Group Members: Richard Padilla, Francisco E. Alderete, Karl Dill, Venkata Surya Dasari

Assignment 3 Rationale (Java):

Java-Calculator represents the same work I did as a team with Francisco, Karl, and Venkata on the Python version - I just re-implemented the code in Java so I could personally better understand the Design Patterns at work.

I believe there *may* be some technical language details within python that keep it from truly implementing the State Design Pattern & Singleton Design Pattern to the fullest.

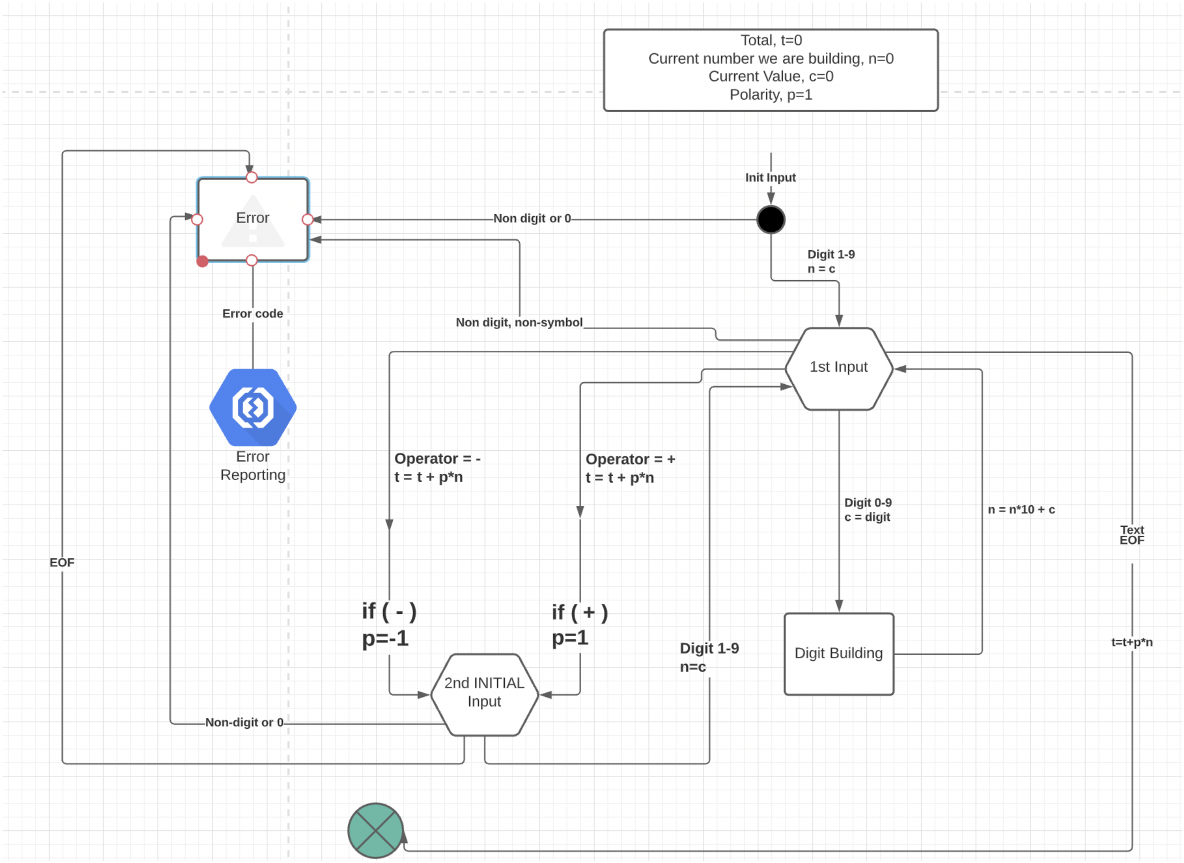
It seems that there are no Pointers in Python and as such it is hard to tell if you are passing the actual Context or just the values of the Context to a given State. Small personal testing led me to believe that Python was merely passing the values and not the true Context Object.

Additionally, It seems that there is not a good way to implement Singletons in Python, because as described above, Python does not support Pointers. Since this is the case I was not sure how to keep a reference to a single State object, to be passed to the Context.

