Work on project. Stage 6/6: Brush up your code

Project: Coffee Machine

Easy 21 minutes

Description

Let's redesign our program and write a class that represents the coffee machine. The class should have a method that takes a string as input. Every time the user inputs a line (a string) to the console, the program invokes this method with one argument: the line that user input to the console. This system simulates pretty accurately how real-world electronic devices work. External components (like buttons on the coffee machine or tapping on the screen) generate events that pass into the single interface of the program.

The class should not use system input at all; it only should handle input that comes to it via this method and its string argument.

The first problem that comes to mind: how to write that method in a way that it represents all that coffee machine can do? If the user inputs a single number, how can the method determine what that number is: a variant of coffee chosen by the user or the number of the disposable cups that a special worker added into the coffee machine?

The right solution to this problem is to store the current state of the machine. The coffee machine has several states it can be in. For example, the state could be "choosing an action" or "choosing a variant of coffee". Every time the user inputs something and a program passes that line to the method, the program determines how to interpret this line using the information about the current state. After processing this line, the state of the coffee machine can be changed or can stay the same. The most efficient way of handling states is using an enum with all predefined states.

20 / 20 Prerequisites

- Units of information — In project — 13
- Computer programming — In project — 4
- Sizes and ranges — In project — 13
- Type casting — In project — 13
- Introduction to OOP — In project — 4

Show all

Join a study group for the project Coffee Machine

Discuss your current project with fellow learners and help each other.

Remember, that:

- For the espresso, the coffee machine needs 250 ml of water and 16 g of coffee beans. It costs $4.
- For the latte, the coffee machine needs 350 ml of water, 75 ml of milk, and 20 g of coffee beans. It costs $7.
- And for the cappuccino, the coffee machine needs 200 ml of water, 100 ml of milk, and 12 g of coffee. It costs $6.

Instruction

Refactor the program. Make it so that you can communicate with the coffee machine through a single method.

Example

Your coffee machine should have the same initial resources as in the example (400 ml of water, 540 ml of milk, 120 g of coffee beans, 9 disposable cups, $550 in cash).
The symbol `>` represents the user input. Note that it's not part of the input.

```
Write action (buy, fill, take, remaining, exit):
> remaining

The coffee machine has:
400 ml of water
540 ml of milk
120 g of coffee beans
9 disposable cups
$550 of money

Write action (buy, fill, take, remaining, exit):
> buy

What do you want to buy? 1 - espresso, 2 - latte, 3 - cappuccino, back - to main menu:
> 2
I have enough resources, making you a coffee!

Write action (buy, fill, take, remaining, exit):
> remaining

The coffee machine has:
50 ml of water
465 ml of milk
100 g of coffee beans
8 disposable cups
$557 of money

Write action (buy, fill, take, remaining, exit):
> buy

What do you want to buy? 1 - espresso, 2 - latte, 3 - cappuccino, back - to main menu:
> 2
Sorry, not enough water!

Write action (buy, fill, take, remaining, exit):
> fill

Write how many ml of water you want to add:
> 1000
Write how many ml of milk you want to add:
> 0
Write how many grams of coffee beans you want to add:
> 0
Write how many disposable cups of coffee you want to add:
> 0

Write action (buy, fill, take, remaining, exit):
> remaining

The coffee machine has:
1050 ml of water
465 ml of milk
100 g of coffee beans
8 disposable cups
$557 of money

Write action (buy, fill, take, remaining, exit):
> buy

What do you want to buy? 1 - espresso, 2 - latte, 3 - cappuccino, back - to main menu:
> 2
I have enough resources, making you a coffee!

Write action (buy, fill, take, remaining, exit):
> remaining

The coffee machine has:
700 ml of water
390 ml of milk
80 g of coffee beans
7 disposable cups
$564 of money

Write action (buy, fill, take, remaining, exit):
> take

I gave you $564
```

```
Write action (buy, fill, take, remaining, exit):
> remaining

The coffee machine has:
700 ml of water
390 ml of milk
80 g of coffee beans
7 disposable cups
$0 of money

Write action (buy, fill, take, remaining, exit):
> exit
```

