|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования  **«Российский технологический университет - МИРЭА»** |
| **РТУ МИРЭА** |

Институт информационных технологий (ИИТ)Кафедра вычислительной техники (ВТ)

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №3**

**по дисциплине**

«Инструментальное программное обеспечение разработки и проектирования информационных систем»

Выполнил студент группы ИКМО-01-20 Баранецкий Д.А.

Фамилия И.О.

Принял Куликов А.А.

Фамилия И.О.

Работы выполнены «\_\_»\_\_\_\_\_\_\_2021 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

«Зачтено» «\_\_»\_\_\_\_\_\_\_2021 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Москва, 2021 г.

Оглавление

[Задание 3](#_Toc66636973)

[Код программы 3](#_Toc66636974)

[Скриншоты 10](#_Toc66636975)

# Задание

**Цель практическое работы:**

Создать приложение по отправки e-mail писем из приложения разработанного на Python.

**Задачи:**

• Создать простое приложение по отправки писем.

• Создать форму в QT Designer

# Код программы

MailInterface.py – файл, формирующий форму отправки

# -\*- coding: utf-8 -\*-  
  
# Form implementation generated from reading ui file 'mailSenderUI.ui'  
#  
# Created by: PyQt5 UI code generator 5.15.0  
#  
# WARNING: Any manual changes made to this file will be lost when pyuic5 is  
# run again. Do not edit this file unless you know what you are doing.  
  
  
from PyQt5 import QtCore, QtGui, QtWidgets  
  
  
class Ui\_ProjectWindow(object):  
 def setupUi(self, ProjectWindow):  
 ProjectWindow.setObjectName("ProjectWindow")  
 ProjectWindow.resize(800, 600)  
 self.centralwidget = QtWidgets.QWidget(ProjectWindow)  
 self.centralwidget.setObjectName("centralwidget")  
  
 self.verticalLayout\_3 = QtWidgets.QVBoxLayout(self.centralwidget)  
 self.verticalLayout\_3.setObjectName("verticalLayout\_3")  
  
 self.verticalLayout\_2 = QtWidgets.QVBoxLayout()  
 self.verticalLayout\_2.setObjectName("verticalLayout\_2")  
  
 # первый горизонтальный блок  
 self.horizontalLayout\_1 = QtWidgets.QHBoxLayout()  
 self.horizontalLayout\_1.setObjectName("horizontalLayout\_1")  
 self.label\_1 = QtWidgets.QLabel(self.centralwidget)  
 self.label\_1.setObjectName("label\_1")  
 self.horizontalLayout\_1.addWidget(self.label\_1)  
 # добавление выпадающего списка серверов  
 self.ServerComboBox = QtWidgets.QComboBox(self.centralwidget)  
 self.ServerComboBox.setObjectName("ServerComboBox")  
 self.horizontalLayout\_1.addWidget(self.ServerComboBox)  
 self.verticalLayout\_3.addLayout(self.horizontalLayout\_1)  
  
 # второй горизонтальный блок  
 self.horizontalLayout\_2 = QtWidgets.QHBoxLayout()  
 self.horizontalLayout\_2.setObjectName("horizontalLayout\_2")  
 self.label\_2 = QtWidgets.QLabel(self.centralwidget)  
 self.label\_2.setObjectName("label\_2")  
 self.horizontalLayout\_2.addWidget(self.label\_2)  
 # добавление поля логина  
 self.LoginEnter = QtWidgets.QLineEdit(self.centralwidget)  
 self.LoginEnter.setObjectName("LoginEnter")  
 self.horizontalLayout\_2.addWidget(self.LoginEnter)  
 self.verticalLayout\_3.addLayout(self.horizontalLayout\_2)  
  
 # третий горизонтальный блок  
 self.horizontalLayout\_3 = QtWidgets.QHBoxLayout()  
 self.horizontalLayout\_3.setObjectName("horizontalLayout\_3")  
 self.label\_3 = QtWidgets.QLabel(self.centralwidget)  
 self.label\_3.setObjectName("label\_3")  
 self.horizontalLayout\_3.addWidget(self.label\_3)  
 # добавление пустого разделяющего пространства  
 spacerItem = QtWidgets.QSpacerItem(98, 20, QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)  
 self.horizontalLayout\_3.addItem(spacerItem)  
 # добавление поля пароля  
 self.PasswordEnter = QtWidgets.QLineEdit(self.centralwidget)  
 self.PasswordEnter.setObjectName("PasswordEnter")  
 self.horizontalLayout\_3.addWidget(self.PasswordEnter)  
 self.verticalLayout\_3.addLayout(self.horizontalLayout\_3)  
  
 self.line = QtWidgets.QFrame(self.centralwidget)  
 self.line.setFrameShape(QtWidgets.QFrame.HLine)  
 self.line.setFrameShadow(QtWidgets.QFrame.Sunken)  
 self.line.setObjectName("line")  
 self.verticalLayout\_3.addWidget(self.line)  
  
