**ПРИЛОЖЕНИЕ**

**Листинг программы**

#ИМПОРТ БИБЛИОТЕК

from PyQt5 import QtCore, QtGui, QtWidgets

from PyQt5.QtGui import (QBrush, QColor, QConicalGradient, QCursor, QFont,

QFontDatabase, QIcon, QKeySequence, QLinearGradient, QPalette, QPainter,

QPixmap, QRadialGradient)

from PyQt5.QtWidgets import QApplication,QLineEdit, QMainWindow, QGridLayout, QWidget, QTableWidget, QTableWidgetItem, QComboBox

from PyQt5.QtCore import QSize, Qt

from PyQt5.Qt import QMainWindow, QVBoxLayout, QWidget, QLabel, QAction, Qt, QMessageBox, QApplication

from PyQt5.QtCore import \*

from PyQt5.QtWidgets import \*

class Window2(QWidget): #КЛАСС ОКНА ОТЧЁТА

def \_\_init\_\_(self): #ФУНКЦИЯ СОЗДАНИЯ ИНТЕРФЕЙСА

super(Window2, self).\_\_init\_\_()

self.setWindowTitle('Отчет')#ЗАГОЛОВОК ОКНА

self.setMinimumWidth(765) #ШИРИНА ОКНА

self.setMinimumHeight(620) #ВЫСОТА ОКНА

#СОЗДАНИЕ ТАБЛИЦЫ ОТЧЁТА

self.tableWidget3 = QtWidgets.QTableWidget(self) #ТАБЛИЦУ ПОМЕСТИЛИ НА WINDOW2 (ОКНО)

self.tableWidget3.setGeometry(QtCore.QRect(10, 10, 548, 604)) #ЗАДАЛИ РАСПОЛОЖЕНИЕ И РАЗМЕР

self.tableWidget3.setRowCount(20) #ЗАДАЛИ КОЛИЧЕСТВО СТРОК

self.tableWidget3.setColumnCount(7)#ЗАДАЛИ КОЛИЧЕСТВО СТОЛБЦОВ

self.tableWidget3.horizontalHeader().hide() #СКРЫЛИ ГОРИЗОНТАЛЬНЫЕ ШАПКИ (ГДЕ ВРЕМЯ)

self.tableWidget3.verticalHeader().hide() #СКРЫЛИ ВЕРТИКАЛЬНЫЕ ШАПКИ (ГДЕ ТИП САМОЛЁТОВ)

self.tableWidget3.setItem(0, 0, QTableWidgetItem("№")) #ВСТАВЛЯЕМ С 0 СТОЛБЦОМ И 0 СТРОКОЙ "№"

self.tableWidget3.setItem(0, 3, QTableWidgetItem("1")) #-//- (1- СТРОКА, 2- СТОЛБЕЦ)

self.tableWidget3.setItem(0, 4, QTableWidgetItem("2")) #-//-

self.tableWidget3.setItem(0, 5, QTableWidgetItem("3")) #-//-

self.tableWidget3.setItem(0, 6, QTableWidgetItem("4")) #-//-

self.tableWidget3.setItem(0, 7, QTableWidgetItem("5")) #-//-

self.tableWidget3.setItem(1, 3, QTableWidgetItem("SSJ-100 ")) #-//-

self.tableWidget3.setItem(1, 4, QTableWidgetItem("Airbus A310")) #-//-

self.tableWidget3.setItem(1, 5, QTableWidgetItem("ИЛ-96М")) #-//-

self.tableWidget3.setItem(1, 6, QTableWidgetItem("Airbus A320")) #-//-

self.tableWidget3.setItem(2, 3, QTableWidgetItem("1")) #-//-

self.tableWidget3.setItem(2, 4, QTableWidgetItem("1")) #-//-

self.tableWidget3.setItem(2, 5, QTableWidgetItem("1")) #-//-

self.tableWidget3.setItem(2, 6, QTableWidgetItem("2")) #-//

self.tableWidget3.setVerticalScrollBarPolicy(Qt.ScrollBarAlwaysOff) #СКРЫВАЕМ СКРОЛЛ ДЛЯ КРАСОТЫ

self.tableWidget3.setSpan(4,0,20,1) #ОБЪЕДИНЯЕМ ЯЧЕЙКИ С 4 СТРОКИ 0ГО СТОЛБЦА ПО 20 СТРОКУ ШИРИНОЙ В 1 СТОЛБЕЦ

self.tableWidget3.setSpan(4,1,9,1) #ОБЪЕДИНЯЕМ ЯЧЕЙКИ С 4 СТРОКИ 1ГО СТОЛБЦА ПО 9 СТРОКУ ШИРИНОЙ В 1 СТОЛБЕЦ

pic = QtGui.QPixmap("data/22.bmp") #ВСТАВЛЯЕМ В ПЕРЕМЕННУЮ ВЕРТИКАЛЬНУЮ КАРТИНКУ, ПОТОМУ ЧТО В ПИТОНЕ НЕЛЬЗЯ ПИСАТЬ ВЕРТИКАЛЬНО

self.label2 = QtWidgets.QLabel(self)#СОЗДАЁМ LABEL (МЕТКА)

