Main changes

1. I introduced fille *settings* where all constants are listed and which can be edited in single place before running the program.
2. I added new class *Checkout* in a new file.
3. When loading the snack at the start of the program or from admin menu one is asked to specify number of each snack and its price.
4. There is a *snack\_menu* variable in *settings* file which automatically changes following number of snacks listed in *INITIAL\_SNACK\_PRICES*. It is later used in *main* for snacks selection .

intro

- for loading the snacks you could instead have a method that supplies the machine with snacks (supply machine) and then put the conditional statement into that method

I’m not sure how that could work? The *supply machine* method would have to be in the *Storage* class but it would have to communicate with *Interface* to gather the user’s input (number of snacks) and that would break SRP as *Storage* would collect data from *main* and *Interface*.

shopping sequence

- The if statements are very long:

I agree that if statements are long and that there is a long list of ELIFs there but to be honest I don’t know how to make them shorter or how to avoid neverending ELIFs. Do you have any suggestions?

- You have hard-coded the user input to specific values, what if these values change or you want to add more? Try using something like an enumerator which basically sets a string to a value, or a dictionary that allows you to define the values before running the code

- e.g. with a dictionary: if user\_input == user\_commands["confirm\_purchase"]:

would allow you to easily change the value of "confirm\_purchase" without having to manually change the method. The dictionary could be defined as a constant at the top of the file/class.

I understand your intention but in this case, we have the machine with a numeric keyboard only so each key is responsible for assigned selection from the menu. I don’t want a user to have to type his commands as typing just a number is easier. Moreover, I use the same selection numbers in 2 different menus in *shopping\_sequence* and *asmin\_choice* functions so I would have to make 2 independent dicts to create different menus. But I replaced (1,2,3,4,5) tuple called several times with a single variable now.

- some of the statements, like for line 63: "user confirms purchase" you could have have a class called checkout with a method called complete checkout that runs all of those methods

- So overall it would something like:

- if user\_input == user\_commands["confirm\_purchase"]:

- checkout.complete\_checkout()

I did as suggested but hen I have all 5 remaining classes passed to the new one Checkout, it definitely brakes the SRP. How can I avoid that?