 # четвёртый горизонтальный блок  
 self.horizontalLayout\_4 = QtWidgets.QHBoxLayout()  
 self.horizontalLayout\_4.setObjectName("horizontalLayout\_4")  
 self.label\_4 = QtWidgets.QLabel(self.centralwidget)  
 self.label\_4.setObjectName("label\_4")  
 self.horizontalLayout\_4.addWidget(self.label\_4)  
 # формирование кнопки "Добавить файл"  
 #self.AddFilePushButton = QtWidgets.QPushButton(self.centralwidget)  
 #self.AddFilePushButton.setObjectName("AddFilePushButton")  
 #self.horizontalLayout\_4.addWidget(self.AddFilePushButton)  
 # формирование кнопки "Удалить файл"  
 #self.DeleteFilePushButton = QtWidgets.QPushButton(self.centralwidget)  
 #self.DeleteFilePushButton.setObjectName("DeleteFilePushButton\_2")  
 #self.horizontalLayout\_4.addWidget(self.DeleteFilePushButton)  
 #self.verticalLayout\_2.addLayout(self.horizontalLayout\_4)  
 # формирование кнопки "Открыть"  
 self.OpenCSVFile = QtWidgets.QPushButton(self.centralwidget)  
 self.OpenCSVFile.setObjectName("OpenCSVFile")  
 self.horizontalLayout\_4.addWidget(self.OpenCSVFile)  
 self.verticalLayout\_3.addLayout(self.horizontalLayout\_4)  
 # формирование списка добавленных файлов  
 #self.FilesListWidget = QtWidgets.QListWidget(self.centralwidget)  
 #self.FilesListWidget.setObjectName("FilesListWidget")  
 #self.verticalLayout\_2.addWidget(self.FilesListWidget)  
 #self.verticalLayout\_3.addLayout(self.verticalLayout\_2)  
  
 # пятый горизонтальный блок  
 #self.horizontalLayout\_5 = QtWidgets.QHBoxLayout()  
 #self.horizontalLayout\_5.setObjectName("horizontalLayout")  
 #self.label\_5 = QtWidgets.QLabel(self.centralwidget)  
 #self.label\_5.setObjectName("label\_5")  
 #self.horizontalLayout\_5.addWidget(self.label\_5)  
 # добавление поля адресата  
 #self.AddressLineEdit = QtWidgets.QLineEdit(self.centralwidget)  
 #self.AddressLineEdit.setObjectName("AddressLineEdit")  
 #self.horizontalLayout\_5.addWidget(self.AddressLineEdit)  
 #self.verticalLayout\_3.addLayout(self.horizontalLayout\_5)  
  
 # шестой горизонтальный блок  
 self.horizontalLayout\_6 = QtWidgets.QHBoxLayout()  
 self.horizontalLayout\_6.setObjectName("horizontalLayout\_7")  
 self.label\_6 = QtWidgets.QLabel(self.centralwidget)  
 self.label\_6.setObjectName("label\_6")  
 self.horizontalLayout\_6.addWidget(self.label\_6)  
 # добавление поля для темы  
 self.ThemeBar = QtWidgets.QLineEdit(self.centralwidget)  
 self.ThemeBar.setObjectName("ThemeBar")  
 self.horizontalLayout\_6.addWidget(self.ThemeBar)  
 self.verticalLayout\_3.addLayout(self.horizontalLayout\_6)  
  
 self.verticalLayout\_1 = QtWidgets.QVBoxLayout()  
 self.verticalLayout\_1.setObjectName("verticalLayout")  
 # добавление поля сообщения  
 self.textEdit = QtWidgets.QTextEdit(self.centralwidget)  
 self.textEdit.setObjectName("textEdit")  
 self.verticalLayout\_1.addWidget(self.textEdit)  
 # добавление кнопки отправки  
 self.SendButton = QtWidgets.QPushButton(self.centralwidget)  
 self.SendButton.setObjectName("SendButton")  
 self.verticalLayout\_1.addWidget(self.SendButton)  
 self.verticalLayout\_3.addLayout(self.verticalLayout\_1)  
 ProjectWindow.setCentralWidget(self.centralwidget)  
  
 self.menubar = QtWidgets.QMenuBar(ProjectWindow)  
 self.menubar.setGeometry(QtCore.QRect(0, 0, 330, 21))  
 self.menubar.setObjectName("menubar")  
 ProjectWindow.setMenuBar(self.menubar)  
  
 self.statusbar = QtWidgets.QStatusBar(ProjectWindow)  
 self.statusbar.setObjectName("statusbar")  
 ProjectWindow.setStatusBar(self.statusbar)  
  
 self.namingInUi(ProjectWindow)  
 QtCore.QMetaObject.connectSlotsByName(ProjectWindow)  
  
  
 def namingInUi(self, ProjectWindow):  
 \_translate = QtCore.QCoreApplication.translate  
 ProjectWindow.setWindowTitle(\_translate("ProjectWindow", "ProjectWindow"))  
  
 self.label\_1.setText(\_translate("ProjectWindow", "Сервер"))  
 self.label\_2.setText(\_translate("ProjectWindow", "Адрес Отправителя"))  
 self.label\_3.setText(\_translate("ProjectWindow", "Пароль"))  
 #self.label\_4.setText(\_translate("ProjectWindow", "Список файлов"))  
 self.label\_4.setText(\_translate("ProjectWindow", "Выбрать CSV файл"))  
  