self.label2.setPixmap(pic) #ВСТАВЛЯЕМ В LABEL ПЕРЕМЕННУЮ С КАРТИНКОЙ

self.tableWidget3.setCellWidget(4, 0, self.label2) #LABEL С ПЕРЕМЕННОЙ, В КОТОРОЙ КАРТИНКА МЫ ЭТО ВСЁ ВСТАВЛЯЕМ В ЯЧЕЙКУ

self.tableWidget3.setItem(4, 0, QTableWidgetItem("0"))#Вставляем в эту ячейку нолик как заглушку, чтобы потом можно было бы закрасить ячейку в цвет

pic = QtGui.QPixmap("data/23.bmp")#АНАЛОГИЧНО

self.label2 = QtWidgets.QLabel(self)#АНАЛОГИЧНО

self.label2.setPixmap(pic)#АНАЛОГИЧНО

self.tableWidget3.setCellWidget(4, 1, self.label2)#АНАЛОГИЧНО

self.tableWidget3.setItem(4, 1, QTableWidgetItem("0"))#АНАЛОГИЧНО

pic = QtGui.QPixmap("data/24.bmp")#АНАЛОГИЧНО

self.label2 = QtWidgets.QLabel(self)#АНАЛОГИЧНО

self.label2.setPixmap(pic)#АНАЛОГИЧНО

self.tableWidget3.setCellWidget(13, 1, self.label2)#АНАЛОГИЧНО

self.tableWidget3.setItem(13, 1, QTableWidgetItem("0"))#АНАЛОГИЧНО

self.tableWidget3.setSpan(13,1,20,1)#опять соединяем ячейки

self.tableWidget3.setSpan(0,0,1,3)#опять соединяем ячейки

self.tableWidget3.setItem(4, 2, QTableWidgetItem("Всего исправных самолетов"))#опять заполняем ячейки заголовками вручную

self.tableWidget3.setItem(5, 2, QTableWidgetItem("Время в рейсе"))#опять заполняем ячейки заголовками вручную

self.tableWidget3.setItem(6, 2, QTableWidgetItem("Обеспечение вылета"))#опять заполняем ячейки заголовками вручную

self.tableWidget3.setItem(7, 2, QTableWidgetItem("В резерве"))#опять заполняем ячейки заголовками вручную

self.tableWidget3.setItem(8, 2, QTableWidgetItem("Простой по метео - условиям и запретам"))#опять заполняем ячейки заголовками вручную

self.tableWidget3.setItem(9, 2, QTableWidgetItem("Исправные, не совершающие полётов"))#опять заполняем ячейки заголовками вручную

self.tableWidget3.setItem(10, 2, QTableWidgetItem("На техническом обслуживании"))#опять заполняем ячейки заголовками вручную

self.tableWidget3.setItem(11, 2, QTableWidgetItem("В ремонте"))#опять заполняем ячейки заголовками вручную

self.tableWidget3.setItem(12, 2, QTableWidgetItem("Ожидание ремонта"))#опять заполняем ячейки заголовками вручную

self.tableWidget3.setItem(13, 2, QTableWidgetItem("Восстановление после повреждения"))#опять заполняем ячейки заголовками вручную

self.tableWidget3.setItem(14, 2, QTableWidgetItem("Отсутствие запасных частей"))#опять заполняем ячейки заголовками вручную

self.tableWidget3.setItem(15, 2, QTableWidgetItem("Отсутствие двигателей"))#опять заполняем ячейки заголовками вручную

self.tableWidget3.setItem(16, 2, QTableWidgetItem("Доработки по бюллетеням"))#опять заполняем ячейки заголовками вручную

self.tableWidget3.setItem(17, 2, QTableWidgetItem("Рекламация промышленности"))#опять заполняем ячейки заголовками вручную

self.tableWidget3.setItem(18, 2, QTableWidgetItem("Рекламация ремонтным заводом"))#опять заполняем ячейки заголовками вручную

self.tableWidget3.setItem(19, 2, QTableWidgetItem("Ожидание списания"))#опять заполняем ячейки заголовками вручную

self.tableWidget3.setEditTriggers(QtWidgets.QTableWidget.NoEditTriggers)#ставим свойство таблицы "No edit", чтобы в проге ее нельзя было менять и вписывать туда всяку фигню

self.tableWidget3.resizeColumnsToContents()#функция подгонки ширины колонок под содержимое внутри

self.tableWidget3.setSpan(1,0,1,3)#опять соединяем ячейки

self.tableWidget3.setItem(1, 0, QTableWidgetItem("Тип самолета"))#опять заполняем ячейки заголовками вручную

self.tableWidget3.setSpan(2,0,1,3)#опять соединяем ячейки

self.tableWidget3.setItem(2, 0, QTableWidgetItem("Количество самолетов"))#опять заполняем ячейки заголовками вручную

self.tableWidget3.setSpan(3,0,1,3)#опять соединяем ячейки

self.tableWidget3.setItem(3, 0, QTableWidgetItem("Календарный фонд времемни самолето-час"))#опять заполняем ячейки заголовками вручную

for j in range(8):#цикл с помощью которого мы будем закрашивать наши заголовки в столбцах

if j == 1 or j==2:#если j = 1 или 2, то пропускаем итерацию

p = 0 #заглушка

elif j == 7:#если j = 7, то пропускаем итерацию

p = 0 #заглушка

else:

