УДК 004

Музыкальный проигрыватель на языке программирования Python

Чингалаев Сергей Алексеевич
Приамурский государственный университет им. Шолом-Алейхема
Студент

Аннотация
Целью данной статьи является разработка программы для проигрывания музыки. Для достижения этой цели был использован язык программирования Python. В этой статье была разработана программа, с помощью которой можно воспроизводить файлы с mp3 расширением.
Ключевые слова: Python, программа, воспроизведение, музыка, разработка.

Music player in Python programming language

Chingalaev Sergey Alekseevich
Sholom-Aleichem Priamursky State University
Student

Abstract
The purpose of this article is to develop a program for playing music. To achieve this goal, the Python programming language was used. In this article, a program was developed with which you can play files with mp3 extension.
Keywords: Python, program, playback, music, development.

Большинство людей в современном мире слушают музыку. Данная программа является музыкальным проигрывателем и разработана для воспроизведения mp3 файлов. Программа, созданная при помощи языка программирования Python, позволит как слушать музыку, так и создавать в ней собственные плейлисты.

Цель исследования: разработка программы для воспроизведения mp3 файлов и создания плейлистов для них при помощи языка программирования Python.

С.Р. Паршикова В своей статье провела сравнительный анализ наиболее популярных языков программирования для, рассматриваются достоинств и недостатков, приводится статистика использования и распространения этих языков [1]. 2. Б. Айзенберг, Д.К. Тивадар, Л.Т. Дэвис в своей статье описали процесс тестирования и оптимизации веб-сайтов и написали руководство по google website optimizer [2]. Т.М. Никулин в своей статье описал способы разработки программы «mp3 плеер» [3]. Также не мало важны иностранные источники [4].

Сначала необходимо подключить все нужные библиотеки (Рис.1).

```
import os
import easygui_qt as gui
from os import listdir
from os.path import isfile, join
import threading
from pygame import mixer
import time
from mutagen.mp3 import MP3
```

Рис.1. Импортирование библиотек

Далее даем пользователю выбрать папку с музыкой и отфильтровываем файлы в папке с расширением mp3 (Рис.2).

```
gui.show_message(message='Выберите папку с музыкой', title='Музыкальный проигрыватель'

directory = gui.get_directory_name(title='Выберите папку с музыкой')

onlyfiles = [f for f in listdir(directory) if isfile(join(directory, f))]

mp3_list = filter(lambda var: var.split('.')[-1] == 'mp3', onlyfiles)
```

Рис.2. Создание списка музыки

Следующим шагом пишем функцию, которая будет воспроизводить музыку (Рис.3)

```
music_i = 0
def play(*mp3_files):
    global music_i
    mixer.init()
    if music_i >= len(mp3_files):
        music_i = 0
    if music_i < 0:
        music_i = len(mp3_files) - 1
    while True:
        if mixer.get_busy():
            continue
        mp3 = os.path.join(directory.replace('/', '\\'), mp3_files[music_i])
        song = MP3(mp3)
        songLength = song.info.length
        mixer.music.load(mp3)
        mixer.music.play()
        time.sleep(songLength)
        music_i+=1
        if music_i >= len(mp3_files):
            music_i = 0
```

Рис.3. Функция воспроизведения музыки

Так как музыка должна воспроизводиться параллельно с работой программы, нужно запускать функцию в новом потоке. (Рис.4)

```
play_thread = None
def play_musics():
    global play_thread
    play_thread = threading.Thread(target=play, args=(mp3_list), daemon=True)
    play_thread.start()
```

Рис.4. Запуск потока

Далее в цикле вызываем окно, которое будет ожидать действий пользователя. При нажатии на кнопку Yes будет проигрываться следующая песня, при нажатии на No – предыдущая, а кнопка Cancel закрывает программу. (Рис 5.).

```
while True:
    choice = gui.get_yes_or_no(message='След песня - Yes, Пред песня - No, Закрыть - Cancel')
    if choice is None:
        break
    if choice:
        music_i += 1
    else:
        music_i -= 1
    play_musics()
```

Рис.5. Код меню управления воспроизведением

Реализовано сохранение и чтение сохраненных плейлистов (Рис 6.).

```
try:
    playlists = list(open('playlists.txt').read())
    playlists_names = list(open('playlist_names.txt').read())
except:
    playlists = []
    playlists_names = []

if plbl:
    playlist = gui.get_list_of_choices(title='Создать плейлист', choices=mp3_list)
    playlist_name = gui.get_string(message='Введите название плейлиста')
    playlists.append(playlist)
    playlists_names.append(playlist_name)
    open('playlists.txt', 'w').write(playlists)
    open('playlist_names.txt', 'w').write(playlists)

pl_i = gui.get_choice(message='Выберите плейлист', choices=playlists_names)
mp3_list = playlists[pl_i]
```

Рис.6. Чтение и сохранение плейлистов

Далее показан наглядный пример использования программы и ее интерфейса. Сначала программа предлагает пользователю вбрать папку, в которой хранятся mp3 файлы (Рис 7).

