

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

**Основы кроссплатформенного программирования
Отчет по лабораторной работе 2.12**

Декораторы функций в языке Python

Выполнил студент группы
ИТС-б-о-21-1

Плещенко Данила Георгиевич
« » _____ 2022г.

Подпись студента _____

Работа защищена « » _____ 2022г.

Проверил к.т.н., доцент
Кафедры инфокоммуникаций
Воронкин Роман Александрович

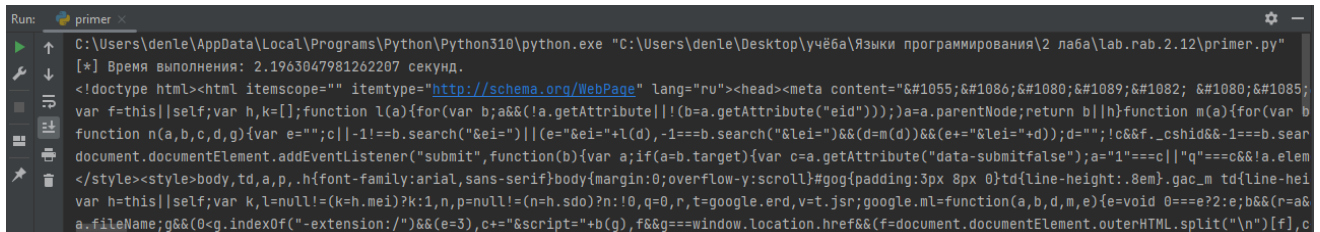
(подпись)

Цель работы: приобретение навыков по работе с декораторами функций при написании программ с помощью языка программирования Python версии 3.x.

Ссылка на репозиторий - [danilaple/lab.rab.2.12 \(github.com\)](https://github.com/danilaple/lab.rab.2.12)

Ход работы:

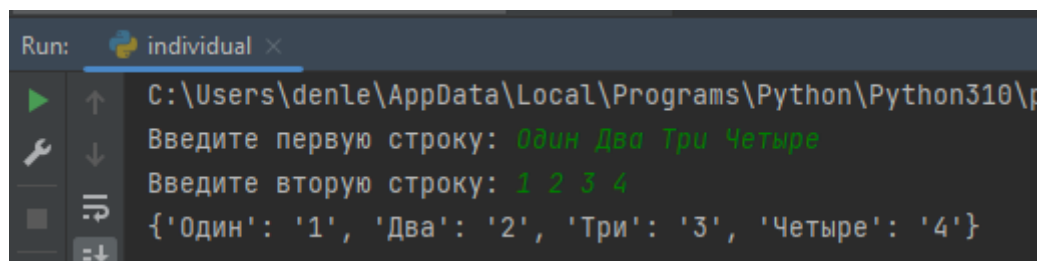
Пример 1. Создаём декоратор, измеряющий время выполнения функции с использованием конструкции `*args` и `**kwargs`.



```
Run: primer
C:\Users\denle\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\denle\Desktop\учёба\Языки программирования\2 лаба\lab.rab.2.12\primer.py"
[*] Время выполнения: 2.1963047981262207 секунд.
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="ru"><head><meta content="&#1055;&#1086;&#1080;&#1089;&#1082; &#1080;&#1085;
var f=this||self;var h,k=[];function l(a){for(var b;a&&(a.getAttribute||(b=a.getAttribute("eid")));a=a.parentNode;return b||h}function m(a){for(var b
function n(a,b,c,d,g){var e="";c||-1==b.search("&ei=")||e="&ei="+l(d),-1==b.search("&lei=")&&(d=m(d))&&(e+="&lei="+d));d="";!c&&f._cshid&&-1==b.sear
document.documentElement.addEventListener("submit",function(b){var a;if(a=b.target){var c=a.getAttribute("data-submitfalse");a="1"==c||"q"==c&&!a.elem
</style><style>body,td,a,p,.h{font-family:arial,sans-serif}body{margin:0;overflow-y:scroll}#gog{padding:3px 8px 0}td{line-height:.8em}.gac_m td{line-hei
var h=this||self;var k,l=null!==(k=h.mei)?k:1,n,p=null!==(n=h.sdo)?n:!0,q=0,r,t=google.erd,v=t.jsr;google.ml=function(a,b,d,m,e){e=void 0===e?2:e;b&&(r=a&
a.fileName;g&&(0<g.index0f("-extension:")&&(e=3),c+="&script="+b(g),f&&g==window.location.href&&(f=document.documentElement.outerHTML.split("\n")[f],c
```

Рисунок 1. Результат выполнения программы

Индивидуальное задание. 13 вариант. Вводятся два списка (каждый с новой строки) из слов, записанных через пробел. Имеется функция, которая преобразовывает эти две строки в два списка слов и возвращает эти списки. Определите декоратор для этой функции, который из этих двух списков формирует словарь, в котором ключами являются слова из первого списка, а значениями — соответствующие элементы из второго списка. Полученный словарь должен возвращаться при вызове декоратора. Примените декоратор к первой функции и вызовите ее. Результат (словарь) отобразите на экране.



```
Run: individual
C:\Users\denle\AppData\Local\Programs\Python\Python310\p
Введите первую строку: Один Два Три Четыре
Введите вторую строку: 1 2 3 4
{'Один': '1', 'Два': '2', 'Три': '3', 'Четыре': '4'}
```

Рисунок 2. Результат выполнения программы

Контрольные вопросы:

1. Что такое декоратор?

Декоратор — это функция, которая позволяет обернуть другую функцию для расширения её функциональности без непосредственного изменения её кода.

2. Почему функции являются объектами первого класса?

Объектами первого класса в контексте конкретного языка программирования называются элементы, с которыми можно делать всё то же, что и с любым другим объектом: передавать как параметр, возвращать из функции и присваивать переменной.

В Python всё является объектом, а не только объекты, которые вы создаёте из классов. Это значит, что в Python всё это — объекты:

- числа;
- строки;
- классы;
- функции.

3. Каково назначение функций высших порядков?

Функции высших порядков — это такие функции, которые могут принимать в качестве аргументов и возвращать другие функции.

4. Как работают декораторы?

Декоратор — это функция, которая позволяет обернуть другую функцию с помощью символа «@»

5. Какова структура декоратора функций?

Сначала записывается функция — декоратор. Потом идет его вызов с помощью @, а затем основная функция, которую оборачивает декоратор.

6. Самостоятельно изучить, как можно передать параметры декоратору, а не декорируемой функции?

Используя замыкание функций.

Вывод: в ходе лабораторной работы были приобретены навыки по работе с функциями с переменным числом параметров при написании программ с помощью языка программирования Python версии 3.x.