

## **Разработка программы для слияния нескольких PDF документов в один на языке программирования Python**

*Кизянов Антон Олегович*

*Приамурский государственный университет имени Шолом-Алейхема студент*

### **Аннотация**

В данной статье продемонстрирован процесс создания программы для склейки 2 и более PDF файлов в один.

**Ключевые слова:** Python, os, PyPDF2.

## **Develop a program to merge multiple PDF documents into one in the Python programming language**

*Kizyanov Anton Olegovic*

*Sholom-Aleichem Priamursky State University student*

### **Abstract**

This article demonstrated the process of creating a program for bonding two or more PDF files into one.

**Keywords:** Python, os, PyPDF2.

Предположим, предстоит выполнить утомительную работу по слиянию нескольких PDF документов в один PDF файл. Каждый из документов начинается с титульного листа на первой странице, но вы не хотите, чтобы первые страницы попадали в окончательный документ. Несмотря на то что существует много бесплатных программ позволяющих объединять PDF документы, все, на что способны многие из них – это просто слияние исходных файлов в единый файл. Давайте напишем программу, позволяющую выбирать страницы, которые должны включаться в результирующий PDF документ.

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].

Прежде всего программа должна получать список всех файлов с расширением .pdf в текущем рабочем каталоге и выполнять их сортировку.

```
import PyPDF2, os

pdfFiles = []
for filename in os.listdir('.'):
    if filename.endswith('.pdf'):
        pdfFiles.append(filename)
pdfFiles.sort()

pdfWriter = PyPDF2.PdfFileWriter()
```

Рис. 1

На рисунке 1 изображен кусок кода, на котором мы импортируем модули `os` и `PyPDF2`. Модуль `os` нужен для работы с директориями, а `PyPDF2` для работы с PDF документами.

Вызов `os.listdir('.')` возвращает список всех файлов, находящихся в текущем рабочем каталоге. Затем код просматривает этот список в цикле и добавляет в новый список `pdfFiles` лишь файлы с расширением `.pdf`. После этого список `pdfFiles` сортируется в алфавитном порядке, который задается именованным аргументом `key=str.lower` при вызове метода `sort()`.

Теперь программа должна прочитать каждый из PDF файлов, входящих в список `pdfFiles`.

```
for filename in pdfFiles:
    pdfFileObj = open(filename, 'rb')
    pdfReader = PyPDF2.PdfFileReader(pdfFileObj)
```

Рис. 2

На рисунке 2 второй кусок кода, в котором в цикле каждый PDF файл открывается в режиме чтения двоичных данных путем вызова метода `open()` с передачей ему строки `'rb'` в качестве второго аргумента. Метод `open()` возвращает объект `File`, который передается функции `PyPDF2.PdfFileReader()`, создающий объект `PdfFileReader` для данного PDF файла.

Для каждого из PDF файлов необходимо отобразить в цикле все страницы, за исключением первой.

```
for pageNum in range(1, pdfReader.numPages):
    pageObj = pdfReader.getPage(pageNum)
    pdfWriter.addPage(pageObj)
```

Рис. 3

На рисунке 3 код в цикле копирует каждый объект `Page` по отдельности в объект `PdfFileWriter`. Поскольку в принятой в модуле `PyPDF2` системе

отчета первой странице соответствует индекс 0, цикл должен выполняться в диапазоне индексов, определяемом значениями 1 и pdfReader.numPages.

После выполнения вложенных циклов переменная pdfWriter будет содержать объект PdfFileWriter, включающий страницы всех объединений PDF-документов. Последний шаг состоит в том, чтобы записать это содержимое в файл на жесткий диск.

```
pdfOutput = open('allminutes.pdf', 'wb')
pdfWriter.write(pdfOutput)
pdfOutput.close()
```

Рис. 4

На рисунке 4 открывается выходной файл allminutes.pdf в режиме записи двоичных данных, функция open() передается строка 'wb'. В результате последующей передаче результирующего объекта File методу write() создается фактический PDF-файл. Программа завершает вызов метода close().

```
import PyPDF2, os

pdfFiles = []
for filename in os.listdir('.'):
    if filename.endswith('.pdf'):
        pdfFiles.append(filename)
pdfFiles.sort()

pdfWriter = PyPDF2.PdfFileWriter()

for filename in pdfFiles:
    pdfFileObj = open(filename, 'rb')
    pdfReader = PyPDF2.PdfFileReader(pdfFileObj)

    for pageNum in range(1, pdfReader.numPages):
        pageObj = pdfReader.getPage(pageNum)
        pdfWriter.addPage(pageObj)

pdfOutput = open('allminutes.pdf', 'wb')
pdfWriter.write(pdfOutput)
pdfOutput.close()
```

Рис. 5

На рисунке 5 представлен весь код программы.

Вывод: Разработана система склейки нескольких pdf-документов в один с выборкой по листам.

**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)
5. PyPDF2. [Электронный ресурс]. URL: <https://pythonhosted.org/PyPDF2/> (дата обращения: 25.01.2017)