

## Espressor Arhitecture Mini-Documentation

### !!! RULES FOR USERS:

1. All inputs are case-sensitive.
2. All steps are done in consecutive order. User cannot make coffee before inserting the coffee sort.
3. You can add water/coffee only **once** per session. You have to restart the espressor to add another value for water amount or for coffee sort.
4. In case you write a wrong value (for starting the machine, water amount, coffee sort), you can rewrite it, without restarting the espressor.
5. When the message ends with „ ... ” it means that the process is „dynamic” and you have to wait for a little bit.

### 0 START

After running the code, the inscription “To turn on the espressor write start” will appear. You have to enter “start”.

```
p\Esspresor_CSharp.csproj
To turn on the espressor write start
start
1. Add Water
2. Insert Coffee Sort
3. Make Coffee
4. Waiting regime
5. Exit
```

### 1 ADD WATER

Amount should be integer type  $\geq 150$  and  $\leq 750$ . On amount of water, depends the pressure in the boiler. If the amount is greater than 650 ml and the temperature is greater than 95 (which you cannot choose), the pressure will be resetted. Respective message will appear in the console. In other case, it will appear that the pressure is normal.

```
1
Enter the amount of water in ml (150 - 750):
780
The amount should be in range 150 - 750 ml. try again!!!
Enter the amount of water in ml (150 - 750):
160
The amount is good.
The pressure is normal
2. Insert Coffee Sort
3. Make Coffee
4. Waiting regime
5. Exit
```

## 2 INSERT COFFEE

You have 3 choices of coffee sorts “arabica, robusta, liberica” and you have to insert only one of them. Don’t forget that the inputs are case-sensitive ☺

```
2
Choose your coffee arabica, robusta, liberica
rbst
Inserting the coffee...
Invalid coffee sort. Please insert arabica liberica or robusta
2. Insert Coffee Sort
3. Make Coffee
4. Waiting regime
5. Exit
2
Choose your coffee arabica, robusta, liberica
robusta
Inserting the coffee...
robusta has been successfully insterted
Choose the next step Make Coffee.
```

## 3 MAKE COFFEE

If the coffee is inserted successful and the water level is normal, espresor starts the process of delivering the heated water from boiler trough the delivery pipe to the receptacle with coffee. After this step, the user gets his coffee in the pot.

```
5. Exit
3
The hot water is delivering to the pot...
Delivered.
Making the coffee...
The coffee is done

4. Waiting regime
5. Exit
```

## 4 WAITING REGIME

If user is not ready to drink his coffee, he can turn on the waiting regime and the pot will be heated trough plate heater to a constant, prescribed temperature – 70 °C. When pot raised this temperature, it will be maintained till the user will take his coffee.

```
4
The temperature at the moment is...
35
The temperature at the moment is...
42
The temperature at the moment is...
49
The temperature at the moment is...
56
The temperature at the moment is...
63
The temperature at the moment is...
70
Constant maintaining this temperature: 70
5. Exit
```

## 5 EXIT

Turn off the espresso.

```
5. Exit
5
Bon appetit!!!
```

## LIGHT INDICATOR

I've weighted all pros and cons and come to conclusion that it will be more practice to use text messages instead of color indicator, because it is a console output. When the espresso will be graphical represented, I will implement light indicators. ☺

## BUGS:

Somewhere if an int value is asked, but user writes string, it exits the code with Unhadled exception.

```
1
Enter the amount of water in ml (150 - 750):
gjdk
Unhandled exception. System.FormatException: The input string 'gjdk'
was not in a correct format.
   at System.Number.ThrowFormatException[TChar](ReadOnlySpan`1 value)
   at System.Convert.ToInt32(String value)
   at Espresso_Components.Espresso.WaterLevelChecker(Int32 amount) i
n C:\Users\ASUS\Esspresor_CSharp\espresso.cs:line 34
   at Program.Main(String[] args) in C:\Users\ASUS\Esspresor_CSharp\Pr
ogram.cs:line 35
```