self.tableWidget3.item(0, j).setBackground(QtGui.QColor(219,235,255))#красим выбранную ячейку в синий

self.tableWidget3.item(1, j).setBackground(QtGui.QColor(219,235,255))#красим выбранную ячейку в синий

self.tableWidget3.item(2, 0).setBackground(QtGui.QColor(219,235,255))#красим выбранную ячейку в синий

self.tableWidget3.item(3, 0).setBackground(QtGui.QColor(219,235,255))#красим выбранную ячейку в синий

self.tableWidget3.item(13, 1).setBackground(QtGui.QColor(219,235,255))#красим выбранную ячейку в синий

self.tableWidget3.item(4, 0).setBackground(QtGui.QColor(219,235,255))#красим выбранную ячейку в синий

self.tableWidget3.item(4, 1).setBackground(QtGui.QColor(219,235,255))#красим выбранную ячейку в синий

for j in range(21):#цикл с помощью которого мы будем закрашивать наши заголовки в строках

if j>=4 and j<20:#если j больше или равно 4, и меньше 20

self.tableWidget3.item(j, 2).setBackground(QtGui.QColor(219,235,255))#красим выбранную ячейку в синий

self.groupBox2 = QtWidgets.QGroupBox(self)#создаем групп бокс для того, чтобы закинуть туда все наши кнопки поля ввода и тд

self.groupBox2.setGeometry(QtCore.QRect(565, 0, 200, 160))#подгоняем размер

font = QtGui.QFont()#переменная шрифта

font.setFamily("Segoe Print")#ставим шрифту его стиль

font.setPointSize(10)#размер

font.setBold(True)#толстым делаем

font.setWeight(75)#ширина??

self.groupBox2.setFont(font)#вот че у нас сверху получилось со шрифтом, устанавливаем его для групп бокс

self.groupBox2.setTitle("Настройки")#заголовок груп бокса

self.qlabel2 = QLabel(self.groupBox2)#создаем лейбл внутри групп бокса

self.qlabel2.move(10,30)#двигаем лейбл

self.qlabel2.setText('Месяц:')#вставляем в лейбл текст

self.combo2 = QComboBox(self.groupBox2)#создаем комбо бокс (выдвигающийся список)

self.combo2.addItem("Январь")#добавляем пункт в комбо бокс

self.combo2.addItem("Февраль")#добавляем пункт в комбо бокс

self.combo2.addItem("Март")#добавляем пункт в комбо бокс

self.combo2.addItem("Апрель")#добавляем пункт в комбо бокс

self.combo2.addItem("Май")#добавляем пункт в комбо бокс

self.combo2.addItem("Июнь")#добавляем пункт в комбо бокс

self.combo2.addItem("Июль")#добавляем пункт в комбо бокс

self.combo2.addItem("Август")#добавляем пункт в комбо бокс

self.combo2.addItem("Сентябрь")#добавляем пункт в комбо бокс

self.combo2.addItem("Октябрь")#добавляем пункт в комбо бокс

self.combo2.addItem("Ноябрь")#добавляем пункт в комбо бокс

self.combo2.addItem("Декабрь")#добавляем пункт в комбо бокс

self.combo2.move(80, 30)#двигаем комбо бокс

self.qlabel3 = QLabel(self.groupBox2)#аналогично с прошлым лейблом

self.qlabel3.move(10,60)#аналогично с прошлым лейблом

self.qlabel3.setText('Год:')#аналогично с прошлым лейблом

self.line = QLineEdit(self.groupBox2)#создаем поле ввода внутри групп бокса

self.line.move(41, 60)#двигаем поле ввода

self.pushButton = QtWidgets.QPushButton(self.groupBox2)#жесть делаем кнопку

self.pushButton.setGeometry(QtCore.QRect(10, 100, 169, 31))#редактируем размер

self.pushButton.clicked.connect(self.tab)#коннектим кнопку к функции с заполнением отчета

self.pushButton.setText('ПРОСМОТР')#текст на кнопке

#функция заполнения отчета

def tab(self):

year=self.line.text()#считываем введенный год из поля ввода

mmm = str(self.combo2.currentIndex()+1)#считываем из комбо бокса выбранный месяц

try:

#это ввсё переменные для подсчета двух самолетов ан-2

#приравниваем их заранее к нулю

dvsegoall = 0

dvsisprall = 0

dvpolall = 0

dobspall = 0

dmeteoall = 0

drezrvall = 0

disneisall = 0

dvsneisprall = 0

dustrnall = 0

dozhoball = 0

dobslall = 0

dozhperall = 0

dperall = 0

dnochall = 0

dvremntall = 0

dozhrall = 0

dvossprall = 0

dotszpall = 0

dotsdvall = 0

ddorbulall = 0

drekprall = 0

drekremzall = 0

dozhspall = 0

for samin in range(5):#цикл, чтобы из файла брать строчки по очереди

#а это ввсё переменные для подсчета основных самолетов

#приравниваем их заранее к нулю

vsegoall = 0 #всего количество часов

vsisprall = 0 #всего исправных часов

vpolall = 0 #часов в полете

obspall = 0 #и дальше по таблице там долго расписывать...