 #self.AddFilePushButton.setText(\_translate("ProjectWindow", "Выбрать файл"))  
 #self.DeleteFilePushButton.setText(\_translate("ProjectWindow", "Удалить файлы"))  
 self.OpenCSVFile.setText(\_translate("ProjectWindow", "Открыть"))  
  
 # ,self.label\_5.setText(\_translate("ProjectWindow", "Адрес получателя"))  
 self.label\_6.setText(\_translate("ProjectWindow", "ТЕМА"))  
  
 self.SendButton.setText(\_translate("ProjectWindow", "Отправить"))  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 import sys  
 app = QtWidgets.QApplication(sys.argv)  
 ProjectWindow = QtWidgets.QMainWindow()  
 ui = Ui\_ProjectWindow()  
 ui.setupUi(ProjectWindow)  
 ProjectWindow.show()  
 sys.exit(app.exec\_())

main.py – файл, отвечающий за отправку сообщений из формы

import smtplib  
from email.mime.multipart import MIMEMultipart  
from email.mime.text import MIMEText  
from email.mime.image import MIMEImage  
from email.mime.audio import MIMEAudio  
from email.mime.base import MIMEBase  
import email.encoders as encoders  
from PyQt5 import QtWidgets, QtCore  
import MailInterface  
import sys  
import os  
import mimetypes  
import csv  
  
class MailAPP(QtWidgets.QMainWindow, QtWidgets.QFileDialog, MailInterface.Ui\_ProjectWindow):  
 def \_\_init\_\_(self):  
 super().\_\_init\_\_()  
 self.setupUi(self)  
 self.setWindowTitle('Почта')  
 self.setFocus()  
 #self.FilesListWidget.setSelectionMode(QtWidgets.QAbstractItemView.MultiSelection)  
 self.myEMAIL = ''  
 self.myPassword = ''  
 self.PasswordEnter.setEchoMode(QtWidgets.QLineEdit.Password)  
 services = ['gmail.com', 'mail.ru', 'yandex.ru']  
 self.servers = {'gmail.com': 'smtp.gmail.com:587',  
 'mail.ru': 'smtp.yandex.ru:467',  
 'yandex.ru': 'smtp.mail.ru:25'}  
 # выбор сервера  
 for service in services:  
 self.ServerComboBox.addItem(service)  
 self.SendButton.clicked.connect(self.sendMail) # привязка кпоки "Отправить" к функции sendMail  
 #self.AddFilePushButton.clicked.connect(self.addFile) # привязка кпоки "Выбрать файл" к функции addFile  
 #self.DeleteFilePushButton.clicked.connect(self.rmvFiles) # привязка кпоки "Удалить файлы" к функции rmvFiles  
 self.OpenCSVFile.clicked.connect(self.open\_file) # привязка кпоки "Удалить файлы" к функции rmvFiles  
 self.msgBox = QtWidgets.QMessageBox()  
 self.csvFile = ''  
  
  
 # реализация функции диалогового окна  
 def showMsgBox(self, msg: str):  
 self.msgBox.setText(msg)  
 self.msgBox.exec\_()  
  
  
 # реализация функции добавить файл  
 def open\_file(self):  
 fileName = QtWidgets.QFileDialog.getOpenFileName(parent=self,  
 caption=u'Открыть файл',  
 directory='E:/',  
 filter='csv (\*.csv);;AllFiles (\*)')[0]  
 if fileName: # Если выбран файл  
 self.csvFile = fileName  
 self.label\_6.setText(os.path.basename(self.csvFile))  
  
 def csvReader(self):  
 receiver = {}  
 with open(self.csvFile, 'r') as csvFile:  
 csvReader = csv.reader(csvFile)  
 for row in csvReader:  
 to = row[0].split(';')[0]  
 file = row[0].split(';')[1:]  
 receiver[to] = file  
 print(receiver)  
 return receiver  
  
  
 # реализация функции добавить файл  
 #def addFile(self):  
 # fileName = QtWidgets.QFileDialog.getOpenFileName(parent=self,  
 # caption=u'Открыть файл',  
 # directory='E:/',  
 # filter='AllFiles (\*)')[0]  
 # if fileName: # Если выбран файл  
 # # Проверка на наличие одинаковых файлов  
 # if len(self.FilesListWidget.findItems(fileName, QtCore.Qt.MatchExactly)) == 0:  
 # self.FilesListWidget.addItem(fileName)  
 # else:  
 # self.showMsgBox('Файл уже добавлен!')  
  
