Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського"

КОМП'ЮТЕРНИЙ ПРАКТИКУМ 1.

Реалізація інтерфейсу командного рядка (взаємодія із користувачем) і парсеру команд

Підготувала студентка групи ФБ-34:

Царик Єлизавета Романівна

```
start = True;
print("Accepts commands (case insensitive): CREATE, INSERT INTO, INSERT, SELRCT FROM")
print("Variable names (first character is a letter, other letters/digits/_)")
print("Command is read until ';'")
print("To finish, type 'stop;'\n")
while start:
    print('c: ', end="")
    command = read_until()
    c = command.split()
    analize(c)
```

Початок роботи. Вивід інформації для користувача; Виклик функції для зчитування команд; Розділення рядків за пробілами. Виклик функції для аналізу рядка.

Функція для зчитування команд

```
def read_until():
    command = ""
    while True:
        line = input()
        command = command + line + "\n"
        if ";" in line:
            command = command.split(';', 1)[0]
            break
    return command
```

Записуємо у змінну рядок, при переносі додаємо новий рядок до попереднього доти, доки ";" не з'явиться у рядку. Завершення функції, повернення результату.

Функція для аналізу рядка

```
def analize(c):
    len_c_words = number_of_words(c)
    first = c[0]
    first = first.lower()
    match first:
        case "create":
            if len_c_words < 3:
                print('ERROR: Command does not contain enough tokens.')
                return
            create(c);
            return
        case "insert":
            if len_c_words <  3:
                print('ERROR: Command does not contain enough tokens.')
                return
            second = c[1]
            second = second.lower()
            if (second == "into")and(len_c_words < 4):
                print('ERROR: Command does not contain enough tokens.')
                return
            insert(c)
            return
        case "select":
            if len_c_words <  3:
                print('ERROR: Command does not contain enough tokens.')
                return
            second = c[1]
            second = second.lower()
            if (second != "from"):
                print('ERROR: The command was written incorrectly.')
                return
            select(c)
            return
        case "stop":
            global start
            start = False
            return
        case _:
            print('EROR: Command not recognized.')
            return
```

Викликаємо функцію для підрахунку кількість токенів; Перший токен команди записуємо у змінну і робимо всі літери нижнього регістру (щоб, регістр літер не мав значення); Порівнюємо з create, insert, select, stop, інше;

- Create (перевіряємо мінімальну кількість токенів для цієї команди і викликаємо create())
- Insert (перевіряємо мінімальну кількість токенів для цієї команди (враховуючи і варіант INSERT INTO і просто INSERT і викликаємо insert())

- Select (перевіряємо мінімальну кількість токенів для цієї команди, перевіряємо чи наступним токеном буде from і викликаємо select())
- Stop (змінюємо значення змінної start на false, після цього приймання команд користувача завершиться, як і сама програма)
- Інше (виводимо повідомлення про те, що команда не була розпізнана)

У разі не проходження перевірок зазначених у create, select, insert – будуть виводиться повідомлення, що сповіщатимуть користувача про помилки.

Функція для пошуку кількості токенів

```
def number_of_words(c):
    result = []

    inside_brackets = False
    temp = []

    for word in c:
        if word.startswith("(") and word.endswith(")"):
            result.append(word)
        elif word.startswith("("):
            inside_brackets = True
            temp = [word]
        elif word.endswith(")") and inside_brackets:
            temp.append(word)
            result.append(" ".join(temp))
            inside_brackets = False
        elif inside_brackets:
            temp.append(word)
        else:
            result.append(word)
    return len(result)
```

Основна ідея привести всі функції до вигляду *<команда> змінна (аргументи)* для підрахунку кількості токенів. Тому дана функція об'єднує вираз у дужках в один токен.

Функції create, insert, select

```
def create(c):
    is_var1_correct = is_correct(c[1])
    if not is_var1_correct:
        print('ERROR: variable entered incorrectly.')
        return;

def insert(c):
    return;

def select(c):
    return;
```

Окрім create, вони пусті. Всі вони будуть реалізовані в подальших практичних практикумах.

У функції create() відбувається виклик функції, що перевіряє правильність змінної.

Функція для перевірки змінних

```python
def is_correct(x):
    allowed_all = set("QWERTYUIOPLKJHGFDSAZXCVBNMqwertyuioplkjhgfdsazxcvbnm_1234567890")
    allowed_first = set("QWERTYUIOPLKJHGFDSAZXCVBNMqwertyuioplkjhgfdsazxcvbnm")
    if x[0] not in allowed_first:
        return False
    for ch in x:
        if ch not in allowed_all:
            return False
    return True
```

Вимоги до змінних: вони можуть складатись з латинських літер верхнього та нижнього регістрів, символу “_” та цифр. Однак перша літера змінної має складатись лише з літер.

Перевіряємо виконання вимог і повертаємо True/False.