Report a typo

See hint

Write a program

Code Editor IDE

Java

```
1  package machine;
2  import java.util.Scanner;
3
4  public class CoffeeMachine{
5      public static int water;
6      public static int milk;
7      public static int bean;
8      public static int dcup;
9      public static int money;
10
11     public static void initialize(){
12         water = 400;
13         milk = 540;
14         bean = 120;
15         dcup = 9;
16         money = 550;
17     }
18
19     public static void main(String[] args) {
20         initialize();
21         action();
22     }
23
24     public static void action(){
25         Scanner sc = new Scanner(System.in);
26         while(true){
27             System.out.println("\nWrite action (buy, fill, take, remaining, exit): ");
28             String str = sc.nextLine();
29             if(str.equals("exit")){
30                 return;
31             }
32             else if(str.equals("take")){
33                 take();
34             }
35             else if(str.equals("fill")){
36                 fill();
37             }
38
39             else if(str.equals("buy")){
40                 buy();
41             }
42             else if(str.equals("remaining")){
43                 machineState();
44             }
45         }
46     }
47
48     public static void buy(){
49         Scanner sc = new Scanner(System.in);
50         System.out.println("\nWhat do you want to buy? 1 - espresso, 2 - latte, 3 - cappuccino, back - to main menu:");
51         if(!(sc.hasNextInt())){
52             String st = sc.nextLine();
53             return;
54         }
55         int choice = sc.nextInt();
56         if(choice == 1){
57             if(isEnough(250, 0, 16)){
58                 System.out.println("I have enough resources, making you a coffee!\n");
```

```
                water -= 250;
                bean -= 16;
                money += 4;
                dcup--;
            }
        }
        else if(choice == 2){
            if(isEnough(250, 0, 16)){
                System.out.println("I have enough resources, making you a coffee!\n");
                water -= 350;
                milk -= 75;
                bean -= 20;
                money += 7;
                dcup--;
            }
        }
        else if(choice == 3){
            if(isEnough(250, 0, 16)){
                System.out.println("I have enough resources, making you a coffee!\n");
                water -= 200;
                milk -= 100;
                bean -= 12;
                money += 6;
                dcup--;
            }
        }
    }

    public static boolean isEnough(int wate, int mil, int bea){
        boolean a = true;
        if(water < wate){
            a = false;
            System.out.println("Sorry, not enough water!");
        }
        if(milk < mil){
            a = false;
            System.out.println("Sorry, not enough milk!");
        }
        if(bean < bea){
            a = false;
            System.out.println("Sorry, not enough coffee beans!");
        }
        if(dcup < 1){
            System.out.println("Sorry, not enough disposable cups");
            a = false;
        }
        return(a);
    }

    public static void fill(){
        Scanner sc = new Scanner(System.in);
        System.out.println("\nWrite how many ml of water you want to add:");
        water += sc.nextInt();
        System.out.println("Write how many ml of milk you want to add:");
        milk += sc.nextInt();
        System.out.println("Write how many grams of coffee beans you want to add:");
        bean += sc.nextInt();
        System.out.println("Write how many disposable cups of coffee you want to add:");
        dcup += sc.nextInt();
    }

    public static void take(){
        System.out.println("\nI gave you $" + (money));
        money = 0;
    }

    public static void machineState(){
        System.out.println("\nThe coffee machine has:");
        System.out.println((water)+ " ml of water");
        System.out.println((milk) + " ml of milk");
        System.out.println((bean) + " g of coffee beans");
        System.out.println((dcup) + " disposable cups");
        System.out.println("$" +(money)+ " of money");
    }

    public static void printOrder(int cup, int order){
        if(cup == order){
            System.out.println("Yes, I can make that amount of coffee");
        }
        else if(cup < order){
            System.out.println("No, I can make only " + (cup) + "cup(s) of coffee");
        }
        else{
            System.out.print("Yes, I can make that amount of coffee (and even " + (cup - order) + "more than that)");
        }
    }
```

```
145
146     public static int getCups(int water, int milk, int coffee){
147         int cup = 0;
148         while(!(water < 1 && milk < 1 && coffee < 1)){
149             water -= 200;
150             milk -= 50;
151             coffee -= 15;
152             if(water >= 0 && milk >= 0 && coffee >= 0){
153                 cup++;
154             }
155         }
156         return(cup);
157     }
158 }
```

✓ Correct.