meteoall = 0

rezrvall = 0

isneisall = 0

vsneisprall = 0

ustrnall = 0

ozhoball = 0

obslall = 0

ozhperall = 0

perall = 0

nochall = 0

vremntall = 0

ozhrall = 0

vossprall = 0

otszpall = 0

otsdvall = 0

dorbulall = 0

rekprall = 0

rekremzall = 0

ozhspall = 0

samin = samin+1 #индекс строчки на один увеличиваем, потому что в файле перва строка пустая :(

for i in range(31):#запускаем цикл, чтобы мы провелили все 31 день в месяце

i = i+1#прибавил однерку, потому что месяцы не начинаются с нуля нифига се

ii = i#запомнил еще раз это число

if i<10:#если день в месяце меньше 10

dddd = "0"+str(i)#прибавляем ко дню вначале ноль, потому что в файле там нолик вначале идет у таких дней :(

with open("base/"+dddd+"-0"+mmm+"-"+year, 'r') as f:#открываем файл с нашей введенной датой

text = f.read().split("\n")#считываем содержимое и делим их по строкам

sp = text[samin].split(' ')#создаю отдельный список из прошлого, из которого мы выбрали одну строку по индексу самолета(первый самолет - первоя строка и т.д.)

vsego = 0#переменная первой строки в отчете "всего количество часов" приравниваю к нулю

for i in sp:#берем каждый элемент из созданного списка по отдельности

if i!='0' and i!='':#если элемент равен нулю или пустоте, то прибавляем единичку к нашей прошлой переменной

vsego+=1 #прибавили йооу

vsegoall += vsego#че в итоге получилось, еще приравниваем к другой переменной, которая подсчитывает общее количество часов выбранной строки отчета за ВЕСЬ месяц, а не за один день

vsego = str((vsego\*10)//60)+"ч. "+str((vsego\*10)%60)+"м."#хитрый алгоримт, который переводит нашу переменную в часы и минуты

vsispr = 0#АНАЛОГИЧНЫЕ АЛГОРИТМЫ

for i in sp: #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

if i=='1' or i=='2'or i=='3'or i=='4'or i=='5': #АНАЛОГИЧНЫЕ АЛГОРИТМЫ ТОЛЬКО ДЛЯ ДРУГОЙ СТРОКИ ТАБЛИЦЫ ОТЧЕТА!!!!!!

vsispr+=1 #АНАЛОГИЧНЫЕ АЛГОРИТМЫ ТОЛЬКО ДЛЯ ДРУГОЙ СТРОКИ ТАБЛИЦЫ

vsisprall += vsispr#АНАЛОГИЧНЫЕ АЛГОРИТМЫ ТОЛЬКО ДЛЯ ДРУГОЙ СТРОКИ ТАБЛИЦЫ

vsispr = str((vsispr\*10)//60)+"ч. "+str((vsispr\*10)%60)+"м."#АНАЛОГИЧНЫЕ АЛГОРИТМЫ ТОЛЬКО ДЛЯ ДРУГОЙ СТРОКИ ТАБЛИЦЫ

vpol = 0 #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

for i in sp:#АНАЛОГИЧНЫЕ АЛГОРИТМЫ

if i=='1':#АНАЛОГИЧНЫЕ АЛГОРИТМЫ

vpol+=1#АНАЛОГИЧНЫЕ АЛГОРИТМЫ

vpolall += vpol#АНАЛОГИЧНЫЕ АЛГОРИТМЫ

vpol = str((vpol\*10)//60)+"ч. "+str((vpol\*10)%60)+"м." #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

obsp = 0 #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

for i in sp: #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

if i=='2': #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

obsp+=1 #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

obspall += obsp #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

obsp = str((obsp\*10)//60)+"ч. "+str((obsp\*10)%60)+"м." #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

meteo = 0 #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

for i in sp: #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

if i=='3': #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

meteo+=1 #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

meteoall += meteo #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

meteo = str((meteo\*10)//60)+"ч. "+str((meteo\*10)%60)+"м." #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

rezrv = 0 #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

for i in sp: #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

if i=='4': #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

rezrv+=1 #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

rezrvall += rezrv #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

rezrv = str((rezrv\*10)//60)+"ч. "+str((rezrv\*10)%60)+"м." #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

isneis = 0 #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

for i in sp: #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

if i=='5': #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

isneis+=1 #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

isneisall += isneis #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

isneis = str((isneis\*10)//60)+"ч. "+str((isneis\*10)%60)+"м." #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

vsneispr = 0 #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

for i in sp: #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

if i!='1' and i!='2'and i!='3'and i!='4'and i!='5'and i!='0'and i!='': #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

vsneispr+=1 #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

vsneispr = str((vsneispr\*10)//60)+"ч. "+str((vsneispr\*10)%60)+"м." #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

ustrn = 0 #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

for i in sp: #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

if i=='6': #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

ustrn+=1 #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

ustrn = str((ustrn\*10)//60)+"ч. "+str((ustrn\*10)%60)+"м." #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

ozhob = 0 #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

for i in sp: #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

if i=='7': #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

ozhob+=1 #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

ozhob= str((ozhob\*10)//60)+"ч. "+str((ozhob\*10)%60)+"м." #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

obsl = 0 #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

for i in sp: #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

if i=='8': #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

obsl+=1 #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

obsl = str((obsl\*10)//60)+"ч. "+str((obsl\*10)%60)+"м." #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

ozhper = 0 #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

for i in sp: #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

if i=='9': #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

ozhper+=1 #АНАЛОГИЧНЫЕ АЛГОРИТМЫ

ozhper = str((ozhper\*10)//60)+"ч. "+str((ozhper\*10)%60)+"м."

