```
class OSCListener():
    def __init__(self):
        self.serverip = "127.0.0.1"
        self.port = 12000
        #Programaufruf
        self.appStart = time.time()
        self.soundFetcher = fsFetcher()
        self.soundFetcher.createDirs()
    def handlerForWLR(self, x, y, z):
        # Will receive message data unpacked in x,yz
        # Koordignatenaufruf, Download
        sounds = self.soundFetcher.selectSounds(geo=[x,y,z])
        download = self.soundFetcher.downloadSounds(sounds)


    def startup(self):
        # Start the system.
        if config.DEBUG:
            print("starting up Server...")
        osc_startup()

        # Make server channels to receive packets.
        osc_udp_server(self.serverip, self.port, "aservername")

        # Associate Python functions with message address patterns, using default
        # argument scheme OSCARG_DATAUNPACK.
        osc_method("/incommingWLR*", self.handlerForWLR)
        if config.DEBUG:
            print("listening...")

    def shutdown(self):
        osc_terminate()

    def listenLoop(self):
        # Periodically call osc4py3 processing method in your event loop.
        finished = False
        while not finished:
            try:
                osc_process()
            except KeyboardInterrupt:
                finished = True
                osc_terminate()
                raise

        # Properly close the system.
        self.shutdown()


server = OSCListener()
server.startup()
server.listenLoop()

appStop  = time.time()
print ("Programmdauer: ")
print  (appStop - appStart)
```