

Задания для выполнения в классе:

1. Написать программу, которая считывает .wav-файл, делит его на две половины и каждую записывает в отдельный файл. Оформите в виде функции, которая берёт на вход имя файла и возвращает имена получившихся файлов.

```
import wave
import struct

def split_wave(filename: str) -> list[str]:
    sampwidth_to_char = {1: "c", 2: "h", 4: "i"}

    f = wave.open(filename)
    num_samples = f.getnframes()
    samplerate = f.getframerate()
    sampwidth = f.getsampwidth()
    num_channels = f.getnchannels()
    fmt = str(num_samples * num_channels) + sampwidth_to_char[sampwidth]
    signal = struct.unpack(fmt, f.readframes(num_samples * num_channels))

    midpoint = len(signal) // 2
    first = signal[:midpoint]
    second = signal[midpoint:]

    res_filenames = []
    for i, part in enumerate([first, second]):
        part_filename = f"{filename.split('.')[0]}_part_{i + 1}.wav"
        res_filenames.append(part_filename)

        fmt = str(len(part)) + sampwidth_to_char[sampwidth]
        data_packed = struct.pack(fmt, *part)
        f = wave.open(part_filename, "wb") # открываем на запись (w) в бинарно
        f.setnchannels(num_channels)
        f.setsampwidth(sampwidth)
        f.setframerate(samplerate)
        f.writeframes(data_packed)
        f.close()

    return res_filenames

import scipy.io.wavfile as wavfile

def split_scipy(filename: str) -> list[str]:
    samplerate, signal = wavfile.read(filename)

    midpoint = len(signal) // 2
```

```

first = signal[:midpoint]
second = signal[midpoint:]

res_filenames = []
for i, part in enumerate([first, second]):
    part_filename = f"{filename.split('.')[0]}_part_{i + 1}.wav"
    res_filenames.append(part_filename)
    wavfile.write(part_filename, samplerate, part)

return res_filenames

```

```
!pip install wavio
```

```
import wavio
```

```

def split_wavio(filename: str) -> list[str]:
    data = wavio.read(filename)

    midpoint = len(data.data) // 2
    first = data.data[:midpoint]
    second = data.data[midpoint:]

    res_filenames = []
    for i, part in enumerate([first, second]):
        part_filename = f"{filename.split('.')[0]}_part_{i + 1}.wav"
        res_filenames.append(part_filename)
        wavio.write(part_filename, part, data.rate)

    return res_filenames

```

2. Написать программу, которая считывает .sbl-файл и сохраняет его как .wav. Оформите в виде функции, которая берёт на вход имя файла, частоту дискретизации, количество байт на отсчёт (по умолчанию - 2) и количество каналов (по умолчанию - 1) и возвращает имя получившегося файла.

```

def sbl_to_wav(filename: str, samplerate: int, sampwidth: int = 2, num_channels: int = 1) ->
    with open(filename, "rb") as fi:
        raw_signal = fi.read()

    # обратите внимание, что распаковывать данные не нужно!
    # потому что их бы всё равно пришлось запаковывать обратно

    res_filename = filename.split('.')[0] + ".wav"
    fo = wave.open(res_filename, "wb") # открываем на запись (w) в бинарном режиме (b)
    fo.setnchannels(num_channels)
    fo.setsampwidth(sampwidth)
    fo.setframerate(samplerate)

```

```
fo.writeframes(raw_signal)
fo.close()
return res_filename
```