#АНАЛОГИЧНЫЕ АЛГОРИТМЫ

per = 0

for i in sp:

if i=='10':

per+=1

perall += per

per = str((per\*10)//60)+"ч. "+str((per\*10)%60)+"м."

#АНАЛОГИЧНЫЕ АЛГОРИТМЫ

noch = 0

for i in sp:

if i=='11':

noch+=1

noch = str((noch\*10)//60)+"ч. "+str((noch\*10)%60)+"м."

#АНАЛОГИЧНЫЕ АЛГОРИТМЫ

ozhr = 0

for i in sp:

if i=='12':

ozhr+=1

ozhrall += ozhr

ozhr = str((ozhr\*10)//60)+"ч. "+str((ozhr\*10)%60)+"м."

#АНАЛОГИЧНЫЕ АЛГОРИТМЫ

vremnt = 0

for i in sp:

if i=='13':

vremnt+=1

vremntall += vremnt

vremnt = str((vremnt\*10)//60)+"ч. "+str((vremnt\*10)%60)+"м."

#АНАЛОГИЧНЫЕ АЛГОРИТМЫ

otszp = 0

for i in sp:

if i=='14':

otszp+=1

otszpall += otszp

otszp = str((otszp\*10)//60)+"ч. "+str((otszp\*10)%60)+"м."

#АНАЛОГИЧНЫЕ АЛГОРИТМЫ

otsdv = 0

for i in sp:

if i=='15':

otsdv+=1

otsdvall += otsdv

otsdv = str((otsdv\*10)//60)+"ч. "+str((otsdv\*10)%60)+"м."

#АНАЛОГИЧНЫЕ АЛГОРИТМЫ

dorbul = 0

for i in sp:

if i=='16':

dorbul+=1

dorbulall += dorbul

dorbul = str((dorbul\*10)//60)+"ч. "+str((dorbul\*10)%60)+"м."

#АНАЛОГИЧНЫЕ АЛГОРИТМЫ

rekpr = 0

for i in sp:

if i=='17':

rekpr+=1

rekprall += rekpr

rekpr = str((rekpr\*10)//60)+"ч. "+str((rekpr\*10)%60)+"м."

#АНАЛОГИЧНЫЕ АЛГОРИТМЫ

rekremz = 0

for i in sp:

if i=='18':

rekremz+=1

rekremzall += rekremz

rekremz = str((rekremz\*10)//60)+"ч. "+str((rekremz\*10)%60)+"м."

#АНАЛОГИЧНЫЕ АЛГОРИТМЫ

rasspr = 0

for i in sp:

if i=='19':

rasspr+=1

rasspr = str((rasspr\*10)//60)+"ч. "+str((rasspr\*10)%60)+"м."

#АНАЛОГИЧНЫЕ АЛГОРИТМЫ

vosspr = 0

for i in sp:

if i=='20':

vosspr+=1

vossprall += vosspr

vosspr = str((vosspr\*10)//60)+"ч. "+str((vosspr\*10)%60)+"м."

#АНАЛОГИЧНЫЕ АЛГОРИТМЫ

ozhsp = 0

for i in sp:

if i=='21':

ozhsp+=1

ozhspall += ozhsp

ozhsp = str((ozhsp\*10)//60)+"ч. "+str((ozhsp\*10)%60)+"м."

if samin >= 4:#если индекс самолета равен и ли больше 4, то мы начинаем прибавлять к новым переменным наши прошлые отдельные перменные для двух самолетов, чтобы потом в конце сложить всё вместе

dvsegoall += vsegoall

dvsisprall += vsisprall

dvpolall += vpolall

dobspall += obspall

dmeteoall += meteoall

drezrvall += rezrvall

disneisall += isneisall

dvsneisprall += vsneisprall

dustrnall += ustrnall

dozhoball += ozhoball

dobslall += obslall

dozhperall += ozhperall

dperall += perall

dnochall += nochall

dvremntall += vremntall

dozhrall += ozhrall

dvossprall += vossprall

dotszpall += otszpall

dotsdvall += otsdvall

ddorbulall += dorbulall

drekprall += rekprall

drekremzall += rekremzall

dozhspall += ozhspall

gg = 2+samin#переменная с номером столбца, чтобы потом заполнить правильный столбец автоматом

#короче, всё, что мы в прошлые алгоритмы наскладывали, мы переводим в часы и минуты

vsegoall = str((vsegoall\*10)//60)+"ч. "+str((vsegoall\*10)%60)+"м."

vsisprall = str((vsisprall\*10)//60)+"ч. "+str((vsisprall\*10)%60)+"м."

vpolall = str((vpolall\*10)//60)+"ч. "+str((vpolall\*10)%60)+"м."

obspall = str((obspall\*10)//60)+"ч. "+str((obspall\*10)%60)+"м."

meteoall = str((meteoall\*10)//60)+"ч. "+str((meteoall\*10)%60)+"м."

rezrvall = str((rezrvall\*10)//60)+"ч. "+str((rezrvall\*10)%60)+"м."

