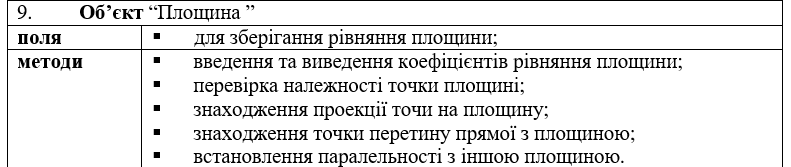
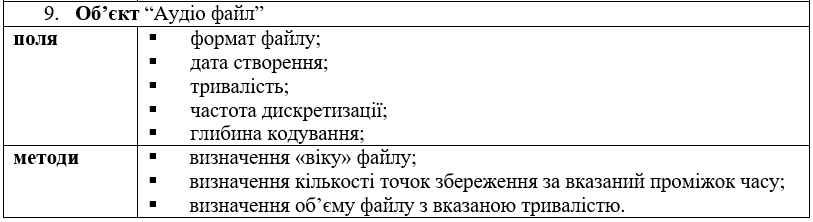
Lab 10

1)



def a1\_and\_a2(a, b):  
 return a / b  
  
  
def a\_coef\_equal\_b\_coef(\*args):  
 for i in range(0, len(args)-3, 2):  
 if a1\_and\_a2(args[i], args[i+1]) != a1\_and\_a2(args[i+2], args[i+3]):  
 return False  
 return True  
  
  
class Plane:  
 def \_\_init\_\_(self, \_a, \_b, \_c, \_d=0):  
 self.a = \_a  
 self.b = \_b  
 self.c = \_c  
 self.d = \_d  
  
 def return\_of\_coefficients(self):  
 return self.a, self.b, self.c, self.d  
  
 def point\_in\_plane(self, \_x, \_y, \_z):  
 return self.a \* \_x + self.b \* \_y + self.c \* \_z + self.d == 0  
  
 def parallel\_planes(self, a1, b1, c1, d1):  
 return a\_coef\_equal\_b\_coef(self.a, a1, self.b, b1, self.c, c1, self.d, d1)  
  
 def intersection\_of\_line\_and\_plane(self, \_x, \_y, \_z, l, m, p):  
 ch = self.a \* \_x + self.b \* \_y + self.c \* \_z + self.d  
 zn = self.a \* l + self.b \* m + self.c \* p  
 t = - (ch / zn)  
 x1 = l \* t + \_x  
 y1 = m \* t + \_y  
 z1 = p \* t + \_z  
 m = [x1, y1, z1]  
 return m  
  
 def point\_projection(self, \_x, \_y, \_z):  
 ch = self.a \* \_x + self.b \* \_y + self.c \* \_z + self.d  
 zn = self.a \*\* 2 + self.b \*\* 2 + self.c \*\* 2  
 t = - (ch / zn)  
 x1 = self.a \* t + \_x  
 y1 = self.b \* t + \_y  
 z1 = self.c \* t + \_z  
 m = [x1, y1, z1]  
 return m

2) 

import datetime  
  
  
class Audio\_file:  
 def \_\_init\_\_(self, \_format, \_date, \_duration, \_frequency, \_deep):  
 self.format = \_format  
 self.date = \_date  
 self.duration = \_duration  
 self.frequency = \_frequency  
 self.deep = \_deep  
  
 def age(self):  
 now = datetime.datetime.today()  
 delta = now - self.date  
 return delta  
  
 def points\_of\_save(self):  
 return self.frequency \* self.duration  
  
 def size\_file(self):  
 return self.deep \* self.duration