

номер автотранспортного средства и его владельцу немедленно будет выслано уведомление о том, что он кому-то заблокировал проезд.

УДК 655.3:004.434.5

Студ. Шаров Р. К.

Науч. рук. асс. Сулим П. Е.

(кафедра полиграфического оборудования и систем обработки информации, БГТУ)

## **ОБРАБОТКА ИЗОБРАЖЕНИЙ НА ЯЗЫКЕ PYTHON**

Выбор в пользу того или иного языка программирования является следствием огромного количества факторов - от требований эффективно-го использования ресурсов вычислительной системы до наличия в нужное время подходящей книжки. Поэтому, чтобы избежать непродуктивного и спорного сравнения различных языков программирования друг с другом, рассмотрим только аргументы в пользу выбора языка Python.

1. Python - сравнительно молодой язык. Создавая его в 1990 - 1991 годах, его автор Гвидо ван Россум учёл все достоинства и недостатки предшествующих языков программирования.

2. Python имеет достаточно долгую историю развития и использования (почти 20 лет). В настоящее время Python поддерживается обширным международным сообществом разработчиков.

3. Python - развивающийся язык, используемый в реальных проектах.

4. Средства для работы с Python относятся к категории свободно распространяемого программного обеспечения (СПО). Это гарантирует во-первых, от каких-либо претензий относительно использования интеллектуальной собственности, а во-вторых, от превращения Python в обозримом будущем в мёртвый язык (например, как популярный Turbo Pascal).

Python прост в использовании. Базовая цель любого языка программирования – построить «мостик» между программистом и компьютером. Такие популярные языки как Visual Basic, C#, Java называют языками высокого уровня. Этот термин предполагает, что они ближе к человеческому языку, чем к машинному. Python с его простыми и ясными правилами еще приближается к английскому языку. Создание программ на Python настолько незамысловатый процесс, что о нем иногда говорят как о «программировании со скоростью мысли». С точки зрения профессионального программиста, легкость Python – залог высокой производительности труда. Программы на Python ко-

роткие и требуют меньше времени на разработку, чем программы на многих других популярных языках.

Python – мощный язык. Python обладает всеми возможностями, которых следует ожидать от современного языка программирования: работа с *xml/html* файлами, работа с *http* запросами, GUI (графический интерфейс), создание веб-сценариев, работа с FTP, работа с изображениями, аудио и видео файлами, робототехника, программирование математических и научных вычислений. Им пользуются такие крупнейшие компании как Google, IBM, Industrial Light + Magic, Microsoft, NASA, Red Hat, Verizon, Xerox и Yahoo!. Профессиональные разработчики игр также применяют Python: Electronic Arts, 2K Games и Disney Interactive Media Group – все эти компании публикуют игры с кодом на Python.

Python – объектно-ориентированный язык. Объектно - ориентированное программирование (ООП) – современный подход к решению задач с помощью вычислительных машин. В рамках ООП собственная информация программы и команды, которые она передает компьютеру, записываются интуитивно понятным образом. Это, конечно, не единственный способ разработки программ, но в больших проектах, как правило, предпочтительный. Python совмещает в себе функциональность и гибкость. Python – «склеивающий» язык. Python легко интегрировать с другими языками, например C, C++ или Java. Таким образом, программист на Python может пользоваться разработками, уже имеющимися на других языках, и обращать в свою пользу сильные стороны этих языков (скажем, быстродействие C/C++).

Python работает на всех платформах. Python является кроссплатформенным языком программирования. Он работает во всех основных операционных системах. Любая программа на языке Python, должна выполняться на любом современном компьютере с установленной поддержкой Python. Способы настройки Python для разных операционных систем слегка отличаются. Python-программу можно запустить на любой машине: от миниатюрного Palm до суперкомпьютера Cray.

Python – бесплатная система с открытым исходным кодом. Интерпретатор Python бесплатен. Лицензия, под которой доступен Python, разрешает и многие другие интересные вещи. Следование концепции открытого исходного кода – одна из многих особенностей, делающих Python столь популярным и успешным.

Для работы с изображениями в Python наиболее часто используют библиотеку PIL (Python Image Library). Однако разработка библиотеки прекращена. Но проект под названием Pillow, являющийся [форком](#) PIL,

развивается и включает, в том числе, поддержку Python 3.x. Этот форк был принят в качестве замены оригинальной библиотеки [1].