That's an awesome solution! What do you think about showing it off? Post it to Solutions so other learners can enjoy it too.

329 users liked this problem. 87 didn't like it. **What about you?**

Comments (160) Hints (13) Useful links (0) Solutions (510) Hide discussion

U2 Share something, User 214842039

Post Place a bounty Please do not post solutions here

Sort by: Last posted

RG **Ricardo Garcia** 4 days ago

If you want to make it even easier for yourself (or harder depending on your skill level) I found it easier to make another Coffee Class so that I could create Espresso, Latte and Cappuccino Objects with private final fields which are assigned in the constructor. I then used Getter methods to have these objects interact with the program rather than having to repeat many lines of code for just all the different variables for different coffee types. - Just an idea! (Oh also if you go this route I'd create a second method called 'buyCoffee' or something which takes the parameter of an object of type 'Coffee', makes it much

Reply Report

C **chuan deeugue** about 2 months ago

I didn't change anything in my code and it was correct immediately when I ran it....

Reply Report

 **Jaime Bolaños** about 2 months ago

Are you able to continue with the rest of tasks or select a new project?

Reply Report

C **chuan deeugue** about 2 months ago

yes, guess i already did it right in Stage 5/6.

Reply Report

U1 **User 185935713** about 1 month ago

Same here, lol

Reply Report

RG **Ricardo Garcia** 4 days ago

that's because the program is identical, you are supposed to make it more friendly to other programmers here using the OOP principals, yeah u can just use what u already have and move on but then you do not learn anything about OOP.

Reply Report

 **Deepak Pitaliya** about 2 months ago

I am having trouble understanding the task. The description says :- "The class should not use system input at all"
Does it mean not to use Scanner utility?

Reply Report

F **Frank** about 2 months ago

"Yes" for the class, "No" for the main method. So inside your main method, you'd probably have some scanners reading from the System.in and pass the inputs to the coffee machine object which has its own single(required in the instruction) interface to communicate.

Reply Report

MM **Margarita Marchenko** 2 months ago

I fought there should be two classes: "The class should not use system input at all; it only should handle input that comes to it via this method and its string argument." But all wrote only one class.
The method with input belongs to the class, how can it be "The class should not use system input at all"?

Another quietion is: all class must have main method. Why this construct is working in my code?

Reply Report

Hubert Michalec 2 months ago Moderator

Only one class has to have main method, otherwise you will get error. JVM searches class with public static void main and run it.

Reply Report

MM **Margarita Marchenko** 2 months ago

O_o I'm not shure I've already learned this. Thanks!

Reply Report

Hubert Michalec 2 months ago Moderator

main method should be in class given by hyperskill because test try to invoke it from this class.
When I doing this project it was class: CoffeeMachine

Reply Report

Hubert Michalec 2 months ago Moderator

Good luck, see you in another thread :D

2 Reply Report

MM **Margarita Marchenko** 2 months ago

I've just tried another task and got an error: Error: Main method not found in class ManufacturingController, please define the main method as:
  public static void main(String[] args)

But it not an entrypoint class!!! Have a look:

Reply Report

J **Justsomeone** 2 months ago

ok @Hubert and have a nice day :)

1 Reply Report

J **Justsomeone** 2 months ago

cause the test did not check if the user has 2 class or only one so you can say they ignore it
and maybe some use one class and make one method responsible for reading and the the logic in the remaining method of the class

1 Reply Report

Hubert Michalec 2 months ago Moderator

@Justsomeone I agree, writing about error I though about people were solving problems on JB and added their own main method and asking "why solution is wrong" and these people had error "two main classes".
There is no point conduct long discussion about main method because it so obvious. One main method in manifest as entry point for whole program.
Anyway, there is no point to create more main method except testing cause.

Reply Report

MM **Margarita Marchenko** 2 months ago

Okey then, because many solutions with only one class and it's confusing.

