Державний торговельно-економічний університет  
Кафедра інженерії програмного забезпечення та кібербезпеки

**ЛАБОРАТОРНА РОБОТА №7**

**З ДИСЦИПЛІНИ «АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ»**

**НА ТЕМУ «ЛІНІЙНІ СТРУКТУРИ ДАНИХ»**

**Виконала**: студентка факультету

інформаційних технологій

групи\_курсу 3-4

Авєріна Наталія Ігорівна

**Перевірила**: Палагута К. О.

Київ 2024

**Лабораторна робота №7**

**Тема**: Лінійні структури даних..

**Мета**: навчитись створювати програми для роботи зі стеками, чергами, деками на мові програмування Python.

**Програмне забезпечення**: PyCharm, MS Word.

**1 варіант**

**Хід роботи:**

1. Спочатку був створений клас Stack. Він містить 4 основні методи:
   1. push - додає новий елемент у стек. Якщо кількість елементів досягла максимальної довжини (10), стек очищається перед додаванням нового елементу.
   2. pop - видаляє та повертає останній елемент зі стеку. Якщо стек порожній, повертається None.
   3. back - повертає останній елемент зі стеку, не видаляючи його. Якщо стек порожній, повертається None.
   4. clear - очищує стек, встановлюючи список елементів у порожній список.

Та ще один метод \_\_init\_\_. Він ініціалізує новий стек з порожнім списком елементів.

class Stack:  
 def \_\_init\_\_(self):  
 self.items = []  
   
 def push(self, item):  
 if (len(self.items) >= 10):  
 self.clear()  
 self.items.append(item)  
   
 def pop(self):  
 if not self.items:  
 return None  
 return self.items.pop()  
   
 def back(self):  
 if not self.items:  
 return None  
 return self.items[-1]  
   
 def clear(self):  
 self.items = []

1. Основний блок програми використовує методи класу Stack.
   1. Створюється екземпляр класу
   2. Далі створюється нескінченний цикл обробки команд користувача

if \_\_name\_\_ == '\_\_main\_\_':  
 stack = Stack()  
 while True:  
 command = input('Enter command: ')  
 if command == 'push':  
 stack.push(input('Enter value: '))  
 elif command == 'pop':  
 pop\_value = stack.pop()  
 if pop\_value is not None:  
 print('Pop value:', pop\_value)  
 else:  
 print('Stack is empty!')  
 elif command == 'back':  
 back\_value = stack.back()  
 if back\_value is not None:  
 print('Back value:', back\_value)  
 else:  
 print('Stack is empty!')  
 elif command == 'clear':  
 stack.clear()  
 print('Stack was cleared!')  
 elif command == 'exit':  
 break

**Висновок**: під час виконання даної лабораторної роботи було набуто практичних навичок роботи зі стеками. Реалізовано в програмі методи додавання нового елемента, видалення останнього елемента, виведення останнього елемента та очищення усіх елементів.