УДК 004

Разработка программы автоматического добавления логотипа на изображение с помощью языка Python

Кизянов Антон Олегович
Приамурский государственный университет имени Шолом-Алейхема
студент

Аннотация
В данной статье продемонстрирован процесс создания программы автоматического добавления логотипа на изображение.
Ключевые слова: Python, логотип

Development programs automatically add the logo image using the Python language

Kizyanov Anton Olegovic
Sholom-Aleichem Priamursky State University
Student

Abstract
This article demonstrated the process of creating a program to automatically add a logo to the image.
Keywords: Python, logo

Предположим, предстоит рутинная работа по изменению размеров тысячи изображений и добавление в углу каждого из них небольшого логотипа в виде водяного знака. Выполнение такой задачи с помощью простых графических программ наподобие Paintbrush или Paint длилось бы целую вечность. В более сложных графических приложениях, таких как Photoshop, существует возможность пакетной обработки, но такое программное обеспечение стоит очень дорого.

Для ознакомления с языком программирования Python прочтите следующие статьи. В.А.Машков, В.И.Литвиненко рассказали о применение языка программирования python для решения задач самодиагностики на системном уровне [1]. Г.Д.Бухарова, и П.С.Комельских рассказали о важность и необходимости внедрения языка программирования Python в процесс обучения студентов [2]. Г.С.Сейдаметов продемонстрировал особенности использования языка программирования python в подготовке будущих инженеров-программистов [3]. Э.А.Усеинов продемонстрировал использование объектно-ориентированного программирования в рамках дисциплины «язык программирования python» [4].

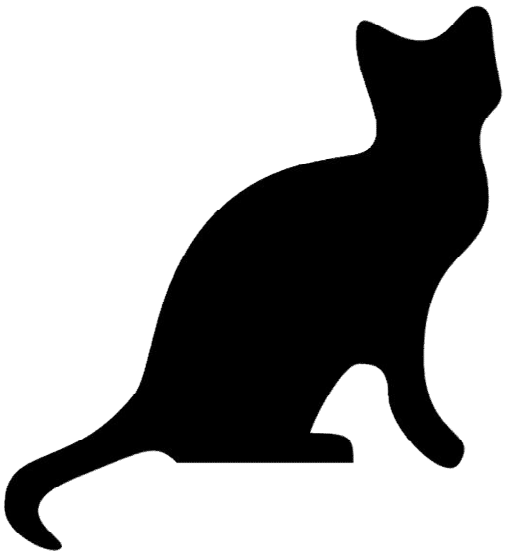

Рис. 1

На рисунке 1 показан логотип, который мы будем добавлять в нижний правый угол каждого изображения. Это логотип представляет собой стилизованный профиль кошки с белым контуром и прозрачной остальной частью изображения.

```
import os
from PIL import Image

SQUARE_FIT_SIZE = 300
LOGO_FILENAME = 'catlogo.png'

logoIm = Image.open(LOGO_FILENAME)
logoWidth, logoHeight = logoIm.size
```

Рис. 2

Задав в начале программы значения констант SQUARE_FIT_SIZEи LOGO_FILENAME, мы упростили внесение возможных изменений в программу в будущем. Предложим, вы захотите использовать в качестве логотипа другой рисунок или ограничить размеры логотипа не 300 пикселями, а другой величиной. Расположив определения этих констант в самом начале программы, вы должны будете внести изменения, если это потребуется, только в одном месте программы. Без использования этих констант вам пришлось бы просматривать весь код в поиске всех вхождений значений 300 и ‘catlogo.png’ и заменять их вручную для каждого нового проекта. Короче говоря, константы делают программу более универсальной.

Объек Image логотипа возвращается вызовом функции Image.open(). Для улучшения удобочитаемости кода программы значения, содержащиеся в атрибуте logoIm.size, присваиваются отдельным переменным logoWidthи logoHeight.

Теперь мы должны организовать в текущем рабочем каталоге последовательный поиск всех .png и jpg файлов. При этом следует учесть, что в добавлении изображения логотипа к самому логотипу нет никакой необходимости, и поэтому программа должна пропускать любое изображение с тем же именем файла, которое содержится в константе LOGO_FILENAME.

```
for filename in os.listdir('.'):
    if not (filename.endswith('.png') or filename.endswith('.jpg')) \
       or filename == LOGO_FILENAME:
        continue

    im = Image.open(filename)
    width, height = im.size
```

Рис. 3

Прежде всего, мы создаем с помощью вызова os.makedirs() отдельную папку withLogo, предназначенную для хранения версий изображений с логотипами, чтобы не затирать исходные файлы. Указав именованный аргумент exist_ok=True, можно избежать возбуждения исключений в методе os.makedirs() в том случае, если папка withLogo уже существует. В процессе выполнения цикла по всем файлам текущего рабочего каталога с использованием вызова os.listdir(‘.’) длинная инструкция if проверяет расширение имени каждого файла. Если файл имеет расширение .pngили .jpgили же если это файл самого изображения, то цикл должен пропустить его и использовать инструкциюcontinueдля перехода к следующему файлу. Если же имя заканчивается расширением .pngили .jpg, то можно открыть его в виде объектаImageи задать ширину и высоту изображения.