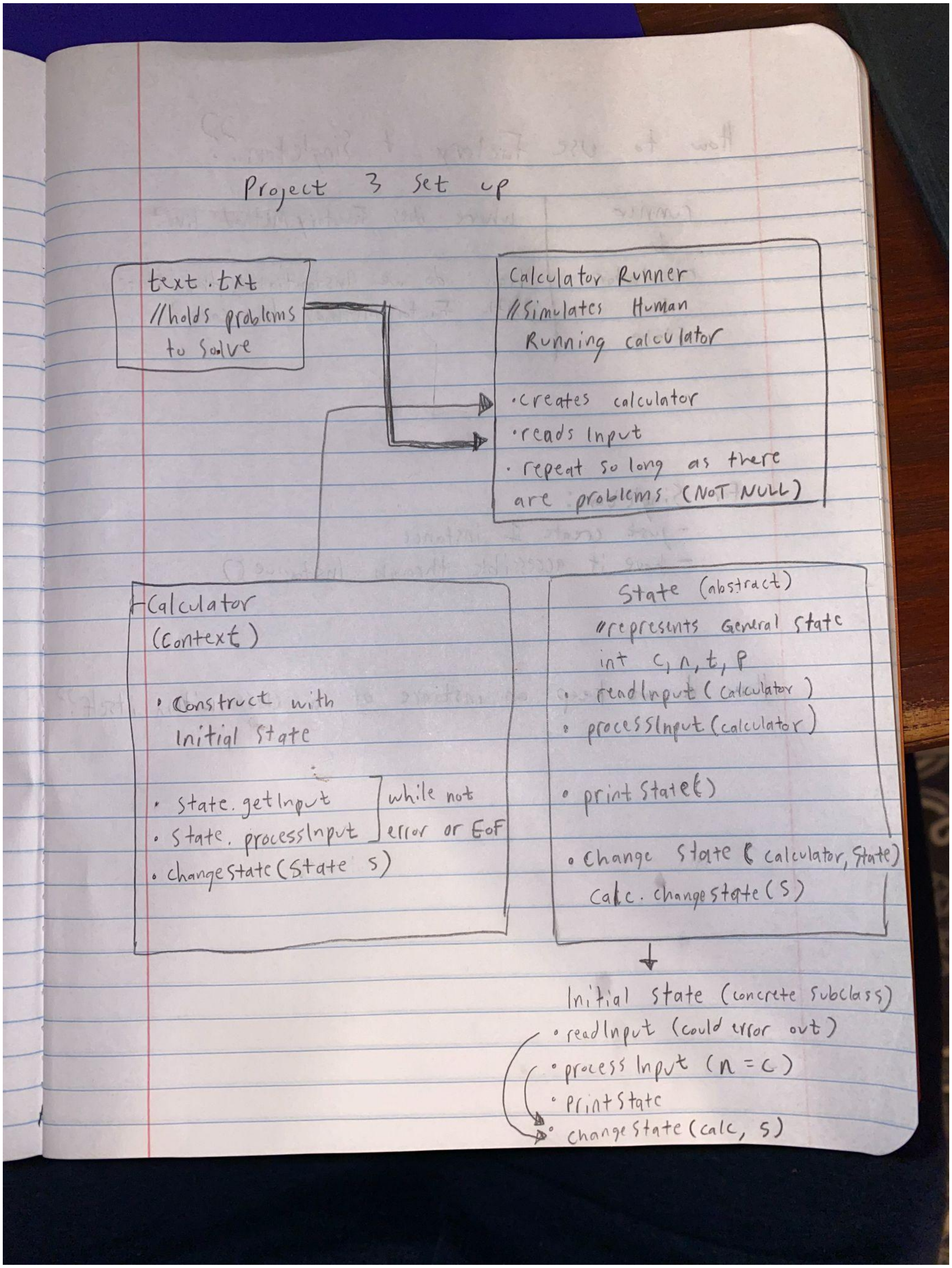
Ultimately I decided to code up an additional version of the Simple Calculator myself in Java because I was able to pass by reference and ensure that I was changing the actual Context, and using the sole Singleton Instances of a given state.

Below is the Team Lucid chart as well as a rough sketch of the Diagram I followed when designing my code:

Group Members: Richard Padilla, Francisco E. Alderete, Karl Dill, Venkata Surya Dasari



Group Members: Richard Padilla, Francisco E. Alderete, Karl Dill, Venkata Surya Dasari



I have a CalculatorRunner class. This class holds the main() function.

Group Members: Richard Padilla, Francisco E. Alderete, Karl Dill, Venkata Surya Dasari

The main() function does the following:

- reads in an input file which has all the given problems to be solved
- creates an instance of Calculator (aka the Context)
- attempts to solve a given problem

The CalculatorContext class represents a Calculator Object, and is the Context part of the State Design pattern.

The CalculatorContext does the following:

- sets the initial state in its constructor
- attempts to solve a given problem through the use of States
- has the ability to change states (which is done THROUGH the states)

The State class represents a State Object, and represents the interface that CalculatorContext uses to interact with a given state without having to know exactly which state the Calculator is in..

The State class does the following:

- sets up the interface for CalculatorContext to use

The State classes (e.g. InitialState) represent ConcreteState Objects, and are the individual States of the State Design Pattern

A given State class does the following:

- processes the current Input
- Prints out the state, as well as relevant Context information
- Change the state of the Context internally

Note: All Concrete State objects are Singletons.

References:
1. Design Pattern 1 Slides & In class discussion
2. Design Patterns Book by Erich Gamma, Richard Helm, Ralph Johnson, & John Vlissides
3. https://refactoring.guru/design-patterns/ for additional help with understanding the Patterns