Reply Report

J **Justsomeone** 2 months ago

what i got from this `Only one class has to have main method, otherwise you will get error`
from my humble knowledge on english only one is not same at least
only one class mean that no more than one class to have main method that did not look like at least one which mean it possible to have on e or more
and of course i talk about exact main method

1 Reply Report

Hubert Michalec 2 months ago Moderator

To be precise, because I express my thoughts in vague way. Each class can contain `public static void main(String[] args)`. At least one class must contains it to run application.
You can have `public static void main(int[] args)` but it isn't main class which can be entry point for app.

1 Reply Report

Hubert Michalec 2 months ago Moderator

Saying about one main method I mean `public static void main(String[] args)` as entry point for application. At least one class must contain it.
Check this topic for more information https://hyperskill.org/learn/step/3489

Reply Report

J **Justsomeone** 2 months ago

you can have main method on each class the jvm would not mind jvm only care that the class you call it (the one you start your app with to have the main method)

for this part The class should not use system input at all" they meant that one class to read the input using the

Reply Report

MM **Margarita Marchenko** 2 months ago

I see. Many thanks for help!

Reply Report

JC **Jon C** 4 months ago

spent some hours on this... only called scanner.nextLine() in one method then pass the input to other methods and process accordingly base on the machine state. cheers.

Reply Report

LeonidBresjnev 4 months ago

I must confess that I have no clue how to solve this excerise correctly. Some examples prior to this exercise is neccessary to understand how to get the input rightly. The solution from previous task was accepted, but I would indeed like to know what was expected here.

Reply Report

Shapnesh Singh Tiwari 3 months ago Moderator

Check that after filling the coffee machine you should use scanner.nextLine(), try to figure out by trying different things.

1 Reply Report

Shapnesh Singh Tiwari 3 months ago Moderator

Also stage 5 code will pass this but you should try to use only one method as mentioned.

1 Reply Report

B **Vera** 5 months ago

Please help
What to do with bunch of inputs when filling coffeemachine?

Reply Report

U1 **User 155573660** 3 months ago

-Use it as parameter in a method
- This method should contain many switch statements that will use the input
as "switching "variable

Reply Report

B **Vera** 5 months ago

Will it be logical if I print "what coffee do you want" in the main class?

Reply Report

Dostonbek Matyakubov 5 months ago

You will simply get errors when you run your code. This is not what you are requested to print. In order to pass the stage, you must do exactly what is requested.

Reply Report

B **Vera** 5 months ago Bountied

Class CoffeeMachine should be in the main file or we should create another file with this class&

Reply Report

Dostonbek Matyakubov 5 months ago

It should be your main file. You can just create one public class for CoffeeMachine and keep all other classes/enums in this file. To do so, you need to declare other classes/enums without keyword public

2 1 Reply Report

NN **Nam Nguyễn** 6 months ago

I just simply copied 100% from stage 5 and BOOM, passed !!

Reply Report

Da **Deleted account** 6 months ago

Try to cut your program in classes, try to use enum, user interface, switch and methods. Just add code from previouse stage it's not interesting. Better doing practice and check your code with this stage checker.

Reply Report

Igor Ch 6 months ago Moderator

Nothing new has been added. Just make something similar to OOP out of spaghetti-code.:)

Reply Report

Tien Tung Nguyen 5 months ago

The checking machine just checks the output of your programm, I guess. But its here all about self learning, it makes no sense when you dont want to apply your new knowledge :))

Reply Report

C **Constantin Balan** 7 months ago

The solution i had from Stage 5/6 is cleaner than most solutions posted here. It's easier to understand and shorter, which i think is the idea of writing good code. I feel like the last topics could have been exemplified better in other smaller programs, as i feel like implementing them here doesn't improve the code much and i don't feel like i got a good grasp of the subjects unfortunately

Reply Report

C **Constantin Balan** 7 months ago

Hopefully the next topics/ projects will make the info stick better

Reply Report

Maurits Silvis 10 months ago

It seems that very few people have carefully considered the line "The [coffee machine] class should not use system input at all; it should only handle input that comes to it via [a special input] method and its string argument." This line means that the coffee machine should not retrieve input itself. In other words, it should neither include a scanner, nor (indirectly) make use of it. Having implemented a solution myself, in which the coffee machine is given input (not taking it itself), I must say though that this exercise goes far and by -- I'd say too far -- beyond what was required in any of the previous project stages, especially