 # реализация функции удалить файл  
 #def rmvFiles(self):  
 # for item in self.FilesListWidget.selectedItems():  
 # self.FilesListWidget.takeItem(self.FilesListWidget.row(item))  
  
  
 def processAttachment(self, files: list):  
 filesList = []  
 for file in files:  
 fileName = ''  
 try:  
 fileName = os.path.basename(file)  
 ctype, encoding = mimetypes.guess\_type(file)  
 if ctype is None or encoding is not None:  
 ctype = 'application/octet-stream'  
 maintype, subtype = ctype.split('/', 1)  
 if maintype == 'text':  
 with open(file) as fp:  
 file = MIMEText(fp.read(), \_subtype=subtype)  
 fp.close()  
 elif maintype == 'image':  
 with open(file, 'rb') as fp:  
 file = MIMEImage(fp.read(), \_subtype=subtype)  
 fp.close()  
 elif maintype == 'audio':  
 with open(file, 'rb') as fp:  
 file = MIMEAudio(fp.read(), \_subtype=subtype)  
 fp.close()  
 else:  
 with open(file, 'rb') as fp:  
 file = MIMEBase(maintype, subtype)  
 file.set\_payload(fp.read())  
 fp.close()  
 encoders.encode\_base64(file)  
 file.add\_header('Content-Disposition', 'attachment', filename=fileName)  
 filesList.append(file)  
 except:  
 self.showMsgBox('Ошибка добавления файла: ' + fileName)  
 return filesList  
  
  
 # реализация функции отправить сообщение  
 def sendMail(self):  
 try:  
 print()  
 if self.LoginEnter.text().split('@')[1] != self.ServerComboBox.currentText():  
 self.showMsgBox('Введенная почта и сервер не совпадают!')  
 else:  
 self.smtpServer = smtplib.SMTP(self.servers[self.ServerComboBox.currentText()]) # Создание объекта SMTP  
 self.myEMAIL = self.LoginEnter.text()  
 self.myPassword = self.PasswordEnter.text()  
 self.smtpServer.starttls() # Шифрование  
 self.smtpServer.login(self.myEMAIL, self.myPassword) # Ввод учетных данных  
  
 for k, v in self.csvReader().items():  
 myMessage = MIMEMultipart() # Создание объекта стандарта MIME  
 myMessage['Subject'] = self.ThemeBar.text() # Определение темы сообщения  
 myMessage['From'] = self.myEMAIL # Определение почты отправки  
 myMessage['To'] = k # Определение адресата  
 messageContent = self.textEdit.toPlainText() # Получение текста с формы  
 myMessage.attach(MIMEText(messageContent, 'plain')) # Добавление текста в сообещение  
  
 for file in self.processAttachment(v):  
 myMessage.attach(file)  
 self.smtpServer.send\_message(myMessage) # Отправка сообщения  
 except:  
 self.showMsgBox('Возникла не предвиденная ошибка')  
  
  
def main():  
 app = QtWidgets.QApplication(sys.argv)  
 window = MailAPP()  
 window.show()  
 sys.exit(app.exec\_())  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

fileDialog.py – файл, отвечающий за отправку сообщений из формы

import sys  
from PyQt5.QtWidgets import (QMainWindow, QTextEdit,  
 QAction, QFileDialog, QApplication)  
from PyQt5.QtGui import QIcon  
  
class Example(QMainWindow):  
  
  
 def \_\_init\_\_(self):  
 super().\_\_init\_\_()  
  
 self.initUI()  
  
  
 def initUI(self):  
  
 self.textEdit = QTextEdit()  
 self.setCentralWidget(self.textEdit)  
 self.statusBar()  
  
 openFile = QAction(QIcon('open.png'), 'Open', self)  
 openFile.setShortcut('Ctrl+O')  
 openFile.setStatusTip('Open new File')  
 openFile.triggered.connect(self.showDialog)  
  
 menubar = self.menuBar()  
 fileMenu = menubar.addMenu('&File')  
 fileMenu.addAction(openFile)  
  
 self.setGeometry(300, 300, 350, 300)  
 self.setWindowTitle('File dialog')  
 self.show()  
  
  
 def showDialog(self):  
  
 fname = QFileDialog.getOpenFileName(self, 'Open file', 'E:/test-doc.txt')[0]  
  
 f = open(fname, 'r')  
  
 with f:  
 data = f.read()  
 self.textEdit.setText(data)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
  
 app = QApplication(sys.argv)  
 ex = Example()  
 sys.exit(app.exec\_())

