# Compiler

* Variables
  + HashMap<String,String> variables
    - This holds the variables from the program the first string is
    - The hash is made from the name string
    - The value is also sorted as a string
  + ArrayList<Strings> commands
    - This is a temporary holder for the interpreter to go through and run the various things that program wants
  + Boolean error
    - This is a flag
    - If the interpreter runs into a syntax error then this will keep the program from running any further
  + JTextArea console
    - This is where the program outputs too
* Methods
  + Constructor(JTextArea console)
    - Declares the console, the variables, and error
  + isError()
    - returns the error Boolean
  + canRun()
    - returns true if commands.size() >0
  + run()
    - this takes the commands sends it to parseLine
    - if an error occurs from that the loop stops and the console outputs to the user that there is an error
  + parseLine(String command)
    - This takes the string input from run and determines how to handle it
    - If there is a syntax error then the flag is set to true
  + Trim(String input)
    - This gets rid of the spaces in the string
  + isNumeric(String input)
    - checks to see if a string has nothing but numbers in it
    - I would use a regex but I also need it to not count . and – for decimals and negative signs
    - Returns a boolean
  + setFile(File file)
    - this takes a file and makes sure it is a .txt file and then loads the file into the arraylist commands

# IDE

* This extends JPanel, and implements ActionListener
* Variables
  + JButton open, save, run
    - These buttons are self-explanatory
  + JTextArea program, console
    - The program is where you can edit the program after you open it in the IDE
    - The console is linked to the compiler
  + JLabel consoleLabel
    - This is for the words that show up above the console “Console:”
  + Final JFileChooser
    - This is for the gui file browser
  + Compiler
    - This is so that the compiler can be linked to the console and to declare it
* Methods
  + Constructor
    - Calls it’s super constructor with a new border Layout
      * This causes the Panel’s layout to be a border layout
    - Declares all buttons and adds a the action listener to them
    - The buttons are put in their own panel with a flow layout
    - A programLabel is created
    - programLabel and program are put in a another BorderLayout which is put in a panel
    - The same thing is done with the console
    - The programPanel and buttonPanel are put in a userPanel
    - The userPanel and the console Panel are added to the IDE
  + getConsole()
    - returns the console
  + actionPerfomed()
    - when open is pressed the JFileChooser springs into action and loads the file into the program JTextArea
    - when run is clicked it runs the program that is loaded
    - when save is clicked it saves the program that was edited
    - when clear is clicked it clear the console so that it doesn’t get crowded

# Main

* This runs everything
* Variables
  + None
* Method
  + Main
    - This sets the look and feel to the system look and feel
    - A JFrame is then made and everything is set up
    - It then displays the JFrame