

Лабораторная работа №10

«Выполнение проекта»

Вариант 34

По дисциплине:
Основы программирования

Выполнили студентки

1 курса гр. 201-321

Натальина Дарья Александровна

Сусллова Диана Алексеевна

Проверила

_____/Никишина Ирина Николаевна

Москва 2020

Цель работы

Получить практические навыки самостоятельной реализации программного продукта. Закрепление теоретических знаний по дисциплине «Основы программирования», а также практических навыков по программированию на языке Python.

Постановка задачи

Сделать программу, которая конвертирует изображение в текстовый эквивалент. Она должна обеспечивать возможность выполнения перечисленных ниже функций:

- Возможность менять размер шрифта;
- Поддержка конвертации разных форматов изображений;
- Программа поддерживает только русский язык.

Теоретическое введение

Модуль Tkinter

В Python реализовано достаточно много модулей и библиотек, позволяющих создавать графический интерфейс пользователя (GUI). Вот небольшой перечень: Tkinter, PyQt5, PyGTK, wxPython, Pythonwin, и др.

Стандартным модулем, на котором, кстати, разработана IDLE, является модуль Tkinter. Этот модуль устанавливается вместе с Python. Сравнивая разработку, например, на PyQt и Tkinter можно сделать вывод, что Tkinter не является универсальной библиотекой. Его следует использовать только в программах, где требуются лишь основные компоненты, где накладывается мало ограничений на ввод пользователя. Другими словами, Tkinter подходит для простых программ с графическим интерфейсом и т.к. имеет более понятный и ясный синтаксис подходит для обучения.

Далее будем рассматривать решение нашей задачи, поставленной в начале этой лабораторной работы, в рамках возможностей модуля Tkinter.

Первый шаг — это структура программы: организация программы, управляемой событиями.

Виджеты

Виджет (англ. widget) - **элемент интерфейса** - примитив графического интерфейса пользователя (GUI), имеющий стандартный внешний вид и выполняющий стандартные действия. Иногда можно встретить и другое название контрол (англ. control). Вот неполный перечень виджетов Tkinter:

- меню (Menu) - используется для организации меню, в том числе и каскадного;
- кнопка (Button) - используется для запуска команды, или, например, для выбора следующего действия. Так, кнопка "Сохранить" запускает процесс сохранения текущих данных, а кнопка "Удалить" удаляет выбранный элемент;
- радиокнопка (Radiobutton) - используется в группе. Позволяет выбрать только одно значение из нескольких;

- флажок (Checkbox) - позволяет выбрать несколько не взаимоисключающих значений;
- список (Listbox) - этот элемент представляет собой перечень элементов. Настройка конфигурации позволяет пользователю выбрать либо только один элемент, либо несколько.
- многострочное текстовое поле (Text) - этот виджет позволяет вывести или получить от пользователя многострочное текстовое сообщение;
- однострочное текстовое поле (Entry) - этот виджет предназначен для вывода или ввода только одной, как правило, небольшой строки, например, название предмета или фамилию;
- метка (Label) - используется, как правило, для информирования пользователя. Например, в качестве метки выступает надпись, информирующая пользователя о назначении однострочного текстового поля;
- раскрывающийся список (Combobox) - этот виджет используется для выбора одного из элементов, хранящихся в его списке;
- счетчик (Spinbox) - такой виджет используется, например, в качестве счетчика записей, выводимых на экран;
- полоса прокрутки (Scrollbar) - используется для просмотра части текста или списка, которые выходят за рамки, например, текстового поля;
- ползунок (Slider) – позволяет выбирать значения из набора, задаваемого программистом. Может быть использован, например, как регулятор громкости;
- окна сообщений (message box) — это диалоговые окна, которые позволяют выводить сообщения, предупреждения или ошибки при взаимодействии с пользователем. Кроме этого, имеются диалоговые окна для работы с файлами: окна выбора файла при открытии или сохранении файла.

Все виджеты имеют два набора конфигурационных параметров. Один набор параметров является общим для всех виджетов: размеры, положение на форме.

Второй набор параметров определяется индивидуальными характеристиками виджета: состояние счетчика виджета Spinbox, или индекс значения, выбранного пользователем в виджете Combobox.

События, связанные с виджетом (клик мыши, нажатие клавиш клавиатуры) могут быть привязаны к методу, который будет обрабатывать событие. Например, при вводе текста в однострочном поле (Entry) и нажатии клавиши Enter (используется для завершения ввода) может быть вызван метод, который прочитает введенный текст.

