МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК

Практическое занятие № 17.

МНОЖЕСТВА И СЛОВАРИ.

Вариант 3.

Выполнил студент:

Герасимов Константин Сергеевич

ПиОА-01.02

**1. Решение задачи 5.1**

Используя тип словарь из пар термин : толкование термина, создать программу для выполнения основных действий с глоссарием терминов. В дополнении к терминам глоссарий должен содержать примеры задач по тематике определенной вариантом (не менее 5 примеров) Пример должен быть записан одной строкой, содержащий для форматирования управляющие последовательности.

***1.1 Постановка задачи***

* создание глоссария (на основе текстового файла)
* сохранение глоссария в файл;

**Дано:**

**Функции:**

def write(filename, glossary): - запись данных в глоссарий

**Входные параметры:**

**Glossary – сам глоссарий.**

**Filename – название текстового файла**

**Выходные параметры:**

f.write(s) – запись в файл строки

* 1. ***Постановка задачи***
* загрузка глоссария из файла;

**Дано:**

**Функции:**

def read(filename, glossary): - функция чтения текстового файла-глоссария

**Входные параметры:**

**Glossary – сам глоссарий.**

**Filename – название текстового файла**

**Выходные параметры:**

glossary[pair[0]] = str(pair[1]) создание строки данных из глоссария

* 1. ***Постановка задачи***
* дополнение глоссария парами термин : толкование термина;

**Дано:**

**Функции:**

def add\_termin(glossary, key: str, value): - функция дополнения глоссария термином.

**Входные параметры:**

**Glossary – сам глоссарий.**

**Key – ключ**

**Value – значение**

**Выходные параметры:**

glossary[key] = value – запись термина

print('Такой термин уже существует в глоссарии. ') – сообщение в случае существования термина

***1.2 Текст программы на языке Python***

def stroking(stroke):

if '.' in stroke:

answerstroke = str()

* 1. ***Постановка задачи***
* удаление из глоссария пар термин : толкование термина;

**Дано:**

**Функции:**

def delete\_termin(glossary, key): - функция удаления термина из глоссария

**Входные параметры:**

**Glossary – сам глоссарий.**

**Key – ключ**

**Выходные параметры:**

del glossary[key] – удаление термина

print('Такого термина нет в глоссарии. ') – сообщение при отсутствии термина

* 1. ***Постановка задачи***
* вывод глоссария на экран;

**Дано:**

**Функции:**

def show\_dict(glossary): - функция вывода глоссария на экран

**Входные параметры:**

**Glossary – сам глоссарий.**

**Выходные параметры:**

print(f'{key}: {str(value)}') – вывод глоссария

* 1. ***Постановка задачи***
* поиск в глоссарии толкования термина и вывод его на экран.

**Дано:**

**Функции:**

def search\_item(glossary, key): - функция поиска термина

**Входные параметры:**

**Glossary – сам глоссарий.**

**Key – ключ**

**Выходные параметры:**

print(f'Термин {key}: {str(glossary[key])} ') – вывод термина на экран

***1.7 Текст программы на языке Python***

glossary = dict()

def read(filename, glossary):

with open(filename, 'r', encoding='utf-8') as f:

for line in f:

line = line.replace('\\n', '\n').replace('\\t', '\t').replace('\\r', '\r') # для замены экранированных символов на всякие отступы и энтеры

line = line.strip() # для удаление всякого мусора

pair = line.split('-')

if len(pair) <= 2:

glossary[pair[0]] = str(pair[1])

def write(filename, glossary):

with open(filename, 'w') as f:

for key, value in glossary.items():

s = key + ':' + value + '\n'

f.write(s)

def add\_termin(glossary, key: str, value):

if glossary.get(key) == None:

glossary[key] = value

else:

print('Такой термин уже существует в глоссарии. ')

def delete\_termin(glossary, key):

if glossary.get(key) != None:

del glossary[key]

else:

print('Такого термина нет в глоссарии. ')

def show\_dict(glossary):

for key, value in glossary.items():

print(f'{key}: {str(value)}')

def search\_item(glossary, key):

print(f'Термин {key}: {str(glossary[key])} ')

def main():

while True:

print('1) Создание/дополнение глоссария на основе файла\n2) Сохранение глоссраия в файл\n3) Дополнение глоссария термином\n4) Удаление термина из глоссария\n5) Вывод глоссария на экран\n6) Поиск термина и его вывод на экран\n7) Выход.')

