**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**УНИВЕРСИТЕТ ИТМО»**

**ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**Отчет**

**по Лабораторной работе №3**

**«Словари»**

Специальность 09.02.07«Информационные системы и программирование»

Дисциплина «Разработка программных модулей»

|  |  |
| --- | --- |
| Преподаватель:  Закарая Борис  «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_202\_\_г.  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Выполнил:  студент группы Y2334  Панаёт В. Т. |

Санкт-Петербург

2020/2021

**Цель лабораторной работы:** изучить тип данных dict

**Задание:**

Реализуйте программу, которая будет эмулировать работу с пространствами имен. Необходимо реализовать поддержку создания пространств имен и добавление в них переменных.

В данной задаче у каждого пространства имен есть уникальный текстовый идентификатор – его имя.

Вашей программе на вход подаются следующие запросы:

* **create <namespace> <parent>**–  создать новое пространство имен с именем **<namespace>**внутри пространства **<parent>**
* **add <namespace> <var>**– добавить в пространство **<namespace>**переменную **<var>**
* **get <namespace> <var>**– получить имя пространства, из которого будет взята переменная **<var>**при запросе из пространства **<namespace>**, или **None**, если такого пространства не существует

**Формат входных данных**

В первой строке дано число **n** (**1 ≤ n ≤ 100**) – число запросов.  
В каждой из следующих **n** строк дано по одному запросу.  
Запросы выполняются в порядке, в котором они даны во входных данных.  
Имена пространства имен и имена переменных представляют из себя строки длины не более **10**, состоящие из строчных латинских букв.

**Формат выходных данных**

Для каждого запроса **get** выведите в отдельной строке его результат.

**Листинг программы:**

n = int(input())

a = {'global': {'vars' :

    [], 'namespaces' :

        [],}}

def add(namespace, var):

    if var not in a[namespace]['vars']:

        a[namespace]['vars'].append(var)

def create(namespace, parent):

    if namespace not in a:

        a[namespace] = {}

        a[namespace]['vars'] = []

        a[namespace]['namespaces'] = []

        a[namespace]['parent'] = parent

        a[parent]['namespaces'].append(namespace)

def get(namespace, var):

    for i in a[namespace]['vars']:

        if var == i :

            return namespace

    if namespace == 'global':

        return None

    return get(a[namespace]['parent'], var)

for i in range(n):

    com = input().split()

    if com[0] == 'add':

        add(com[1], com[2])

    if com[0] == 'create':

        create(com[1], com[2])

    if com[0] == 'get':

        print(get(com[1], com[2]))

Пример работы программы:

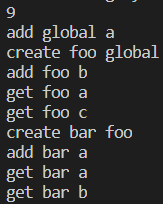


Рис. 1. Входные данные



Рис. 2 Результат исполнения программы

**Вывод:** в ходе выполнения лабораторной работы был изучен тип данных dictionary.