# Скриншоты

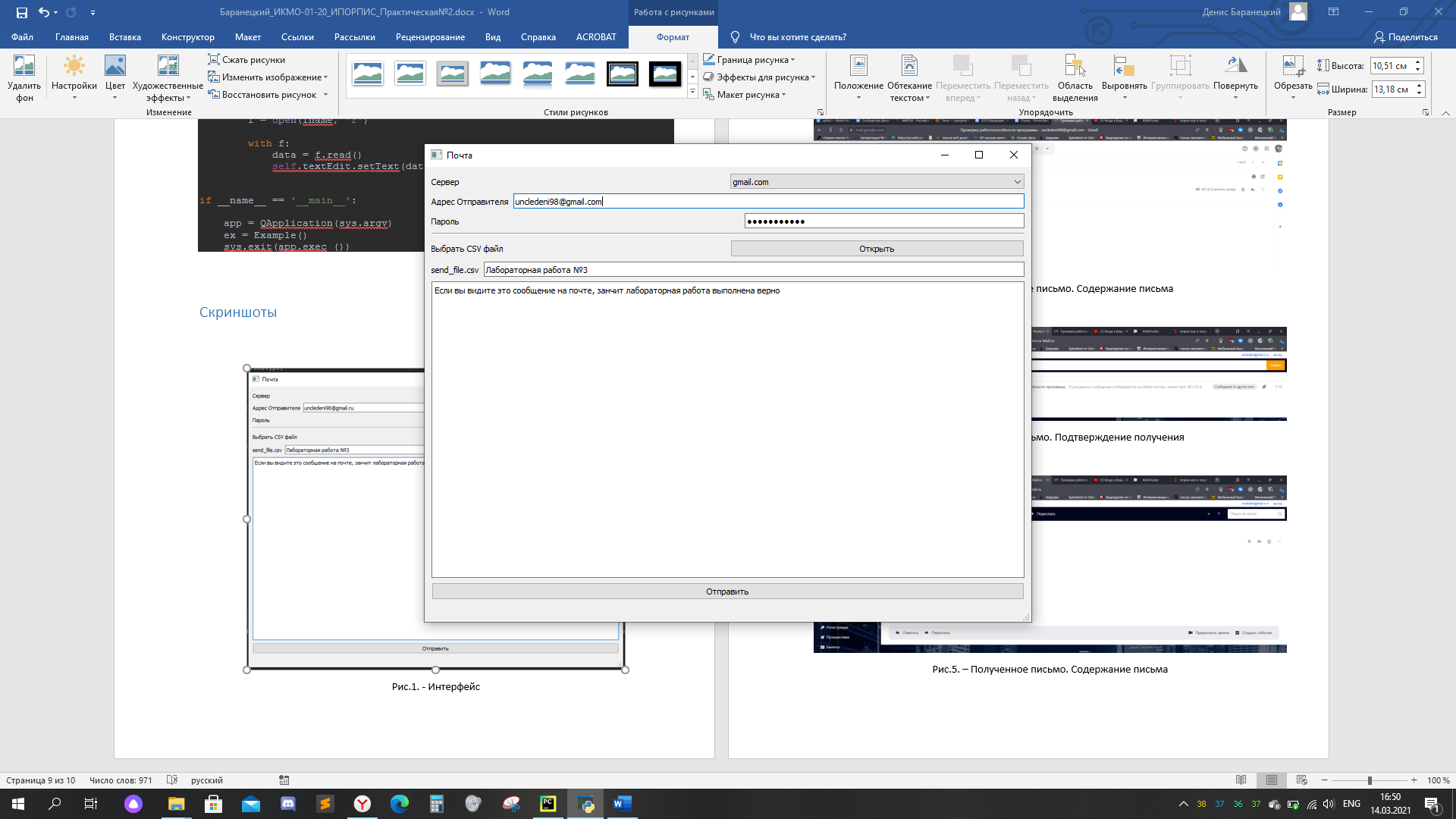


Рис.1. - Интерфейс

