# Библиотеки Python. Часть 3. Работа с графическими файлами и звуком

# Прозрачность

Kлассная работа

макс. 2 балл.

|  |  |
| --- | --- |
| Ограничение времени | 60 секунд |
| Ограничение памяти | 1Gb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Два изображения можно совместить так, что одно из них будет как бы «полупрозрачным». Для этого значения цветовых компонент каждого совмещаемого пикселя обоих изображений нужно попарно суммировать с определёнными весовыми коэффициентами.  
  
Например, если итоговый цвет (в нотации RGB) вычислять по формуле:

   R = 0.8 \* R1 + 0.2 \* R2    
   G = 0.8 \* G1 + 0.2 \* G2    
   B = 0.8 \* B1 + 0.2 \* B2

то получится, что первое изображение будет иметь 20 процентов прозрачности.  
  
Продемонстрируем эффект прозрачности:  
  
  
  
  
  
  
  
Напишите функцию **transparency(ﬁlename1, ﬁlename2)**, которая, используя PIL, смешивает изображения в пропорции 50/50.

## Формат ввода

Тестирующая система вызовет вашу функцию, передав в неё имена файлов с изображениями:

   transparency("image1.jpg", "image2.jpg")

## Формат вывода

Результат смешивания изображений сохраните в файл с именем **res.jpg**в той же папке, где находится ваша программа.

## Примечания

Результат вычисления каждой компоненты нового пикселя необходимо приводить к **целому**числу с помощью функции **int**.  
Например, так:

   r = int(0.2 \* r1 + 0.8 \* r2)

# Чип и Дейл

Kлассная работа

макс. 3 балл.

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или sound.wav |
| Вывод | стандартный вывод или output.txt |

Мы очень любим мультфильм «Чип и Дейл». А вы?  
Давайте попробуем написать функцию **chip\_and\_dale(number)**, которая сможет монозвуковой файл с любой речью переделать так, чтобы казалось, что слова произносит герой любимого нами мультфильма **Чип**.  
Для этого надо ускорить воспроизведение.

В функцию передаётся натуральное число **i** из диапазона [2..5].  
Необходимо прочитать файл **in.wav**, оставить каждый **i-ый** фрейм и сохранить результат в файле **out.wav**

В качестве образца по работе со звуковым файлом возьмите пример из урока.

## Формат ввода

**chip\_and\_dale(3)**

## Формат вывода

Звуковой файл **out.wav**

**Поворот с размытием**

Kлассная работа

макс. 3 балл.

|  |  |
| --- | --- |
| Ограничение времени | 100 секунд |
| Ограничение памяти | 64Mb |
| Ввод | image.jpg |
| Вывод | стандартный вывод |

В файле **image.jpg** лежит изображение. Напишите функцию **motion\_blur(n)** которая:

1. открывает изображение **image.jpg**
2. поворачивает его на 270 градусов против часовой стрелки без использования циклов (только встроенными средствами PIL)
3. обрабатывает полученное изображение с помощью размытия Гаусса (**GaussianBlur**) с параметром **n**
4. сохраняет результат в файле **res.jpg**

**Формат ввода**

Изображение **image.jpg** в текущей папке



вызов функции:  
**motion\_blur(10)**

**Формат вывода**

Изображение **res.jpg** в текущей папке



# Убрать тишину

Kлассная работа

макс. 3 балл.

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или in.wav |
| Вывод | стандартный вывод или output.txt |

Напишите функцию **break\_the\_silence()**, которая будет убирать тишину из файла **in.wav** и сохранять полученный результат в файле **out.wav**

Обратите внимание, что даже если мы не слышим звука в аудио-файле и не видим колебания звукового спектра на спектрограмме, это не значит, что в файле находится абсолютная тишина.  
В реальности во фреймах участка музыкального файла будут небольшие колебания в диапазоне **[-5; 5]**. Именно такие фреймы ваша функция и должна считать содержащими только тишину.

## Формат ввода

Звуковой файл **in.wav** в формате «моно», лежащий в папке с программой.

## Формат вывода

Звуковой файл **out.wav**, лежащий в папке с программой.

**Миниатюра для сайта**

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или image.jpg |
| Вывод | стандартный вывод или output.txt |

В файле **image.jpg**лежит изображение. Напишите функцию **make\_preview(size,** **n\_colors)**которая:

1. открывает изображение **image.jpg**
2. уменьшает его до размера **size**
3. уменьшает число цветов в изображении до **n\_colors**(такая процедура называется квантование или **quantize**)
4. сохраняет результат в файле **res.bmp**

**Формат ввода**

Изображение **image.jpg**в текущей папке.



вызов функции:

make\_preview((400, 200), 64)

**Формат вывода**



**Примечания**

Обратите внимание, что результат сохраняется в формате **bmp**.

# Кручу-верчу

Kлассная работа

макс. 4 балл.

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или in.wav |
| Вывод | стандартный вывод или output.txt |

Напишите функцию **pitch\_and\_toss()**, которая будет делить файл **in.wav** на четыре части, менять местами первую часть с третьей, а вторую с четвертой и сохранять полученный результат в файле **out.wav**.

Обратите внимание, что в общем случае файл может содержать количество фреймов, не кратное 4. Поэтому первые три части можно сделать одинаковыми с помощью целочисленного деления, а в четвертую записать все оставшиеся данные.

## Формат ввода

Звуковой файл **in.wav** в формате «моно», лежащий в папке с программой.

## Формат вывода

Звуковой файл **out.wav**, лежащий в папке с программой.