Computer Science Mid-Term Project by Group(13)

8.3.2025

Chosen Topic: Calculator

Team Members

- Rachata Raktham
- Aung Thuya Han
- Pamoda Tharangani Kodikara
- Gayanga Dilhani Paranagamage
- Amir Oladi
- Joseph Gabrielle De Arco

Our Project can be found on github at https://github.com/Amirol89/final-project.git

Interactive Calculator

This project implements an interactive calculator built using Python and the ipywidgets library within a Jupyter Notebook environment. The calculator provides a user-friendly interface for performing basic arithmetic operations and mathematical functions, including addition, subtraction, multiplication, division, squaring a number, and calculating square roots.

In [ ]:
```
```

Import required libraries for creating interactive widgets and mathematical operations

In [ ]:
```
import ipywidgets as widgets
from IPython.display import display, clear_output
import math
```

Define a Calculator class and initialize the calculator by creating all necessary methods

In [ ]:
```
class Calculator:
    def __init__(self):
        self.create_widgets()

    def add(self, a, b):
        return a + b

    def subtract(self, a, b):
        return a - b

    def multiply(self, a, b):
        return a * b

    def divide(self, a, b):
        if b == 0:
            return "Error: Division by zero"
        return a / b

    def power(self, a):
        return a ** 2

    def square_root(self, a):
        if a < 0:
            return "Error: Cannot take square root of a negative number"
        return math.sqrt(a)
```

Main calculation method that handles all operations

In [ ]:
```
def calculate(self, change=None):
        operation = self.select_operation.value
        num1 = self.num1.value

        if operation in ["Second Power", "Square Root"]:
            self.num2_box.layout.display = 'none'
        else:
            self.num2_box.layout.display = 'flex'
```

```
        try:
            if operation in ["Addition", "Subtraction", "Multiplication", "Division"]:
                num2 = self.num2.value
                if operation == "Addition":
                    result = self.add(num1, num2)
                elif operation == "Subtraction":
                    result = self.subtract(num1, num2)
                elif operation == "Multiplication":
                    result = self.multiply(num1, num2)
                elif operation == "Division":
                    result = self.divide(num1, num2)

            elif operation == "Second Power":
                result = self.power(num1)
            elif operation == "Square Root":
                result = self.square_root(num1)

        except Exception as e:
            result = f"Error: {e}"

        self.result.value = f"Result: {result}"
```

Create and set up all UI widgets

In [ ]:
```
def create_widgets(self):
        self.select_operation = widgets.Dropdown(
            options=["Addition", "Subtraction", "Multiplication", "Division", "Second Power", "Square Root"
            value="Addition",
            layout=widgets.Layout(width="200px")
        )
        self.select_operation.observe(self.calculate, names='value')

        self.num1 = widgets.FloatText(layout=widgets.Layout(width="200px"))
        self.num2 = widgets.FloatText(layout=widgets.Layout(width="200px"))

        self.num1_box = widgets.HBox(
            [widgets.Label("Enter 1st Number:", layout=widgets.Layout(width="120px")), self.num1])
        self.num2_box = widgets.HBox(
            [widgets.Label("Enter 2nd Number:", layout=widgets.Layout(width="120px")), self.num2])

        self.calc_button = widgets.Button(description="Calculate", layout=widgets.Layout(margin="10px 0 0 0
        self.calc_button.on_click(self.calculate)

        self.result = widgets.Label(value="Result: ", layout=widgets.Layout(margin="10px 0 0 0"))

        display(
            widgets.VBox([
                widgets.HBox(
                    [widgets.Label("Select Operation:", layout=widgets.Layout(width="120px")), self.select_
                self.num1_box,
                self.num2_box,
                self.calc_button,
                self.result
            ])
        )
        self.calculate()
```

Create and intialize display of the calculator

In [ ]:
```
Calculator();
```

In [ ]:
```

```

The full code itself created by our members and testings are provided below.

