Программа простого сигнала

import numpy as np

import matplotlib.pyplot as plt

import sounddevice as sd

duration = 1

amplitude = 0.2

fs = 80000

timeSamples = np.arange(np.ceil(duration \* fs)) / fs

signal = amplitude \* np.sin(duration \* np.pi \* 130 \* timeSamples)

signal2 = amplitude \* np.sin(duration \* np.pi \* 146 \* timeSamples)

signal3 = amplitude \* np.sin(duration \* np.pi \* 164 \* timeSamples)

signal4 = amplitude \* np.sin(duration \* np.pi \* 174 \* timeSamples)

signal5 = amplitude \* np.sin(duration \* np.pi \* 196 \* timeSamples)

signal6 = amplitude \* np.sin(duration \* np.pi \* 220 \* timeSamples)

signal7 = amplitude \* np.sin(duration \* np.pi \* 246 \* timeSamples)

signal\_2 = amplitude \* np.sin(duration \* np.pi \* 523 \* timeSamples)

signal2\_2 = amplitude \* np.sin(duration \* np.pi \* 587 \* timeSamples)

signal3\_2 = amplitude \* np.sin(duration \* np.pi \* 659 \* timeSamples)

signal4\_2 = amplitude \* np.sin(duration \* np.pi \* 698 \* timeSamples)

signal5\_2 = amplitude \* np.sin(duration \* np.pi \* 784 \* timeSamples)

signal6\_2 = amplitude \* np.sin(duration \* np.pi \* 880 \* timeSamples)

signal7\_2 = amplitude \* np.sin(duration \* np.pi \* 987 \* timeSamples)

signalSumm = np.concatenate((signal\_2, signal6\_2, signal5, signal, signal4\_2, signal4, signal5\_2, signal5, signal7\_2, signal\_2, signal4, signal3, signal2), axis=0)

sd.play(signalSumm, fs)

Преобразование Фурье

import numpy as np

import matplotlib.pyplot as plt

import sounddevice as sd

duration = 1

amplitude = 0.2

fs = 80000

timeSamples = np.arange(np.ceil(duration \* fs)) / fs

signal = amplitude \* np.sin(duration \* np.pi \* 130 \* timeSamples)

signal2 = amplitude \* np.sin(duration \* np.pi \* 146 \* timeSamples)

signal3 = amplitude \* np.sin(duration \* np.pi \* 164 \* timeSamples)

signal4 = amplitude \* np.sin(duration \* np.pi \* 174 \* timeSamples)

signal5 = amplitude \* np.sin(duration \* np.pi \* 196 \* timeSamples)

signal6 = amplitude \* np.sin(duration \* np.pi \* 220 \* timeSamples)

signal7 = amplitude \* np.sin(duration \* np.pi \* 246 \* timeSamples)

signal\_2 = amplitude \* np.sin(duration \* np.pi \* 523 \* timeSamples)

signal2\_2 = amplitude \* np.sin(duration \* np.pi \* 587 \* timeSamples)

signal3\_2 = amplitude \* np.sin(duration \* np.pi \* 659 \* timeSamples)

signal4\_2 = amplitude \* np.sin(duration \* np.pi \* 698 \* timeSamples)

signal5\_2 = amplitude \* np.sin(duration \* np.pi \* 784 \* timeSamples)

signal6\_2 = amplitude \* np.sin(duration \* np.pi \* 880 \* timeSamples)

signal7\_2 = amplitude \* np.sin(duration \* np.pi \* 987 \* timeSamples)

N = duration \* fs

signalSumm =signal\_2 + signal6\_2 + signal5 + signal + signal4\_2 + signal4 + signal5\_2 + signal5 + signal7\_2 + signal\_2 + signal4 + signal3 + signal2

spector = np.fft.rfft(signalSumm)

x\_axis = np.fft.rfftfreq(N, 1/fs)

plt.plot(x\_axis, np.abs(spector))

plt.show()

new\_sig = np.fft.irfft(spector)

plt.plot(new\_sig[:500])

plt.show()