

The CarInfo class contain a static variable n of type int which is initialized by the constructor. Also it contains static variable num of type int which is also initialized by the constructor. Then various methods are formed which designs various JLabel,JTextField and a JButton as per requirement. The JButton has the method `actionPerformed()` which contains if-elseif statement.

If n is less than num then all the inputs are stored in the 2D array arr1 and then the window is disposed and then another object of carInfo is made.

When n==num, then all the inputs are stored in the 2d array arr1 and then the window is disposed. The start1 variable which is declared in the AssemblyLine class is initialized by `System.nanoTime();` functions which takes a random time as input. Then we implement a for loop, in which object of assemblyLine class was formed. As soon as the object was formed ,the value i+1 was passed as a parameter and the instance variable n of assemblyLine was initialized. Then the threads were formed by cari.start() and the call reached the run() method of AssemblyLine class.

Run() method of AssemblyLine class

The run() method has a for loop which contains if-ladder. It will call the functions according to the value of i. The function has n and arr[n-1][i] passed as arguments.

If i is 0;

Then frame() function is called .The m variable in the argument has the value of arr[n-1][0] .Thus according to the value of m, the reference of frame Interface is created and then the display method is called.

Then i is increment to 1;

Then Wheel() function is called. The m variable in the argument has the value of arr[n-1][1] .Thus according to the value of m, the reference of wheel Interface is created and then the display method is called.

Then i is increment to 2;

Then engine() function is called. The m variable in the argument has the value of arr[n-1][2] .Thus according to the value of m, the reference of engine Interface is created and then the display method is called.

Then i is increment to 3;

Then interior() function is called. The m variable in the argument has the value of arr[n-1][3] .Thus according to the value of m, the reference of interior Interface created and then the display method is called.

Then i is increment to 4;

Then Chasis() function is called. The m variable in the argument has the value of arr[n-1][4] .Thus according to the value of m, the reference of Chasis Interface is created and then the display method is called.

Then i is increment to 5;

Then Color() function is called. The m variable in the argument has the value of arr[n-1][5] .Thus according to the value of m, the reference of Color Interface is created and then the display method is called.