Важно отметить, что все виджеты будут показаны на форме только после их обработки упаковщиком.

Существуют три метода упаковки виджетов:

Метод pack. В этом методе указывается, как виджет будет заполнять пространство формы, будет ли он растягиваться при изменении размеров формы, как виджет будет размещаться на форме.

Метод place. Этот метод позволяет программисту контролировать положение виджета через задание координат, и его размеры через задание ширины и высоты виджета. Координаты положения задаются относительно верхнего левого угла окна, к которому привязан виджет. При этом, если окно является вложенным и его координаты меняются, то относительные координаты виджета не меняются.

Метод grid. Наиболее часто используемый упаковщик. В этом методе окно делится условной сеткой на ячейки (grid). Программист задает ячейку, в которой будет размещаться виджет, через номер столбца и колонки. При этом можно указать, сколько строк и столбцов будет занимать виджет.

Последний шаг - это представление о том, как можно писать программу с GUI используя такие понятия, как объектно-ориентированное программирование, класс, объект.

Описание программы

Программа выполнена на алгоритмическом языке Python 3.9 (64-bit), Windows 10 и состоит из следующих частей: ввод данных (изображение), конвертация и вывод данных (текстовый эквивалент изображения) на экране монитора.

Описание входных и выходных данных

Входные данные – изображение, выходные данные имеют тип string.

Листинг программы

```
from tkinter import *
import Pmw as Pmw
import numpy as np
from PIL import Image

# 70 levels of gray
gscale1 = "$@B%8&WM#*oahkbdpqwmZO0QLCJUYXzcvunxrjft/\|()1{}[]?-_+~<>i!lI;:,\\"^`'. "

# 10 levels of gray
gscale2 = '@%#*+=-:. '

def getAverageL(image):
    # get image as numpy array
    im = np.array(image)

    # get shape
    w, h = im.shape

    # get average
    return np.average(im.reshape(w * h))

def convertImageToAscii(fileName, cols, scale):
    # declare globals
    global gscale1, gscale2
```

```

# open image and convert to grayscale
image = Image.open(fileName).convert('L')

# store dimensions
W, H = image.size[0], image.size[1]
print("Size of your initial image is %d x %d" % (W, H))

# compute width of tile
w = W / cols

# compute tile height based on aspect ratio and scale
h = w / scale

# compute number of rows
rows = int(H / h)

# print("tile dims: %d x %d" % (w, h))
print("Size of future ASCII-art will take %d cols and %d rows" % (cols,
rows))

# check if image size is too small
if cols > W or rows > H:
    print("This image too small for this convertor!")
    exit(0)

# ascii image is a list of character strings
aimg = []
# generate list of dimensions
for j in range(rows):
    y1 = int(j * h)
    y2 = int((j + 1) * h)

    # correct last tile
    if j == rows - 1:
        y2 = H

    # append an empty string
    aimg.append("")

    for i in range(cols):

        # crop image to tile
        x1 = int(i * w)
        x2 = int((i + 1) * w)

        # correct last tile
        if i == cols - 1:
            x2 = W

        # crop image to extract tile
        img = image.crop((x1, y1, x2, y2))

        # get average luminance
        avg = int(getAverageL(img))

        # look up ascii char

```

```

        gsval = gscale2[int((avg * 9) / 255)]
        aimg[j] += gsval

# return txt image
return aimg

# main() function
def main():
    imgFile = file.get()

    scale = 0.43

    outFile = 'OPEN_ME.txt'

    cols = num_of_cols.get()

    print('Start converting the image...')
    # convert image to ascii txt
    aimg = convertImageToAscii(imgFile, cols, scale)

    # open file
    f = open(outFile, 'w')

    # write to file
    for row in aimg:
        # cleanup
        f.write(row + '\n')

    f.close()
    print("The finished ASCII-art is located in %s" % outFile)

    # create the GUI for watching the converted image
    filename = outFile
    root = Tk()
    root.title('ConvertToText')
    root.geometry('1600x900')
    top = Frame(root)
    top.pack(side='top')
    text = Pmw.ScrolledText(top,
                            borderframe=0,
                            vscrollmode='dynamic',
                            hscrollmode='dynamic',
                            text_width=180,
                            text_height=45,
                            labelpos='n',
                            text_wrap='none',
                            )
    text.pack()
    text.insert('end', open(filename, 'r').read())
    Button(root, text="OK", command=root.destroy).pack()
    root.mainloop()

# create the GUI with settings
root = Tk()
root.title("ConvertToText")