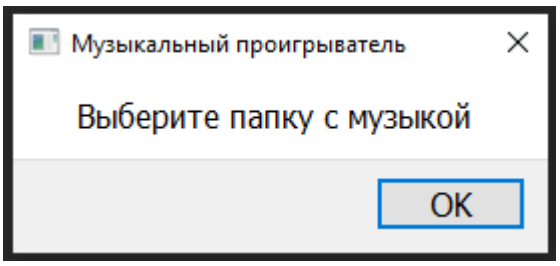

Рис.7. Стартовое окно

После этого пользователю необходимо указать путь к файлу или папке на котором хранятся аудиофайлы (Рис 8).

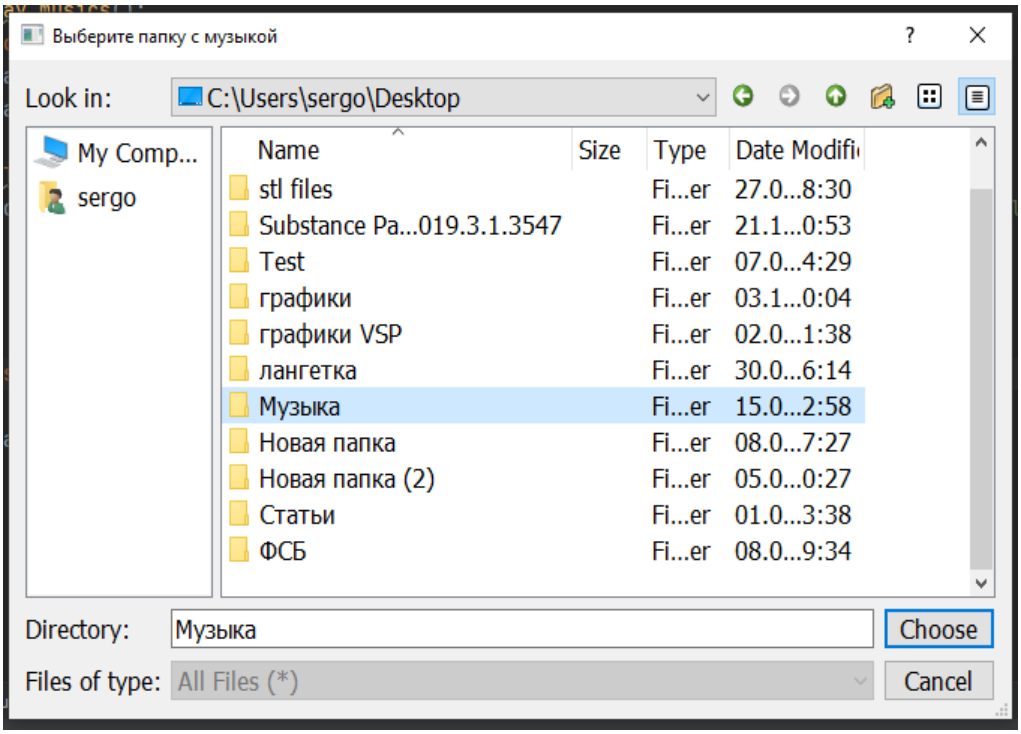

Рис.8. Указание пути к файлам

Окно, отвечающее за переключение плейлиста на «следующий» или «предыдущий» аудиофайл. Также в окне показано какой сейчас воспроизводится аудиофайл (Рис 9).

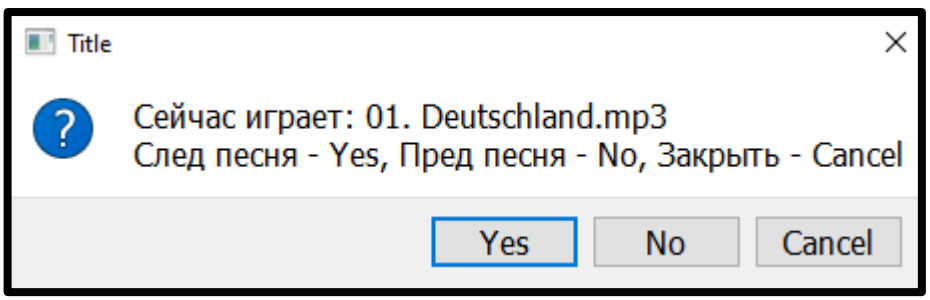

Рис.9. Управление воспроизведением

Для создания своего плейлиста в появившемся следует нажать «Yes» (Рис.10)