Reply Report

PR **Pawel Raus** 10 months ago

What's the actual point of using enums like that? I mean, what is the difference if I make 2 switch statements, 1st for input and changing the state and the second for performing action depending on the state? Why not just make 1 switch statement depending on the input and instead of changing the state, just perform the action already? The first and only reason I think of would be maybe the fact that the possible error that'd occur in the program would be easier to localize and analyze when you know in which state machine was just before the error occurred

Reply Report

> **Maurits Silvis** 10 months ago
>
> I guess the problem is the following: The current exercise describes that the coffee machine should get one line of input at a time through a specific method. The coffee machine should not incorporate a scanner that allows the coffee machine to take input itself. Then the following scenario can arise: The first line of input provided to the coffee machine is "buy". While waiting until the next line of input ("1", for example) comes and can be processed, the coffee machine has to remember that it is the 'buying' state. In other words, a coffee machine that is designed to expect one command
>
> Reply Report

Matyas Kocka 10 months ago

the enums here are absolutely useless.. and even harder to use. not the right time I suppose. Quite good source about enums was: https://www.youtube.com/watch?v=LYKHxwQ0QH8 check it out :)

1 Reply Report

> **Maurits Silvis** 10 months ago
>
> Why is a switch with an enum like `enum States {BUY, FILL, TAKE, REMAINING}` so much harder than a switch with the strings "buy", "fill", "take" and "remaining"?
>
> Reply Report

Ekaterina Bogacheva 11 months ago

Check if the IDE is updated to the last version (that was my problem).

1 2 Reply Report

SM **Sabir Mammadov** about 1 year ago

Error in test #1

Cannot find a class with a main method.
Check if you declared it as "public static void main(String[] args)".

Reply Report

> **Igor Ch** about 1 year ago Moderator
>
> Hi, if you still have this problem, please contact our Support Team
> https://support.hyperskill.org/hc/en-us/requests/new
>
> Reply Report

IT **Issatay Tleuov** about 1 year ago

Guys, can you help me?
If I want to solve a task in IDE. The window where I have a task gives me an error "Cannot Open the Page

file:///jbcefbrowser/331464568#url=about:blank

Reply Report

Rogan7007 about 1 year ago

worked on this for like 2 weeks just to find out my previous code from stage5 met the requirements. really dont understand WHY it met the requirements it certainly used system input and did not return strings via a single method...i think the testing is broken or not strict enough or something. i dont feel like i learned what i needed to in this chapter.

Reply Report

> **Moeen Qureshi** about 1 year ago
>
> Totally feel the same. I felt this chapter was essential but was not taught well enough.
>
> Reply Report
>
> > **Matyas Kocka** 10 months ago
> >
> > yep. same for me and I had to find a lot of info on the net.... could have been done better.
> >
> > Reply Report
>
> **Maurits Silvis** 10 months ago
>
> Yup, I also accidentally found out that the stage-five code passed the stage-six tests. I guess, only the functioning of the coffee machine is tested and the internals are ignored in the tests -- unfortunately!
>
> Reply Report

U3 **User 35277276** about 1 year ago

After last changes in this topic i decided to pass a last part. It possible to observe that it would take a 7 minutes to complete - what a joke! I took for me more then 7 hours to refactor: i had to use external resources to discover all possible uses of enums that not covered here and practice usage of enums. However know i believe that my solution is the best so far - please look Solutions sections.

1 Reply Report

> **Rogan7007** about 1 year ago
>
> your code really is great. taught me a bit about enums just reading it.
>
> 2 Reply Report

a **apexpred** about 1 year ago

it feels kind of stupid to have number input for the buy part (1,2,3) and the last option you need to type a string to go back to main menu. seems pretty inconsistent

2 Reply Report

> **Valentina V** 7 months ago
>
> you can process number input as string : "1", "2", "3"
>
> Reply Report