```

```

file = StringVar()
num_of_cols = IntVar()

file_label = Label(text="Введите название файла с картинкой для
конвертации:")
num_of_cols_label = Label(text="Введите желаемый размер ASCII-арта:")

file_label.grid(row=0, column=0, sticky="w")
num_of_cols_label.grid(row=1, column=0, sticky="w")

file_entry = Entry(textvariable=file)
num_of_cols_entry = Entry(textvariable=num_of_cols)

file_entry.grid(row=0, column=1, padx=5, pady=5)
num_of_cols_entry.grid(row=1, column=1, padx=5, pady=5)

exit = Button(text="Выход", command=root.destroy)
exit.grid(row=2, column=0, padx=3, pady=5, sticky="e")
message_button = Button(text="Конвертировать", command=main)
message_button.grid(row=2, column=1, padx=4, pady=5, sticky="e")

root.mainloop()

```

Результат работы программы

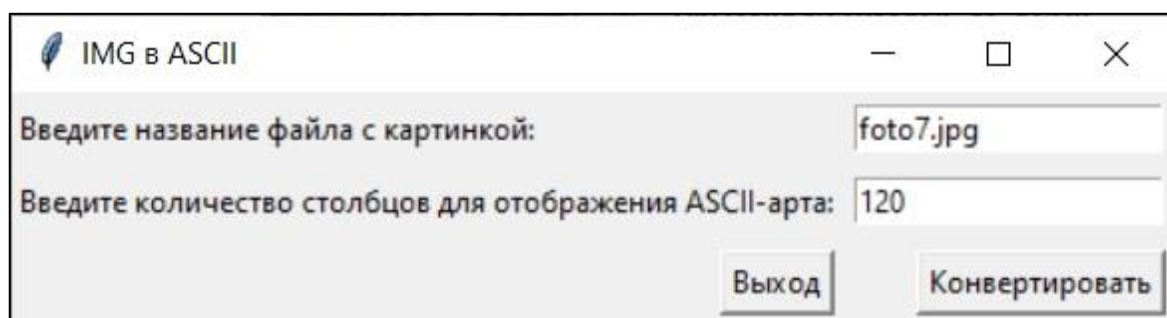


Рис. 1 – Результат работы программы

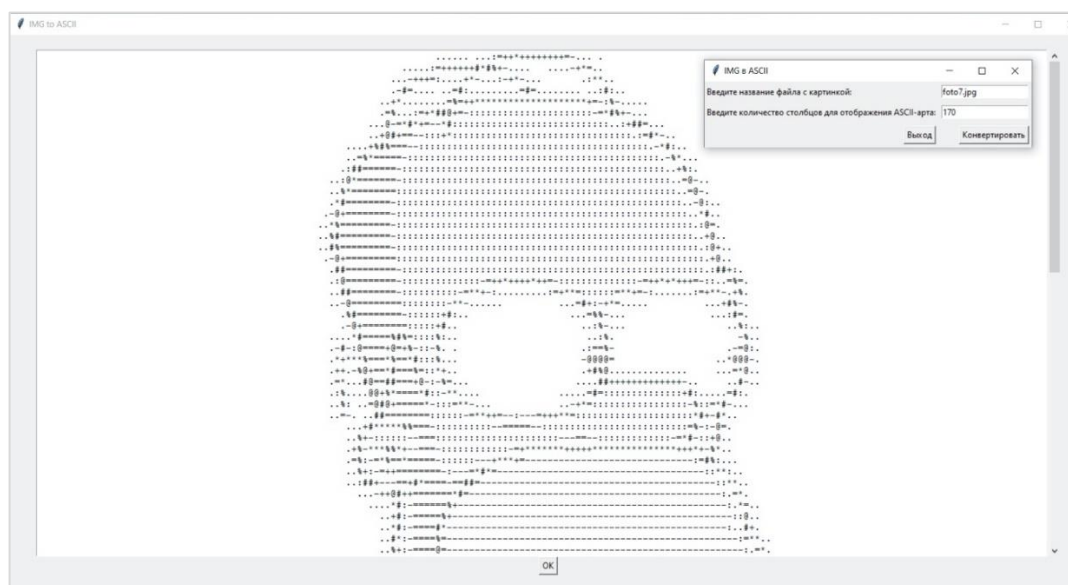


Рис. 2 – Результат работы программы в GUI



Рис. 3 – Результат работы программы в Блокноте

Список использованных интернет-ресурсов

1. <https://pythonworld.ru/>;
2. <https://devpractice.ru/>.