Лабораторна #2: Calculator

Порядок виконання роботи

1. Прийміть запрошення до GitHub Academy (якщо ще не зареєстровані).
2. Створіть свій репозиторій для завдання Lab 2 (invite link).
3. Налаштуйте GitHub Actions так, щоб тести запускалися на кожен push та pull request.
4. Виконайте алгоритмічне завдання: реалізацію поставленого завдання та юніт-тести для нього.
5. Додайте файл README.md в якому вкажіть ваш ПІБ, групу, а також інструкції для налаштування середовища і запуску тестів. За цією інструкцією, людина, що перевіряє повинна бути здатна однією командою налаштувати середовище/встановити залежності та іншою - запустити тести.

Критерії оцінювання робіт

Оцінюється якість тестів, а не код, що вирішує задачу. Це означає, що вирішена задача без тестів оцінюється в 0 балів.

Списана робота оцінюється в 0 балів.

Алгоритмічне завдання

Необхідно реалізувати спрощений клавішний калькулятор.

Клавіші калькулятора:

- "0" ... "9" — введення цифри
- "+", "-", "*", "/" — виконання арифметичної операції над цілими числами
- "=" — виконання операції

Для спрощення виконання завдання вводимо наступні обмеження, які відсутні у звичайному калькуляторі:

- обчислення виконуються над цілими числами (int, не float, не double),
- користувач може тільки ввести послідовність цифр, потім операцію, потім знову цифри, потім "=". Будь-яка інша послідовність клавіш є забороненою.

Формат вводу

Вхідний файл містить назви клавіш, які натиснув користувач. Назви клавіш розділені одним символом пробіл.

Формат виводу

Вихідний файл містить одне число — результат обчислень, який відображається на екрані калькулятора.

Приклади вводу та виводу

Input	Output	Пояснення
	0	На початку на екрані відображається нуль.
5	5	Цифри, які вводить користувач, додаються в молодших розрядах числа на екрані.
1 2	12	
1 2 3 + 4 5 6 =	579	Обчислення виконуються за правилами арифметики над цілими числами.
1 2 3 - 2 3 =	100	
1 0 - 1 0 0 =	-90	
1 0 * 2 2 =	220	
1 0 0 / 3 =	33	Обчислення виконуються над цілими числами.
9 / 1 0 =	0	Округлення завжди вниз.
1 2 3 +	123	Користувач ще не почав вводити другий операнд, на екрані все ще відображається перший операнд.
1 2 3 + 4	4	Користувач ще не натиснув клавішу "=", на екрані все ще відображається другий операнд.
1 2 3 + 4 5 6	456	

Приклад плану розв'язання

1. Реалізувати функцію Parse(), яка приймає рядок у описаному вище форматі та розбиває його по пробілу. Результат: масив рядків, кожен з яких є назвою клавіші.

2. Створити клас CalculatorState з наступними полями:

    - screen: число на екрані (тип: ціле число)
    - first_number: перше введене число
    - op: обрана арифметична операція (тип: символ, можливі значення "+", "-", "*", "/")

- start_second_number: прапорець, який встановлюється коли користувач починає вводити нове число

3. Реалізувати функцію HandleKeyPress(), яка приймає об'єкт класу CalculatorState та назву натиснутої клавіші. Функція змінює значення полів об'єкту в залежності від клавіші:

    - "0" ... "9": якщо start_new_number, цифра стає першою цифрою на екрані. Якщо не start_new_number, цифра додається в новий молодший розряд на екран. В будь-якому разі, скинути прапорець start_new_number.
    - "+", "-", "*", "/": записати операцію в last_op, встановити прапорець start_new_number, скопіювати screen в first_number.
    - "=": виконати операцію op над first_number та screen, результат записати в screen.

4. Реалізувати функцію Calculate(), яка приймає масив рядків (назв клавіш) та повертає результат, який буде відображений на екрані калькулятора. Ця функція створює CalculatorState, проходить по масиву рядків та викликає функцію HandleKeyPress().