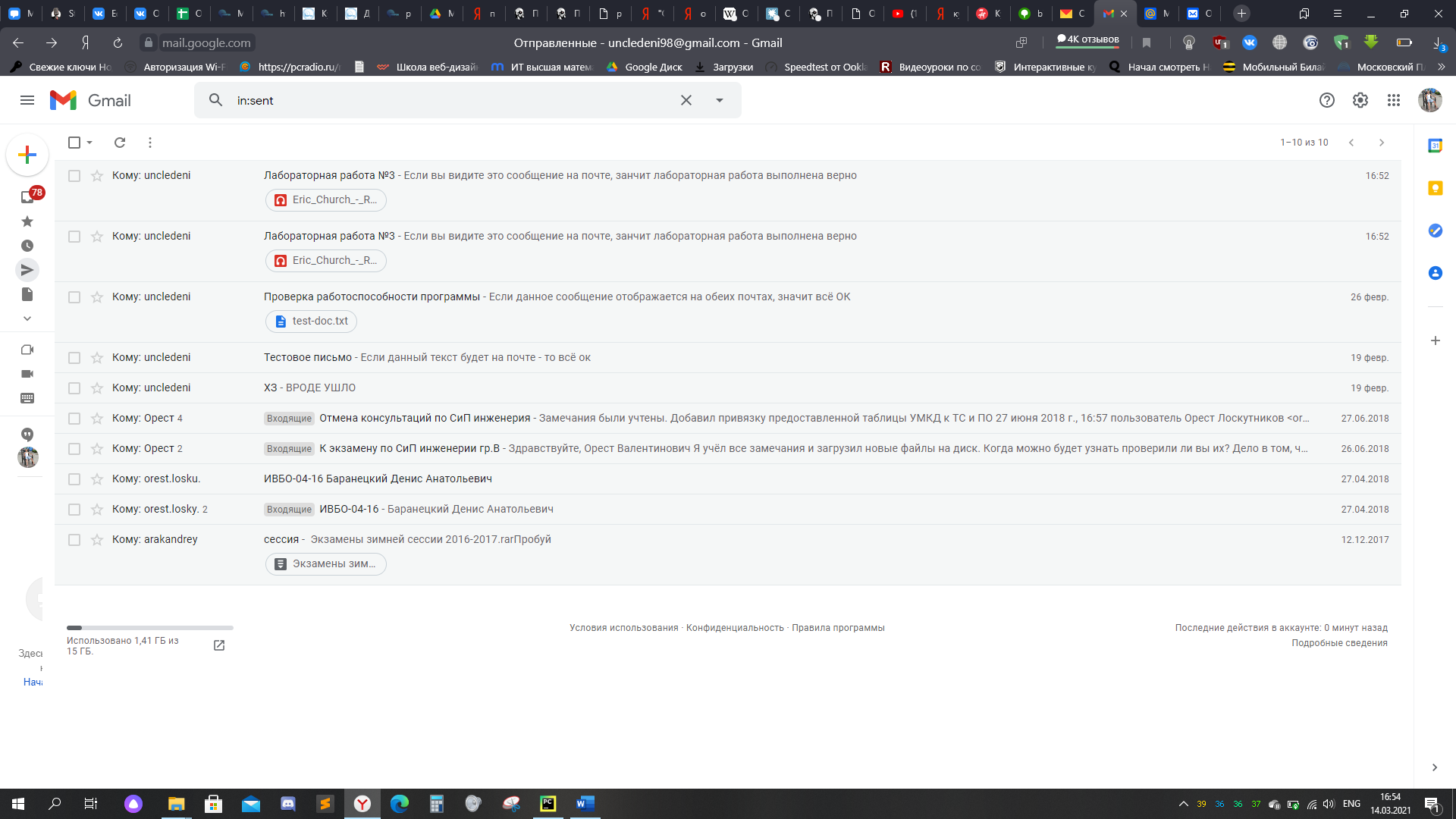


Рис.2. – Отправленные письма

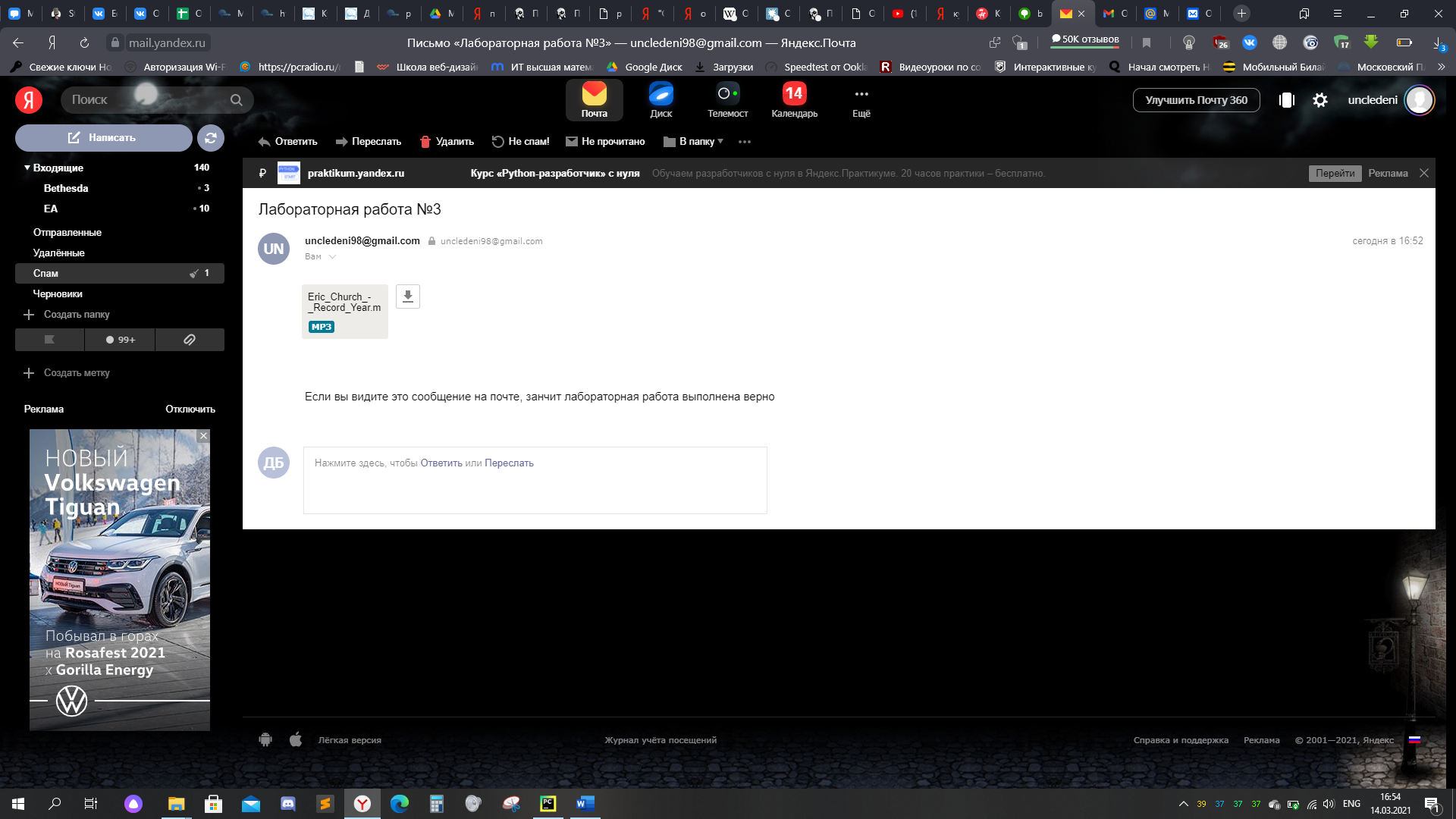