Возможности библиотеки: поддержка бинарных, полутоновых, индексированных, полноцветных и CMYK изображений; поддержка форматов BMP, EPS, GIF, JPEG, PDF, PNG, PNM, TIFF и некоторых других на чтение и запись; поддержка множества форматов (ICO, MPEG, PCX, PSD, WMF и др.) только для чтения; преобразование изображений из одного формата в другой; правка изображений (использование различных фильтров, масштабирование, рисование, матричные операции и т. д.); использование библиотеки из Tkinter и PyQt.

Например надпись на цветном фоне можно создать с помощью метода `text` объекта-изображения. В ниже приведенном примере надпись наносится на вновь созданное изображение размером 100 на 50 пикселей:

```
from PIL import Image, ImageDraw
text = "Hello, PIL!!!"
color = (0, 0, 120)
img = Image.new('RGB', (100, 50), color)
imgDrawer = ImageDraw.Draw(img)
imgDrawer.text((10, 20), text)
img.save("pil-basic-example.png")
```

Результат программного кода представлен на рисунке 1.

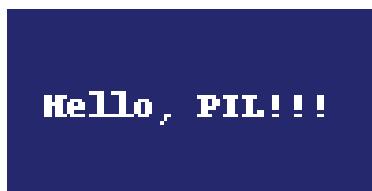


Рисунок 1 - Надпись «Hello, PIL» на цветном фоне

В материале было описано, что Python имеет простой и ясный синтаксис, его библиотеки содержат лаконичную документацию, а процесс тестирования и кодирования - достаточно комфортный. Перенос кода с одной платформы на другую - не всегда безболезненный, удобен для обработки изображения [2, 3].

#### ЛИТЕРАТУРА

1. Интернет ссылка [https://ru.wikipedia.org/wiki/Python\\_Imaging\\_Library](https://ru.wikipedia.org/wiki/Python_Imaging_Library) - cite\_note-pillow-2. – Дата доступа : 15.04.17

2. Сулим, П. Е. Модельное управление ризографической печатью [Электронный ресурс] / П. Е. Сулим. – Мир печати. – Минск, 2011. – Режим доступа: <http://printmedia.mgup.ru/magazine/2011/5/uvpravlennie-rizograficheskoy-pechetju.html>. – Дата доступа: 08.02.2016.

3. Сулим, П. Е. Improvement of the printing quality on a risograph on the basis of the adaptive screening method Proceeding of the 6th Intern. Scientific Conf. / П. Е. Сулим, В. С. Юденков. – Chemnitz: Printing Future Days, 2015. – P. 109 – 116.

УДК 004.942

Студ. Бугаев Е. А.

Науч. рук. асс. Грудо С. К.

(кафедра полиграфического оборудования и систем обработки информации, БГТУ)

### УЛУЧШЕНИЕ КАЧЕСТВА ИЗОБРАЖЕНИЙ В СРЕДЕ MATLAB

Техническая грамотность специалистов по допечатной подготовке сводится не только к умению работы с аппаратными средствами, но и включает грамотное использование различных программных алгоритмов. Построение алгоритмов несет в себе цель улучшить качество представляемой к печати изобразительной информации путем повышения контраста и детализации, уменьшив тем самым визуальные ошибки оператора и автоматизировав процесс допечатной подготовки. Для написания подобного алгоритма в данной работе использовался язык программы *Matlab* с использованием специального пакета функционального расширения *imageprocessingtoolbox*. Программа *Matlab* позволяет работать с изображениями любого размера и использовать любое цветовое пространство, а при помощи пакета *imageprocessingtoolbox* появляется возможность преобразовывать геометрические характеристики изображений, изменять их контраст и детализацию, работать с цветовыми каналами или устранять зашумление [1].

Разработка алгоритма состоит из нескольких этапов. В начале происходит загрузка оригинала изображения в среду *Matlab* и перевод его в полутоновый режим. Загрузка производится при помощи функции *imread*, а его перевод – при помощи функции *rgb2gray*.

Для выравнивания гистограммы полученного полутонового изображения необходимо использовать функцию *histeq*. Данная функция предназначена для преобразования полутоновых и палитровых изображений (рисунок 1).