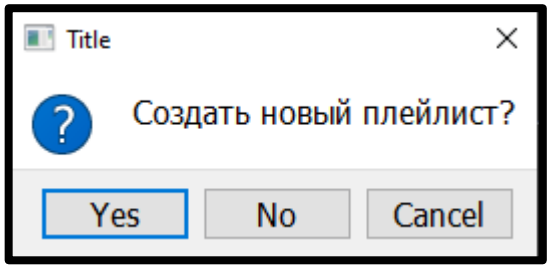

Рис.10. Создание плейлиста

После этого в следующем меню из имеющихся файлов, добавленных раньше можно составить свой плейлист. После нажатия «ok» он будет сохранен (Рис.11)

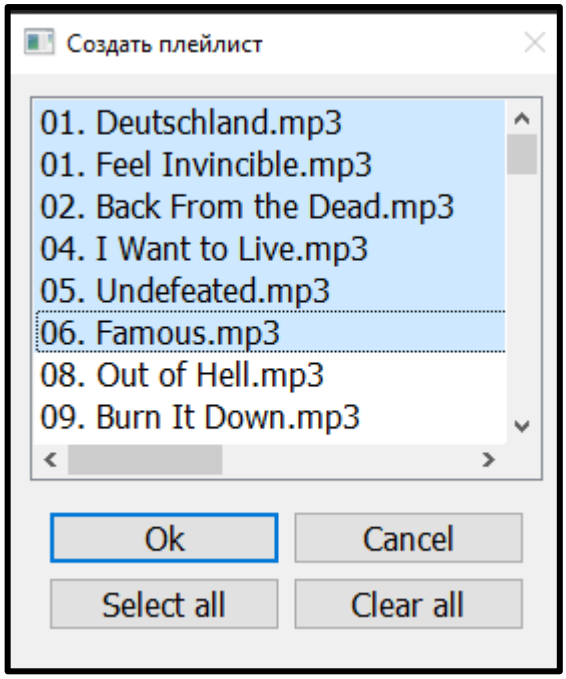

Рис.11. Редактор плейлиста

Далее в появившемся окне вводим название плейлиста, который мы планируем сохранить (Рис.12)

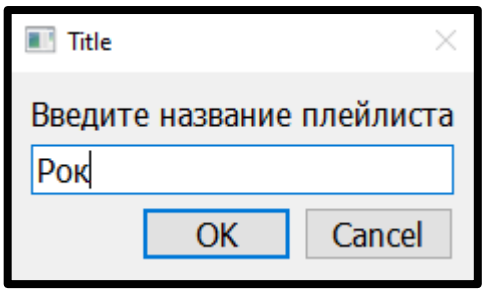

Рис.12. Сохранение плейлиста

Теперь мы можем выбирать сохраненные раннее плейлисты для прослушивания (Рис.13)

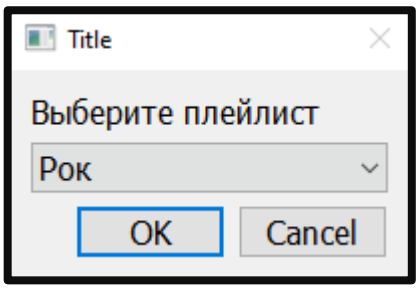

Рис.13. Выбор плейлиста

Таким образом, была написана программа, для воспроизведения mp3 файлов и создания плейлистов для них при помощи языка программирования Python.

Библиографический список

1. Паршикова С.Р. Анализ языков программирования для веб-разработки: php, python и ruby. // Образование и наука в современном мире. Инновации. 2016. № 6-1. С. 195-202.
2. Келлер П.Д., Келли М.Д. Музыкальный проигрыватель и способ хранения звуковых дорожек в музыкальном проигрывателе // Патент на изобретение RU 2370832 C2, 20.10.2009. Заявка № 2003114729/28 от 20.05.2003.
3. Никулин Т.М. Разработка программы "mp3 плеер" // Научно-техническая конференция студентов, аспирантов и молодых специалистов НИУ ВШЭ им. Е.В. Арменского. Материалы конференции. Московский институт электроники и математики Национального исследовательского университета «Высшая школа экономики». 2015. С. 116-117.
4. Erdmann M., Fischer R., Hinzmann A., Klimkovich T., Müller G., Steggemann J., Hegner B. Visualization of the cms python configuration system // Journal of Physics: Conference Series. 2010. Т. 219. № 1 Part 4. С. 042008.