isneisall = str((isneisall\*10)//60)+"ч. "+str((isneisall\*10)%60)+"м."

vsneisprall = str((vsneisprall\*10)//60)+"ч. "+str((vsneisprall\*10)%60)+"м."

ustrnall = str((ustrnall\*10)//60)+"ч. "+str((ustrnall\*10)%60)+"м."

ozhoball = str((ozhoball\*10)//60)+"ч. "+str((ozhoball\*10)%60)+"м."

obslall = str((obslall\*10)//60)+"ч. "+str((obslall\*10)%60)+"м."

ozhperall = str((ozhperall\*10)//60)+"ч. "+str((ozhperall\*10)%60)+"м."

perall = str((perall\*10)//60)+"ч. "+str((perall\*10)%60)+"м."

nochall = str((nochall\*10)//60)+"ч. "+str((nochall\*10)%60)+"м."

vremntall = str((vremntall\*10)//60)+"ч. "+str((vremntall\*10)%60)+"м."

ozhrall = str((ozhrall\*10)//60)+"ч. "+str((ozhrall\*10)%60)+"м."

vossprall = str((vossprall\*10)//60)+"ч. "+str((vossprall\*10)%60)+"м."

otszpall = str((otszpall\*10)//60)+"ч. "+str((otszpall\*10)%60)+"м."

otsdvall = str((otsdvall\*10)//60)+"ч. "+str((otsdvall\*10)%60)+"м."

dorbulall = str((dorbulall\*10)//60)+"ч. "+str((dorbulall\*10)%60)+"м."

rekprall = str((rekprall\*10)//60)+"ч. "+str((rekprall\*10)%60)+"м."

rekremzall = str((rekremzall\*10)//60)+"ч. "+str((rekremzall\*10)%60)+"м."

ozhspall = str((ozhspall\*10)//60)+"ч. "+str((ozhspall\*10)%60)+"м."

#дальше мы эти переменные вставляем в ячейки таблицы, столбец заполняем

self.tableWidget3.setItem(3, gg, QTableWidgetItem(vsegoall))

self.tableWidget3.setItem(4, gg, QTableWidgetItem(vsisprall))

self.tableWidget3.setItem(5, gg, QTableWidgetItem(vpolall))

self.tableWidget3.setItem(6, gg, QTableWidgetItem(obspall))

self.tableWidget3.setItem(7, gg, QTableWidgetItem(meteoall))

self.tableWidget3.setItem(8, gg, QTableWidgetItem(rezrvall))

self.tableWidget3.setItem(9, gg, QTableWidgetItem(isneisall))

self.tableWidget3.setItem(10, gg, QTableWidgetItem(vsneisprall))

self.tableWidget3.setItem(11, gg, QTableWidgetItem(ustrnall))

self.tableWidget3.setItem(12, gg, QTableWidgetItem(obslall))

self.tableWidget3.setItem(13, gg, QTableWidgetItem(ozhperall))

self.tableWidget3.setItem(14, gg, QTableWidgetItem(perall))

self.tableWidget3.setItem(15, gg, QTableWidgetItem(nochall))

self.tableWidget3.setItem(16, gg, QTableWidgetItem(vremntall))

self.tableWidget3.setItem(17, gg, QTableWidgetItem(ozhrall))

self.tableWidget3.setItem(18, gg, QTableWidgetItem(otszpall))

self.tableWidget3.setItem(19, gg, QTableWidgetItem(otsdvall))

self.tableWidget3.setItem(20, gg, QTableWidgetItem(dorbulall))

self.tableWidget3.setItem(21, gg, QTableWidgetItem(rekprall))

self.tableWidget3.setItem(22, gg, QTableWidgetItem(rekremzall))

self.tableWidget3.setItem(23, gg, QTableWidgetItem(ozhspall))

if samin >= 4:#если индекс самолета равен или больше 4

#кхм, теперь к прошлой переменной, где мы запомнили данные одного самолета, мы прибавляем второй самолет и переводим в часы :D

ddvsegoall = str((dvsegoall\*10)//60)+"ч. "+str((dvsegoall\*10)%60)+"м."

ddvsisprall = str((dvsisprall\*10)//60)+"ч. "+str((dvsisprall\*10)%60)+"м."

ddvpolall = str((dvpolall\*10)//60)+"ч. "+str((dvpolall\*10)%60)+"м."

ddobspall = str((dobspall\*10)//60)+"ч. "+str((dobspall\*10)%60)+"м."

ddmeteoall = str((dmeteoall\*10)//60)+"ч. "+str((dmeteoall\*10)%60)+"м."

ddrezrvall = str((drezrvall\*10)//60)+"ч. "+str((drezrvall\*10)%60)+"м."

ddisneisall = str((disneisall\*10)//60)+"ч. "+str((disneisall\*10)%60)+"м."

ddvsneisprall = str((dvsneisprall\*10)//60)+"ч. "+str((dvsneisprall\*10)%60)+"м."

ddustrnall = str((dustrnall\*10)//60)+"ч. "+str((dustrnall\*10)%60)+"м."

ddozhoball = str((dozhoball\*10)//60)+"ч. "+str((dozhoball\*10)%60)+"м."

ddobslall = str((dobslall\*10)//60)+"ч. "+str((dobslall\*10)%60)+"м."

