Технології програмування

**Завдання для РГР**

РГР по дисципліні Технології програмування включає в себе завдання для самостійного виконання, що надаються після кожної теми. Результатом виконання РГР є звіт, що включає в себе програмний код, результати роботи та скріни розміщення коду на github по кожній темі.

Завдання для Теми 1 - **Функції та змінні**

1. Обернути рядок в зворотному порядку.
2. Виконати тестування функцій, що працюють з рядками: strip(), capitalize(), title(), upper(), lower().
3. Написати функцію пошуку дискримінанту квадратного рівняння.

Завдання до Теми 2 – **Умовний перехід**

1. Написати **функцію** пошуку коренів квадратного рівняння використовуючи **функцію** розрахунку дискримінанту з попередньої теми та умовні переходи.
2. Написати програму калькулятор використовуючи **if else** конструкцію. Кожна операція має бути виконана в **окремій** функції.
3. Написати програму калькулятор використовуючи **match** конструкцію. Кожна операція має бути виконана в **окремій** функції.

Завдання до Теми 3 – **Цикли**

1. Розширити програму калькулятор, що була створена до Теми 2, постійними запитами на введення нових даних та операцій. Реалізувати механізм завершення програми після отримання відповідної команди.
2. Написати програму тестування функцій списків таких як: extend(), append(), insert(id, val), remove(val), clear(), sort(), reverse(), copy()
3. Написати програму тестування функцій словників таких як: update(), del(), clear(), keys(), values(), items()
4. Маючи відсортований список, написати функцію пошуку позиції для вставки нового елементу в список.

Завдання до Теми 4 – **Виняткові ситуації**

Попередні умови: реалізована програма калькулятор, що використовує метод нескінченного введення даних для обробки. Всі дії (додавання, віднімання, множення, ділення) реалізовані як окремі функції та використовуються у відповідних місцях. Вказана умова є результатом виконання завдання 1 до Теми 3.

1. Розширити програму калькулятор функцією запитів даних для виконання операцій від користувача, що обробляє виняткові ситуації введення некоректних даних для математичних операцій.
2. Розширити функцію ділення обробкою виняткової ситуації ділення но нуль
3. Ознайомитись зі списком виняткових ситуацій за посиланням <https://docs.python.org/3/library/exceptions.html>

Завдання до Теми 5 – **Бібліотеки**

1. Гра з комп’ютером: камінь, ножиці, папір. Програма виконує запит від користувача на введення одного із значень ["rock", "scissor", "paper"]. Наступним кроком, використовуючи модуль random, програма у випадковому порядку вибирає одне із значень ["rock", "scissor", "paper"]. В залежності від умови, що камінь перемагає ножиці, ножиці перемагають папір, а папір перемагає камінь визначити переможця.
2. Програма конвертування іноземної валюти в українську гривню. Для отримання актуальних курсів валют необхідно використовувати API НБУ та модуль, що надає можливість виконувати запити до сторонніх сервісів requests. Достатня умова роботи – можливість конвертації для трьох іноземних валют EUR, USD, PLN. Користувачу надається можливість введення кількості та типу валюти, результат роботи програми – конвертоване значення в українських гривнях.
3. Використання модулів для програми калькулятор. Функції додавання, віднімання, множення та ділення перенести в файл functions.py. Функції запиту на введення даних для операцій та самих операцій перемістити в файл operations.py. Програму калькулятор реалізувати в файлі calc.py, до якого підключають файл functions.py та operations.py.

Завдання до Теми 6 – **Робота з файлами**

Попередні умови: Реалізована програма калькулятор в файлі calc.py, до якого підключають файл functions.py та operations.py. Інструкції до оформлення вмісту файлів розміщенні в завданні 3 до теми 5.

1. Розробити механізм логування всіх дій, що виконує програма. Забезпечити зберігання інформації про введені данні, виконану операцію та результат виконання операції над даними.
2. Маючи не відсортований список, елементами якого є словники з двома параметрами (ім’я та оцінка) виконати сортування списку, використовуючи стандартну функцію sorted(). Другим параметром для функції sorted() має бути lambda функція, що повертає ім’я або оцінку із елемента словника.

Завдання до Теми 7 – **ООП**

1. Ознайомитись з документацією що описує можливості використання класів у мові Python <https://docs.python.org/3/tutorial/classes.html>
2. Ознайомитись з існуючими за замовченням методами класу по типу \_\_init\_\_(self) \_\_str\_\_(self)\_\_ та надати приклади використання.
3. Розробити клас **Student** атрибутами якого э два параметра **name** та **age**. Створити список елементами якого є об'єкти класу **Student**. Написати цикл який виводить на екран елементи списку у відсортованому порядку. Для сортування використати стандартну функцію **sorted**. Функція **sorted** має використовувати **lambda** функцію для визначення ключа сортування.
4. Використовуючи принципи ООП переписати програму Калькулятор. Завдання має бути виконано використовуючи модульний підхід.