Рис.3. – Полученное письмо. Яндекс почта

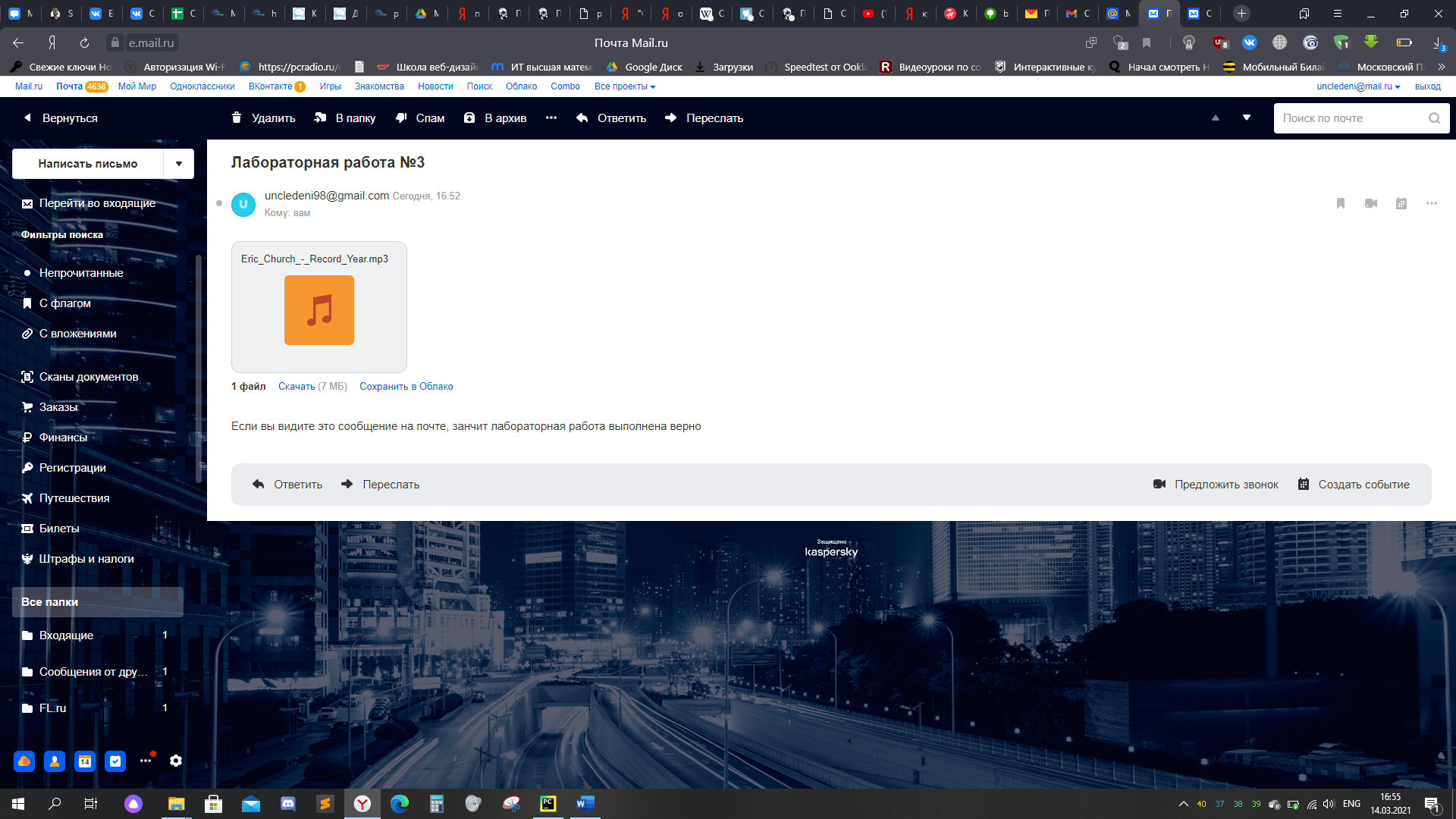


