ФУНКЦИИ В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON

Тилляев Диёрбек Улугбекович
студент, кафедра экономики транспорта,
Ташкентский государственный транспортный университет,
РУз, г. Ташкент
E-mail: diyorbektillyayev@gmail.com
Уралов Нодир Бохадирович
научный руководитель, ассистент кафедры,
Ташкентский государственный транспортный университет,
РУз, г. Ташкент
E-mail: uralovnod@gmail.com

FUNCTIONS IN PYTHON PROGRAMMING LANGUAGE

Diyorbek Tillyayev
Student, Department of Transport Economics,
Tashkent State Transport University,
Uzbekistan, Tashkent
Nodir Uralov
Scientific supervisor, assistant of the Department IT,
Tashkent State Transport University,
Uzbekistan, Tashkent

АННОТАЦИЯ

В данной статье рассматриваются функции в языке программирования Python. Функции — основные инструменты для структурирования кода и повышения его повторного использования. Описаны ключевые особенности функций, методы их создания и применения. Особое внимание уделяется различным типам функций: встроенным и пользовательским, а также механизмам передачи аргументов и возвращаемых значений. Обсуждаются вопросы области видимости переменных, рекурсия и использование анонимных функций (lambda). Приведены примеры практического применения функций для решения типичных задач. Исследуются преимущества и недостатки различных подходов к определению функций, включая стандартные параметры, произвольное количество аргументов и именованные аргументы. Обсуждаются концепции замыканий, функций высшего порядка и декораторов, которые помогают создавать более гибкие и эффективные решения. Эти темы помогают лучше понять функции и их важность в программировании на Python.

ABSTRACT

This article discusses functions in the Python programming language. Functions are the main tools for structuring code and enhancing its reuse. The key features of functions, methods of their creation and application are described. Special attention is paid to different types of functions: built-in and user, as well as mechanisms for the transmission of arguments and returned values. The discussion concerns the visibility of variables, recursion and use of anonymous functions (lambda). Examples of practical application of functions to typical tasks are given. The advantages and disadvantages of different approaches to function definition are investigated, including standard parameters, arbitrary number of arguments and named arguments. Concepts of closure, higher order functions and decorators are discussed that help create more flexible and effective solutions. These topics help to better understand functions and their importance in Python programming.

Ключевые слова: Python, функция, программирование, структура кода, повторное использование, дублирование, параметры, аргументы.

Keywords: Python, function, programming, code structure, reuse, duplication, parameters, arguments.

ВВЕДЕНИЕ

Функции играют ключевую роль в любом современном языке программирования, в том числе и в Python. Они позволяют разделить основной код на логически связанные блоки, что упрощает разработку, тестирование и поддержку программного обеспечения. Основная цель функций – обеспечить повторное использование кода и сократить его повторное использование или дублирование. В языке Python функции представляют собой не только важный элемент синтаксиса, но и мощный инструмент для решения задач, связанных с эффективной организацией кода. Python предоставляет широкие возможности для создания функций, включая поддержку различных типов аргументов, возврата значений и использования анонимных функций. Функции в Python могут быть использованы не только для упрощения кода, но и для реализации более сложных концепций, таких как замыкания, декораторы, функции высшего порядка и рекурсия. Данная статья направлена на изучение различных аспектов использования функций в Python, включая основные типы функций, способы их создания и применения, а также ключевые концепции, такие как область видимости, передача параметров, использование анонимных функций и принципы модульного тестирования. Также в статье исследуются более сложные техники, включая функции высшего порядка и декораторы.

ЛИТЕРАТУРА И МЕТОДЫ

Для исследования были использованы книги и онлайн-документация, такие как “Python Documentation”, книга уроков от сайта devpractice.ru (Абдрахманов М.И., 2019), Дрянкова Дарья Александровна, Замулин Иван Сергеевич «Лямбда-функции в языке программирования Python», Downey, A. (2015). *Think Python: How to Think Like a Computer Scientist*. O'Reilly Media. и учебные материалы на платформе “GeeksforGeeks”. В числе используемых источников — официальная документация Python, учебники по программированию и статьи, посвященные современным методам разработки на Python. Также рассматривались научные работы, анализирующие принципы функционального программирования, а также работы, касающиеся оптимизации и тестирования кода. Анализировались теоретические аспекты функций, включая их определение, использование параметров, возвращаемых значений и области видимости переменных.

Метод исследования включал написание и тестирование программного кода для иллюстрации основных концепций. Для выполнения задачи был использован подход, включающий создание и анализ различных типов функций, таких как стандартные, анонимные (lambda), а также функции высшего порядка и декораторы. Все примеры кода были реализованы в среде разработки Python, что обеспечивало возможность интерактивного тестирования и отладки. В рамках исследования также применялись методы рефакторинга кода, направленные на улучшение структуры и производительности программ, а также принципы модульного тестирования с использованием библиотеки **unittest**, что позволило оценить правильность работы реализованных функций. Для более глубокого понимания особенностей работы функций в Python были проведены эксперименты, включающие использование различных типов аргументов (позиционных, именованных, произвольного количества) и анализ их влияния на эффективность и читаемость кода.

ОБСУЖДЕНИЕ

Функции в Python определяются с помощью ключевого слова *def*.

Например:

```
def greet(name):
return f"Hello, {name}!"
```

Функции могут иметь параметры и аргументы, которые передаются при вызове. Существует возможность указания значений по умолчанию, что позволяет гибко управлять их поведением.

Важным аспектом является область видимости переменных. Переменные внутри функции локальны, что помогает избежать конфликтов с переменными

из глобальной области. Также функции могут быть вложенными, что расширяет их функциональность.

РЕЗУЛЬТАТЫ

В результате анализа и тестирования кода были выделены следующие ключевые преимущества использования функций:

1) Повышение читаемости и структурирования кода.
2) Уменьшение количества дублирующего кода.
3) Улучшение тестируемости и модифицируемости программного обеспечения.

Пример функции для вычисления факториала:

```
def factorial(n):
if n == 0:
return 1
else:
return n * factorial(n-1)
```

Список литературы:

1) “Python: Уроки” второе издание, стр.143-147 Абдурахмонов М.И., 2019. URL: https://devpractice.ru/python-lessons/
2) “Лямбда-функции в языке программирования Python” Дрянкова Дарья Александровна, н.у: Замулин Иван Сергеевич., 2023
3) “Использование рекурсивных функций для решения сложных задач в Python” Оразова О.Б., Аннамаммедов С.Д., 2023