ddozhperall = str((dozhperall\*10)//60)+"ч. "+str((dozhperall\*10)%60)+"м."

ddperall = str((dperall\*10)//60)+"ч. "+str((dperall\*10)%60)+"м."

ddnochall = str((dnochall\*10)//60)+"ч. "+str((dnochall\*10)%60)+"м."

ddvremntall = str((dvremntall\*10)//60)+"ч. "+str((dvremntall\*10)%60)+"м."

ddozhrall = str((dozhrall\*10)//60)+"ч. "+str((dozhrall\*10)%60)+"м."

ddvossprall = str((dvossprall\*10)//60)+"ч. "+str((dvossprall\*10)%60)+"м."

ddotszpall = str((dotszpall\*10)//60)+"ч. "+str((dotszpall\*10)%60)+"м."

ddotsdvall = str((dotsdvall\*10)//60)+"ч. "+str((dotsdvall\*10)%60)+"м."

dddorbulall = str((ddorbulall\*10)//60)+"ч. "+str((ddorbulall\*10)%60)+"м."

ddrekprall = str((drekprall\*10)//60)+"ч. "+str((drekprall\*10)%60)+"м."

ddrekremzall = str((drekremzall\*10)//60)+"ч. "+str((drekremzall\*10)%60)+"м."

ddozhspall = str((dozhspall\*10)//60)+"ч. "+str((dozhspall\*10)%60)+"м."

#так же заполняем столбец

self.tableWidget3.setItem(3, 6, QTableWidgetItem(ddvsegoall))

self.tableWidget3.setItem(4, 6, QTableWidgetItem(ddvsisprall))

self.tableWidget3.setItem(5, 6, QTableWidgetItem(ddvpolall))

self.tableWidget3.setItem(6, 6, QTableWidgetItem(ddobspall))

self.tableWidget3.setItem(7, 6, QTableWidgetItem(ddmeteoall))

self.tableWidget3.setItem(8, 6, QTableWidgetItem(ddrezrvall))

self.tableWidget3.setItem(9, 6, QTableWidgetItem(ddisneisall))

self.tableWidget3.setItem(10, 6, QTableWidgetItem(ddvsneisprall))

self.tableWidget3.setItem(11, 6, QTableWidgetItem(ddustrnall))

self.tableWidget3.setItem(12, 6, QTableWidgetItem(ddobslall))

self.tableWidget3.setItem(13, 6, QTableWidgetItem(ddozhperall))

self.tableWidget3.setItem(14, 6, QTableWidgetItem(ddperall))

self.tableWidget3.setItem(15, 6, QTableWidgetItem(ddnochall))

self.tableWidget3.setItem(16, 6, QTableWidgetItem(ddvremntall))

self.tableWidget3.setItem(17, 6, QTableWidgetItem(ddozhrall))

self.tableWidget3.setItem(18, 6, QTableWidgetItem(ddotszpall))

self.tableWidget3.setItem(19, 6, QTableWidgetItem(ddotsdvall))

self.tableWidget3.setItem(20, 6, QTableWidgetItem(dddorbulall))

self.tableWidget3.setItem(21, 6, QTableWidgetItem(ddrekprall))

self.tableWidget3.setItem(22, 6, QTableWidgetItem(ddrekremzall))

self.tableWidget3.setItem(23, 6, QTableWidgetItem(ddozhspall))

except: #если чето пошло не так, то выскакивает ошибка

QMessageBox.about(self, "Ошибка", "Ошибка данных!")

class Window1(QWidget): #класс окна1, он аналогичен прошлому окну, отличий собо нет!!!!!!!!!

def \_\_init\_\_(self):

super(Window1, self).\_\_init\_\_()

self.setWindowTitle('Табель учета исправности и использования самолетов')

self.setMinimumWidth(800)

self.setMinimumHeight(600)

self.tableWidget2 = QtWidgets.QTableWidget(self)

self.tableWidget2.setGeometry(QtCore.QRect(10, 10, 501, 591))

self.tableWidget2.setRowCount(27)

self.tableWidget2.setColumnCount(34)

self.tableWidget2.horizontalHeader().hide()

self.tableWidget2.verticalHeader().hide()

self.tableWidget2.setItem(0, 0, QTableWidgetItem("№"))

self.tableWidget2.setItem(0, 1, QTableWidgetItem("Состояние эксплуатации"))

self.tableWidget2.setItem(0, 2, QTableWidgetItem("Усл. обозн."))

self.tableWidget2.setVerticalScrollBarPolicy(Qt.ScrollBarAlwaysOff)

self.tableWidget2.setItem(25, 0, QTableWidgetItem(" "))

self.tableWidget2.setItem(26, 0, QTableWidgetItem(" "))

self.tableWidget2.setItem(1, 1, QTableWidgetItem("Всего самолёто - часов"))

self.tableWidget2.setItem(2, 1, QTableWidgetItem("Всего исправных самолёто - часов"))

self.tableWidget2.setItem(3, 1, QTableWidgetItem("Время в рейсе"))

self.tableWidget2.setItem(4, 1, QTableWidgetItem("Обеспечение рейса"))

self.tableWidget2.setItem(5, 1, QTableWidgetItem("Простой по метео - условиям и запретам"))

self.tableWidget2.setItem(6, 1, QTableWidgetItem("В резерве"))

self.tableWidget2.setItem(7, 1, QTableWidgetItem("Исправные - неиспользованные"))

self.tableWidget2.setItem(8, 1, QTableWidgetItem("Всего неисправных самолёто - часов"))

self.tableWidget2.setItem(9, 1, QTableWidgetItem("Устранение неисправностей при Ф-А"))

self.tableWidget2.setItem(10, 1, QTableWidgetItem("Ожидание обслуживания по Ф-Б"))

self.tableWidget2.setItem(11, 1, QTableWidgetItem("Обслуживание по Ф-Б"))

self.tableWidget2.setItem(12, 1, QTableWidgetItem("Ожидание периодического Т.О."))