Рис.4. – Полученное письмо. Mail.ru

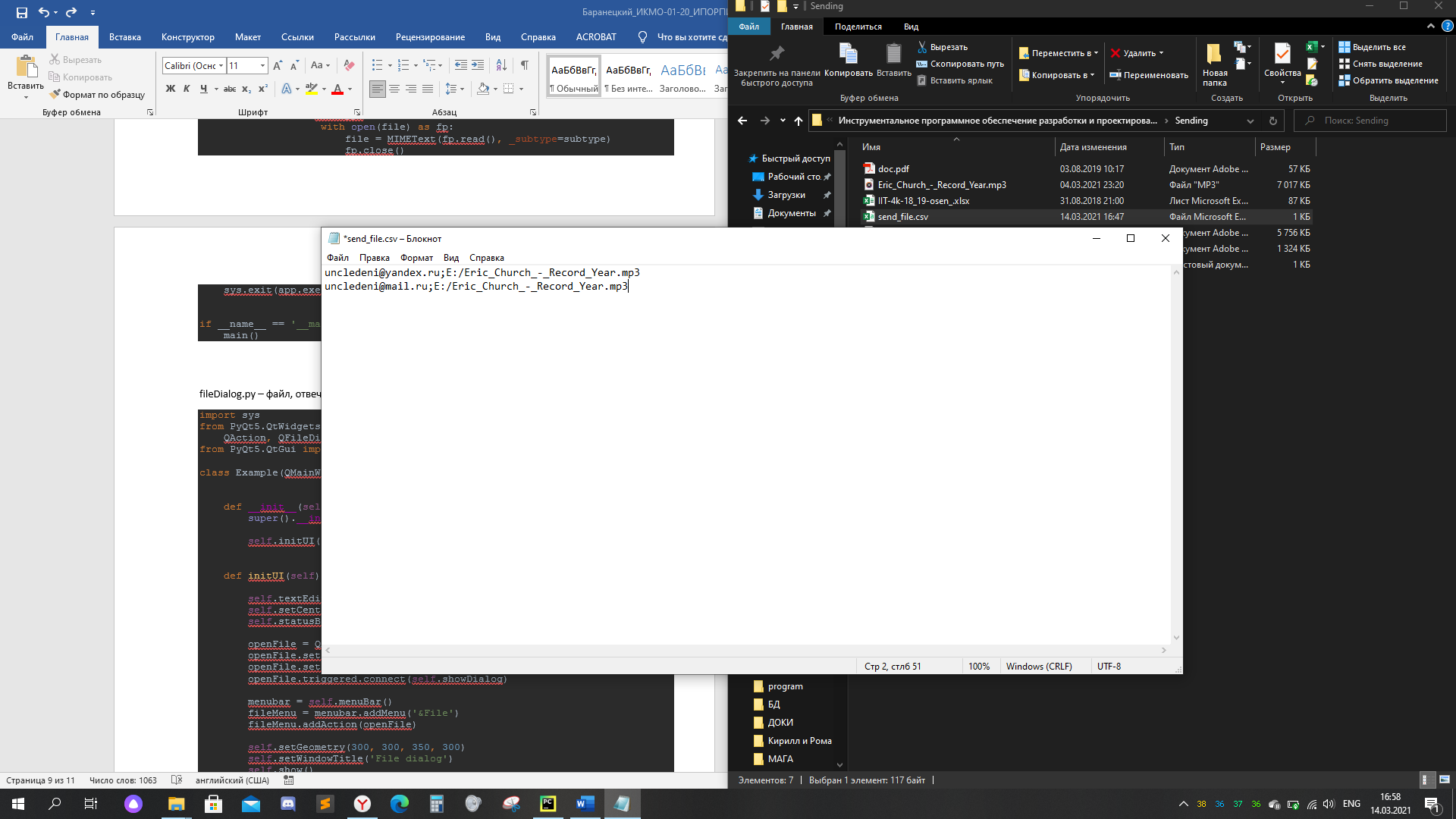


Рис.5. – csv файл