

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”
ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Звіт
про виконання лабораторної роботи №2
з дисципліни
«Спеціалізовані мови програмування»

студента групи ІТ-32
Костельного Д.-А. А.

прийняв
Щербак С. С.

Мета роботи: розробка консольного калькулятора в об'єктно орієнтованому стилі з використанням класів.

Опис виконаної роботи

Код головного фалу програми та класу Calculator:

domain/calculator.py

```
from config.opts import read_opts
from domain.action import Action
from domain.state import CalculatorState
from std.read import read_choose_from_list
from std.repeat import repeat_while_requested

class Calculator:
    def __init__(self, actions: list[Action]):
        self.actions = actions
        self.state = CalculatorState()

    def perform_action(self, action: Action):
        config_opts = read_opts()
        action(opts=config_opts, state=self.state)

def fmt_action(action: Action) -> str:
    return action.name

def request_action(action_list: list[Action]) -> Action:
    return read_choose_from_list(
        options=action_list, title="Choose operation:", formatter=fmt_action
    )

class Runner:
    def __init__(self, calculator: Calculator):
        self.calculator = calculator
        pass

    def run_thread(self) -> None:
        action = request_action(self.calculator.actions)
        self.calculator.perform_action(action=action)

    def run(self) -> None:
        repeat_while_requested(action=self.run_thread)
```

main.py

```
from actions.calc_action import calc_action
from actions.config_action import config_action
from actions.exit_action import exit_action
from actions.history_action import history_action
from actions.memory import clear_memory_action, save_to_memory_action
from domain.action import Action
from domain.calculator import Calculator, Runner
```

```

actions = [
    Action(name="Change config", action_fn=config_action),
    Action(name="Display history", action_fn=history_action),
    Action(name="Clear memory", action_fn=clear_memory_action),
    Action(name="Save to memory", action_fn=save_to_memory_action),
    Action(name="Do calculation", action_fn=calc_action),
    Action(name="Exit", action_fn=exit_action),
]

def main():
    calculator = Calculator(actions=actions)
    runner = Runner(calculator=calculator)
    runner.run()

if __name__ == "__main__":
    main()

```

1. Обчислення.

На Рис. 1. зображено спробу виконання обчислення, включно з діленням на 0:

```

(lab-2) [dkostmii@hello-world lab-2]$ python main.py
Choose operation:
1. Change config
2. Display history
3. Clear memory
4. Save to memory
5. Do calculation
6. Exit
Your choice: 5
Choose an operation [+ , - , * , / , ^ , sqrt , %]: /
Enter a first number: 1
Enter a second number: 0
Error: Cannot divide by zero.
Do you want to continue? [y/N]: █

```

1

Рис. 1. Спроба виконання обчислення, включно з діленням на 0.

На Рис. 2. зображено спробу введення неправильної операції обчислення:

```
Choose operation:
1. Change config
2. Display history
3. Clear memory
4. Save to memory
5. Do calculation
6. Exit
Your choice: 5
Choose an operation [+ , - , * , / , ^ , sqrt , %]: --
Invalid operator. Try again
Choose an operation [+ , - , * , / , ^ , sqrt , %]: █
```

Рис. 2. Спроба виконання неіснуючої операції "--"

2. Запис та відображення історії обчислень.

На Рис. 3. зображено спробу відображення історії нещодавніх обчислень:

```
Choose operation:
1. Change config
2. Display history
3. Clear memory
4. Save to memory
5. Do calculation
6. Exit
Your choice: 2
Displaying last 10 history entries:
3.0 * 2.0 = 6.0
6.0 / 2.0 = 3.0
5.0 - 5.0 = 0.0
1.0 - 2.0 = -1.0
2.0 + 1.0 = 3.0
1.0 / 0.0 = Error: Cannot divide by zero.
Do you want to continue? [y/N]: █
```

Рис. 3. Відображення історії нещодавніх обчислень.

3. Функція пам'яті

На Рис. 4. зображено збереження результату обчислення у пам'ять для використання у наступних операціях обчислення:

```
Choose operation:
1. Change config
2. Display history
3. Clear memory
4. Save to memory
5. Do calculation
6. Exit
Your choice: 5
Choose an operation [+ , - , * , / , ^ , sqrt , %]: +
Enter a first number: 123
Enter a second number: 321
=> 444.0
Do you want to continue? [y/N]: y
Choose operation:
1. Change config
2. Display history
3. Clear memory
4. Save to memory
5. Do calculation
6. Exit
Your choice: 4
Value 444.0 saved to memory.
Do you want to continue? [y/N]: y
Choose operation:
1. Change config
2. Display history
3. Clear memory
4. Save to memory
5. Do calculation
6. Exit
Your choice: 5
Choose an operation [+ , - , * , / , ^ , sqrt , %]: +
The memory contains value [444.0]. Do you want to use that value? [y/N]: y
```

Рис. 4. Запис результату обчислення у пам'ять та подальше використання

5. Налаштування користувача

На Рис. 5.1. та 5.2. зображено зміну та результат зміни налаштувань користувача, а саме – кількості десяткових розрядів та кількості записів історії нещодавніх обчислень.

```
Choose operation:
1. Change config
2. Display history
3. Clear memory
4. Save to memory
5. Do calculation
6. Exit
Your choice: 1
Current config:
1. decimals => 0
2. history_count => 10

Choose config option to change:
1. decimals
2. history_count
Your choice: 1
Enter a value for decimals: 5
Set option decimals to 5.
Do you want to continue? [y/N]: y
Choose operation:
1. Change config
2. Display history
3. Clear memory
4. Save to memory
5. Do calculation
6. Exit
Your choice: 1
Current config:
1. decimals => 5
2. history_count => 10

Choose config option to change:
1. decimals
2. history_count
Your choice: 2
Enter a value for history_count: 2,
```

*Рис. 5.1. Зміна налаштувань
користувача.*

```

(lab-2) [dkostmii@hello-world lab-2]$ python main.py
Choose operation:
1. Change config
2. Display history
3. Clear memory
4. Save to memory
5. Do calculation
6. Exit
Your choice: 5
Choose an operation [+ , - , * , / , ^ , sqrt , %]: +
Enter a first number: 2
Enter a second number: 2
=> 4.00000
Do you want to continue? [y/N]: y
Choose operation:
1. Change config
2. Display history
3. Clear memory
4. Save to memory
5. Do calculation
6. Exit
Your choice: 2
Displaying last 2 history entries:
123.00000 + 321.00000 = 444.00000
2.00000 + 2.00000 = 4.00000
Do you want to continue? [y/N]: █

```

Рис. 5.3. Результат зміни налаштувань користувача.

Висновок: виконавши ці завдання, я перетворив консольний калькулятор у об'єктно-орієнтований калькулятор, використовуючи класи в Python. Цей проект допоміг мені вивчити концепції об'єктно-орієнтованого програмування та організацію, зберігаючи функціональність і інтерфейс користувача калькулятора.