

ИЗОБРАЖЕНИЯ И ВИЗУАЛИЗАЦИЯ ДАННЫХ В PYTHON

Красочкин С.Г.

*Красочкин Сергей Геннадьевич - ведущий разработчик,
Компания SberCloud, г. Москва*

Аннотация: в статье рассматривается целесообразность и особенности применения средств визуализации приложений в процессе обучения программированию на языке Python. Рассмотрены примеры применения онлайн-сервиса Pythontutor для визуализации особенностей синтаксиса языка.

Ключевые слова: обучение, программирование, визуализация.

В последнее время объемы обрабатываемых данных в различных областях знаний стали очень большими и требуют различных методов обработки изображений в рамках инструментов Python для работы с изображениями.

Специалисты в сфере IT всегда ищут кроссплатформенный, бесплатный, простой в освоении язык, с отличной поддержкой сообщества разработчиков и хорошими встроенными функциями, которые могли бы помочь нам быстро разрабатывать свои приложения. Если задаться вопросом, существует ли такой язык?

Python - один из самых популярных языков программирования общего назначения. Он входит в число самых быстрорастущих языков программирования в мире и используется инженерами-программистами, математиками, аналитиками данных, учеными, сетевыми инженерами, студентами и бухгалтерами. Рассмотрим функции, которые делают Python таким мощным языком. Это интерпретируемый, объектно-ориентированный и высокоуровневый язык программирования. Python называется интерпретируемым языком, поскольку его исходный код компилируется в байт-код, который затем интерпретируется. CPython обычно компилирует код Python в байт-код перед его интерпретацией.

Он поддерживает динамическую типизацию и динамическую привязку. В таких языках, как Java, C и C++, вы не можете инициализировать строковое значение переменной `int`, и в таких случаях программа не будет компилироваться. Python не знает тип переменной до тех пор, пока код не будет выполнен.

В языке Python существует много различных библиотек и пакетов для работы с изображениями. Все они позволяют проводить множество самых различных преобразований с исходным изображением.

Целью данной работы является исследование и применение различных библиотек и алгоритмов обработки изображений на языке программирования Python.

Рассмотрим визуализацию некоторых алгоритмических конструкций и типов данных языка программирования Python, применив онлайн-приложение Pythontutor [2]. С помощью данного онлайн-приложения можно проиллюстрировать очерёдность выполнения команд и поток данных, которые изменяются в результате выполнения операторов. По мере пошагового выполнения программы, онлайн-приложение отображает значения переменных как базовых типов (`int`, `float`, `str`) так и ссылочных типов (`list`, `dict`, `tuple` и др.). Значения, хранящиеся в переменных последнего типа, показаны отдельно, а все переменные, которые ссылаются на эти данные, показаны стрелочками.

В языке Python такими библиотеками являются:

- Пакет `skikit-image`;
- Библиотека `NumPy` как инструмент работы с массивами;
- Модуль `SkiPy` для решения задач по обработке изображений;
- Библиотека `PIL` (`PythonImageLibrary`) и `Pillow`, поддерживающие базовый функционал по обработке изображений;
- Библиотека `OpenCV` (`Open Source Computer Vision Library`) для приложений по компьютерному зрению;
- Фреймворк `SimpleCV` с открытым кодом для создания приложений по компьютерному зрению [1].

Все библиотеки активно взаимодействуют с массивами `NumPy`, т.к. изображения представляются данным типом данных [2]. Библиотеки имеют функции открытия и записи в файл. Большинство библиотек имеет собственный набор тестовых изображений. Все библиотеки имеют методы обработки изображений.

Обработка изображения - метод изменения изображения для получения информации из изображения или улучшения качества изображения и выполнения над ним определённой заданной функции, например, фильтрации по определённому алгоритму и бинаризации изображения [3]. Обработка изображений использует математические алгоритмы работы с матрицами как способом представления точек на плоскости [4].