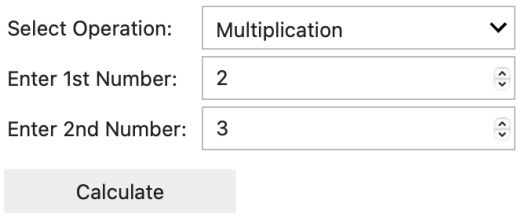
In [ ]:
```

```

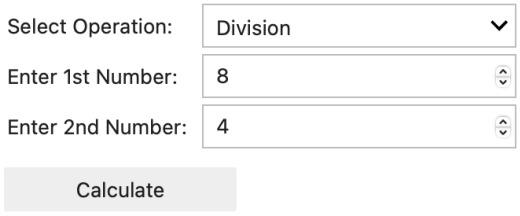
In [ ]:
```
import ipywidgets as widgets
from IPython.display import display, clear_output
import math


class Calculator:
    def __init__(self):
        self.create_widgets()

    def add(self, a, b):
        return a + b
```

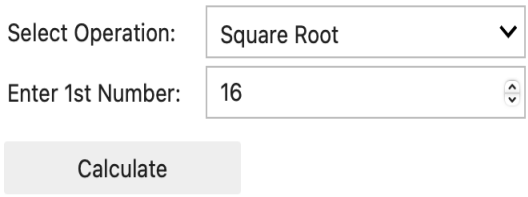
```
    def subtract(self, a, b):
        return a - b

    def multiply(self, a, b):
        return a * b

    def divide(self, a, b):
        if b == 0:
            return "Error: Division by zero"
        return a / b

    def power(self, a):
        return a ** 2

    def square_root(self, a):
        if a < 0:
            return "Error: Cannot take square root of a negative number"
        return math.sqrt(a)

    def calculate(self, change=None):
        operation = self.select_operation.value
        num1 = self.num1.value

        if operation in ["Second Power", "Square Root"]:
            self.num2_box.layout.display = 'none'
        else:
            self.num2_box.layout.display = 'flex'

        try:
            if operation in ["Addition", "Subtraction", "Multiplication", "Division"]:
                num2 = self.num2.value
                if operation == "Addition":
                    result = self.add(num1, num2)
                elif operation == "Subtraction":
                    result = self.subtract(num1, num2)
                elif operation == "Multiplication":
                    result = self.multiply(num1, num2)
                elif operation == "Division":
                    result = self.divide(num1, num2)

            elif operation == "Second Power":
                result = self.power(num1)
            elif operation == "Square Root":
                result = self.square_root(num1)

        except Exception as e:
            result = f"Error: {e}"

        self.result.value = f"Result: {result}"

    def create_widgets(self):
        self.select_operation = widgets.Dropdown(
            options=["Addition", "Subtraction", "Multiplication", "Division", "Second Power", "Square Root"
            value="Addition",
            layout=widgets.Layout(width="200px")
        )
        self.select_operation.observe(self.calculate, names='value')

        self.num1 = widgets.FloatText(layout=widgets.Layout(width="200px"))
        self.num2 = widgets.FloatText(layout=widgets.Layout(width="200px"))

        self.num1_box = widgets.HBox(
            [widgets.Label("Enter 1st Number:", layout=widgets.Layout(width="120px")), self.num1])
        self.num2_box = widgets.HBox(
            [widgets.Label("Enter 2nd Number:", layout=widgets.Layout(width="120px")), self.num2])

        self.calc_button = widgets.Button(description="Calculate", layout=widgets.Layout(margin="10px 0 0 0
        self.calc_button.on_click(self.calculate)

        self.result = widgets.Label(value="Result: ", layout=widgets.Layout(margin="10px 0 0 0"))

        display(
            widgets.VBox([
                widgets.HBox(
                    [widgets.Label("Select Operation:", layout=widgets.Layout(width="120px")), self.select_
                self.num1_box,
                self.num2_box,
                self.calc_button,
                self.result
            ])
        )
        self.calculate()
```

```
Calculator();
```


Select Operation:
Addition
Enter 1st Number:
2
Enter 2nd Number:
3
Calculate
Result: 5.0


Select Operation:
Multiplication
Enter 1st Number:
2
Enter 2nd Number:
3
Calculate
Result: 6.0


Select Operation:
Division
Enter 1st Number:
8
Enter 2nd Number:
4
Calculate
Result: 2.0


Select Operation:
Square Root
Enter 1st Number:
16
Calculate
Result: 4.0