answer = input()

if answer == '1':

read('termins.txt', glossary)

if answer == '2':

write('termins.txt', glossary)

if answer == '3':

add\_termin(glossary, input('Введите термин '), input('Введите толкование '))

if answer == '4':

delete\_termin(glossary, input('Введите термин, который нужно удалить '))

if answer == '5':

show\_dict(glossary)

if answer == '6':

search\_item(glossary, input('Введите название термина: '))

if answer == '7':

break

main()

***1.3 Результат тестирования программы на языке Python***

***Изображение выглядит как текст

Автоматически созданное описание***

***Изображение выглядит как текст

Автоматически созданное описание***

**2. Решение задачи 5.2**

***1.1 Постановка задачи***

Дана строка символов из n слов произвольной длины (длина слова не превышает 80 символов). Символами могут быть только заглавные латинские буквы. Найти и вывести все глухие согласные буквы (без повторений), которые встретились в двух самых коротких словах.

**Дано:**

**Функции:**

def utin(string: str): - функция нахождения глухих согласных

**Входные параметры:**

**String = input() – обрабатываемая строка**

**Начальные значения:**

words = {'T', 'S', 'H', 'F', 'K', 'X', 'C', 'P'} – множество глухих согласных букв

ans = set() – итоговое множество

flag = False – флаг, сигнализирующий, верная ли строка введена

**Найти:**

Найти и вывести все глухие согласные буквы (без повторений), которые встретились в двух самых коротких словах.

**Выходные параметры:**

print(\*listik[0], \*listik[1]) – вывод первых двух коротких слов и их глухих букв

print('Вы ввели неверное предложение! ')

print('Вы ввели неверное предложение! ') – сообщение при неверной введенной строке

***1.2 Текст программы на языке Python***

def utin(string: str):

words = {'T', 'S', 'H', 'F', 'K', 'X', 'C', 'P'}

listik = string.split()

ans = set()

for i in range(len(listik)):

listik[i] = set(listik[i])

listik.sort(key = lambda s: len(s))

for i in range(len(listik)):

listik[i] = listik[i] & words

print(\*listik[0], \*listik[1])

print('Введите строку, состоящую из слов, состоящих только из заглавных латинских букв: ')

flag = False

while flag == False:

centence = input()

if centence.isupper() == False:

cent = centence.split()

for i in range(len(cent)):

if len(cent[i]) > 80:

print('Вы ввели неверное предложение! ')

print('Вы ввели неверное предложение! ')

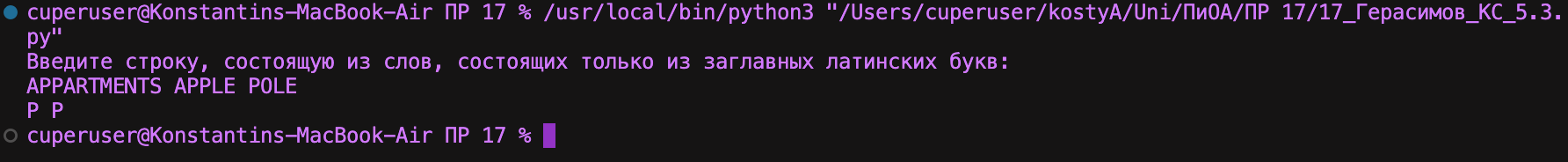
else:

flag = True

break

utin(centence)

***1.3 Результат тестирования программы на языке Python***

****