Программа должна изменить размеры изображений лишь в том случае, если ширина или высота превышает значение, определяемое константой SQUARE_FIT_SIZE, поэтому необходимый для этого код следует поместить в инструкцию if, проверяющую значения переменных widthи height.

```
if width > SQUARE_FIT_SIZE and height > SQUARE_FIT_SIZE:
    if width > height:
        height = int((SQUARE_FIT_SIZE / width) * height)
        width = SQUARE_FIT_SIZE
    else:
        width = int((SQUARE_FIT_SIZE / height) * width)
        height = SQUARE_FIT_SIZE

    print('Resizing %s...' % (filename))
    im = im.resize((width, height))
```

Рис. 4

Если размеры изображения действительно надо изменить, то в том случае следует определить, какой именно из размеров превышает допустимый передел – ширина или высота. Если ширина изображения больше его высоты, то следует уменьшить в той же пропорции, что и

ширину. Величина коэффициента пропорциональности находится делением значения SQUARE_FIT_SIZEна текущее значениеширины. Тогда новое значение высоты будет равно её текущему значению, умноженному нанайденное значение коэффициента пропорциональности. Поскольку результатом операции деления является вещественное число, метод resize() требует задания целочисленных размеров, не забудьте преобразовать полученный результат в целое число с помощью функции int(). Наконец, новое значение ширины просто устанавливается равным SQUARE_FIT_SIZE.

Случай, когда высота изображения больше ширины или равна ей, обрабатывается с использование такой же расчетной схемы, за исключением того, что переменные heightи widthменяются ролями.

Установив в переменных widthи heightновые значения размеров, передайте их методу resize() и сохраните возвращаемый объектImageв переменной im.

Независимо от того, изменялись ли размеры изображения, логотип должен помещаться в нижний правый угол изображения. В какую именно позицию он должен вставляться, определяется размерами как изображения, так и самого логотипа.

После того как наш код вставит логотип в изображение, он должен сохранить измененный объект Image.

```
print('Adding logo to %s...' % (filename))
im.paste(logoIm, (width - logoWidth, height - logoHeight), logoIm)

im.save(os.path.join('withLogo', filename))
```

Рис. 5

Новый код выводит сообщение, которое извещает пользователя о добавлении логотипа, помещает изображение logoIm в позицию с рассчитанными координатами и сохраняет изменения в файле в каталоге withLogo.

```
import os
from PIL import Image

SQUARE_FIT_SIZE = 300
LOGO_FILENAME = 'catlogo.png'

logoIm = Image.open(LOGO_FILENAME)
logoWidth, logoHeight = logoIm.size

os.makedirs('withLogo', exist_ok=True)
for filename in os.listdir('.'):
    if not (filename.endswith('.png') or filename.endswith('.jpg')) \
       or filename == LOGO_FILENAME:
        continue

    im = Image.open(filename)
    width, height = im.size

    if width > SQUARE_FIT_SIZE and height > SQUARE_FIT_SIZE:
        if width > height:
            height = int((SQUARE_FIT_SIZE / width) * height)
            width = SQUARE_FIT_SIZE
        else:
            width = int((SQUARE_FIT_SIZE / height) * width)
            height = SQUARE_FIT_SIZE

        print('Resizing %s...' % (filename))
        im = im.resize((width, height))

    print('Adding logo to %s...' % (filename))
    im.paste(logoIm, (width - logoWidth, height - logoHeight), logoIm)

    im.save(os.path.join('withLogo', filename))
```

Рис. 6

На рисунке 6 представлен полный код программы.

Вывод: Написали программу автоматической генерации случайных билетов неограниченногоколичества вариантов и ключей к ним.

Библиографический список

1. Машков В.А., Литвиненко В.И. Использование языка программирования python 3 и системы компьютерной алгебры sympy на факультативных занятиях по теории чисел // В сборнике: Электротехнические и компьютерные системы Издательство: Одесский национальный политехнический университет (Одесса)С. 48-54 [Электронный ресурс]. URL: http://elibrary.ru/item.asp?id=23422667 (дата обращения: 25.01.2017)
2. Бухарова Г.Д., Комельских П.С. Важность и необходимость внедрения языка программирования python в процесс обучения студентов // В сборнике: новые информационные технологии в образовании Материалы VII международной научно-практической конференции. Российский государственный профессионально-педагогический университет. 2014 Издательство: Российский государственный профессионально-

педагогический университет (Екатеринбург) С. 40-42. [Электронный ресурс]. URL: http://elibrary.ru/item.asp?id=22278620 (дата обращения: 25.01.2017)
3. Сейдаметов Г.С. Особенности использования языка программирования python в подготовке будущих инженеров-программистов // В сборнике: INTERNATIONAL SCIENTIFIC REVIEW Издательство: Олимп (Иваново) С. 50-51 [Электронный ресурс]. URL: http://elibrary.ru/item.asp?id=24983350 (дата обращения: 25.01.2017)
4. Усеинов Э.А. объектно-ориентированное программирование в рамках дисциплины «язык программирования python» // В сборнике: ученые записки крымского инженерно-педагогического университета Издательство: Государственное бюджетное образовательное учреждение высшего образования Республики Крым «Крымский инженерно-педагогический университет» (Симферополь) С. 157-160. [Электронный ресурс]. URL: http://elibrary.ru/item.asp?id=24836776 (дата обращения: 25.01.2017)