Библиотеки языка Python имеют встроенные методы работы с изображениями, рассматриваемыми как массив `NumPy`.

Выбор той или иной библиотеки зависит от решаемой задачи. Например, для задач компьютерного зрения используются библиотеки `OpenCV` и `SimpleCV` [5].

`Viewer` Простой графический пользовательский интерфейс для визуализации результатов и изучения

параметров Их использование обеспечивается вызовом соответствующей функции пакета [12]. В других библиотеках реализованы схожие функции. Вызов функций соответствующей библиотеки возможен после её указания в команде `import`.

Пакет `skikit-image` представляет собой пакет с открытым исходным кодом. Данный пакет работает с массивами `NumPy/Python`. Используется в научно-исследовательских, промышленных и прикладных программах

Библиотека `NumPy` как инструмент работы с массивами. `NumPy` - мощная библиотека `Python` с поддержкой массивов [14]. В данной библиотеке реализовано множество возможностей, которые отсутствуют при работе с простыми списками [12]. В языке `Python` любое изображение представляется массивом `NumPy`, определяющим пиксели изображения.

Наследование, наряду с композицией является основой построения комплексных приложений. В примере, представленном на рисунке 1, описаны два класса - `Point` и его класс-наследник `LabeledPoint` [5]. На изображении показан один из шагов создания экземпляра класса `LabeledPoint` и последовательность вызовов конструкторов классов (методы `init`). При пошаговом выполнении примера, можно проследить за тем, как с помощью функции `super` происходит обращение к методам родительского класса и на какие объекты ссылается параметр `self` в каждый момент времени.

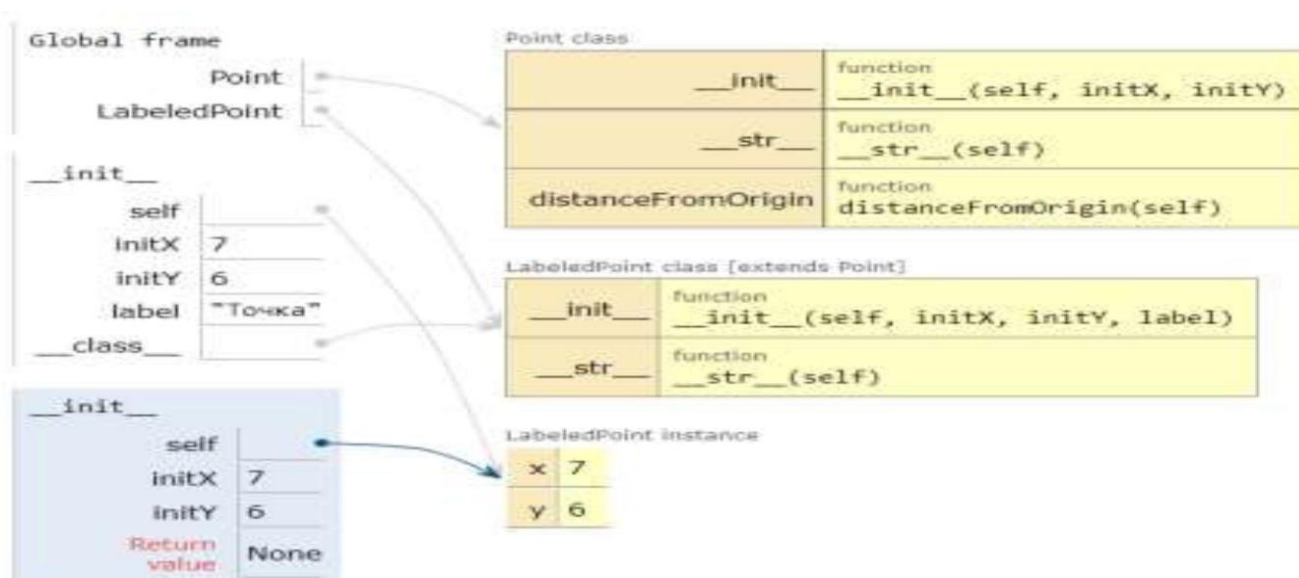


Рис. 1. Пример наследования классов

В случае с композицией, одно из полей класса ссылается на экземпляр другого класса. Помимо иллюстрации основных синтаксических конструкций `Python`, визуализацию работы программы можно применить в ряде других случаев: демонстрация рекурсии, визуализация алгоритмов (нахождение НОД, численных методов и т.д.), визуализация различных структур данных (стек, очередь, и др.), демонстрация более сложных синтаксических конструкций (исключения, `lambda`- функции, декораторы, генераторы и т. д.).

Модуль `SciPy` для решения задач по обработке изображений. `SciPy` - это очень часто используемый модуль в языке `Python`, как и `NumPy` [16]. Он используется для решения задач по обработке изображений.

Например, в подмодуле `scipy.ndimage` есть функции, работающие в многомерных массивах `NumPy`. Пакет включает в себя функции для бинарной морфологии, линейной и нелинейной фильтрации, интерполяции.

Таким образом, инструмент визуализации `Python`, помимо иллюстрации основных синтаксических конструкций `Python`, визуализацию работы программы можно применить в ряде других случаев: демонстрация рекурсии, визуализация алгоритмов (нахождение НОД, численных методов и т.д.), визуализация различных структур данных (стек, очередь, и др.), демонстрация более сложных синтаксических конструкций (исключения, `lambda`- функции, декораторы, генераторы и т.д.).

Список литературы / References

1. Чорна А., Таганова Д. Використання алгоритмів розв'язування графічних задач засобами `PyShop` // Інформаційні технології в освіті та науці зб. наук.пр., 2018. № 10. С. 310—315.
2. Кабаков Р. Р. в действии. Анализ и визуализация данных на языке `R`. Litres, 2017. С. 209—213.
3. Маккинни У. `Python` и анализ данных/УэсМаккинни; [пер. с англ. А. Слинкин] // М.: ДМК Пресс, 2015. С. 482.

4. *Бэрри П.* Изучаем программирование на Python. Litres, 2019. С. 332.
5. *Васильев А.* Python на примерах. Практический курс по программированию, 2016.
6. Цифровая обработка сигналов и изображений в радиофизических приложениях / М. Басараб, В. Волосюк, О. Горячкин, А. Зеленский, В. Кравченко, А. Ксендзук, Б. Кутуза, В. Лукин // Кравченко ВФ-М.: ФИЗМАТЛИТ. — 2007.
7. *Васильев А.* Python на примерах, практический курс // [Электронный ресурс]. Режим доступа: <https://pythonworld.ru/bookshop/61.html>. (дата обращения: 20.04.2021), 2017.
8. *Гуриков С.* Основы алгоритмизации и программирования на Python // М.: «Форум», 2018. Т. 343.
9. *Златопольский Д.* Основы программирования на языке Python. Litres, 2019.
10. *Гэддис Т.* Начинаем программировать на Python.-4-е изд. // Пер. с англ./ Т. Гэддис-СПб.: БХВ-Петербург, 2019.
11. *Свейгарт Э.* Автоматизация рутинных задач с помощью Python: практическое руководство для начинающих // М.: ООО «ИД Вильямс», 2017.
12. *Беляев Е.А., Сухов Т.М., Шостацкий Н.Н.* Сжатие видеоинформации на основе трехмерного дискретного псевдо-косинусного преобразования для энергоэффективных систем видеонаблюдения // Компьютерная оптика, 2010. Т. 34, № 2.
13. *Любанович Б.* Простой Python. Современный стиль программирования. "Издательский дом Питер", 2016.
14. *Акулич В.* Инструменты работы с Bigdata в социальных науках: учебная программа учреждения высшего образования по учебной дисциплине для специальности 1-25 80 01 «Экономика» (профилизация «Экономическая социология»), 2021.