self.tableWidget2.setItem(13, 1, QTableWidgetItem("Периодическое Т.О."))

self.tableWidget2.setItem(14, 1, QTableWidgetItem("Ночные и межсменные простои Т.О."))

self.tableWidget2.setItem(15, 1, QTableWidgetItem("Ожидание ремонта"))

self.tableWidget2.setItem(16, 1, QTableWidgetItem("В ремонте"))

self.tableWidget2.setItem(17, 1, QTableWidgetItem("Отсутствие запасных частей"))

self.tableWidget2.setItem(18, 1, QTableWidgetItem("Отсутствие двигателей"))

self.tableWidget2.setItem(19, 1, QTableWidgetItem("Доработки по бюллетеням"))

self.tableWidget2.setItem(20, 1, QTableWidgetItem("Рекламация промышленности"))

self.tableWidget2.setItem(21, 1, QTableWidgetItem("Рекламация ремонтным заводом"))

self.tableWidget2.setItem(22, 1, QTableWidgetItem("Расследование проишествий"))

self.tableWidget2.setItem(23, 1, QTableWidgetItem("Восстановление после проишествий"))

self.tableWidget2.setItem(24, 1, QTableWidgetItem("Ожидание списания"))

self.tableWidget2.setItem(25, 1, QTableWidgetItem("Процент исправности"))

self.tableWidget2.setItem(26, 1, QTableWidgetItem("Процент использования"))

self.tableWidget2.setItem(1, 2, QTableWidgetItem("Ф"))

self.tableWidget2.setItem(2, 2, QTableWidgetItem("И"))

self.tableWidget2.setItem(3, 2, QTableWidgetItem("К"))

self.tableWidget2.setItem(4, 2, QTableWidgetItem("Е"))

self.tableWidget2.setItem(5, 2, QTableWidgetItem("М"))

self.tableWidget2.setItem(6, 2, QTableWidgetItem("Г"))

self.tableWidget2.setItem(7, 2, QTableWidgetItem("А"))

self.tableWidget2.setItem(8, 2, QTableWidgetItem("Н"))

self.tableWidget2.setItem(9, 2, QTableWidgetItem("У"))

self.tableWidget2.setItem(10, 2, QTableWidgetItem("Об"))

self.tableWidget2.setItem(11, 2, QTableWidgetItem("Тб"))

self.tableWidget2.setItem(12, 2, QTableWidgetItem("Оп"))

self.tableWidget2.setItem(13, 2, QTableWidgetItem("Тп"))

self.tableWidget2.setItem(14, 2, QTableWidgetItem("Ш"))

self.tableWidget2.setItem(15, 2, QTableWidgetItem("Ор"))

self.tableWidget2.setItem(16, 2, QTableWidgetItem("Р"))

self.tableWidget2.setItem(17, 2, QTableWidgetItem("З"))

self.tableWidget2.setItem(18, 2, QTableWidgetItem("Дв"))

self.tableWidget2.setItem(19, 2, QTableWidgetItem("Д"))

self.tableWidget2.setItem(20, 2, QTableWidgetItem("Ж"))

self.tableWidget2.setItem(21, 2, QTableWidgetItem("Жр"))

self.tableWidget2.setItem(22, 2, QTableWidgetItem("Л"))

self.tableWidget2.setItem(23, 2, QTableWidgetItem("В"))

self.tableWidget2.setItem(24, 2, QTableWidgetItem("С"))

self.tableWidget2.setItem(25, 2, QTableWidgetItem("Киспр"))

self.tableWidget2.setItem(26, 2, QTableWidgetItem("Кисп"))

self.tableWidget2.setEditTriggers(QtWidgets.QTableWidget.NoEditTriggers)

nn = 1

for j in range(34):

if j >= 1 and j<25:

self.tableWidget2.setItem(j, 0, QTableWidgetItem(str(j)))

if j<27:

self.tableWidget2.item(j, 0).setBackground(QtGui.QColor(219,235,255))

self.tableWidget2.item(j, 1).setBackground(QtGui.QColor(219,235,255))

self.tableWidget2.item(j, 2).setBackground(QtGui.QColor(219,235,255))

if j >=3 and j<34:

self.tableWidget2.setItem(0, j, QTableWidgetItem(str(nn)))

nn+=1

self.tableWidget2.item(0, j).setBackground(QtGui.QColor(219,235,255))

self.tableWidget2.resizeColumnsToContents()

self.groupBox2 = QtWidgets.QGroupBox(self)

self.groupBox2.setGeometry(QtCore.QRect(520, 1, 271, 221))

font = QtGui.QFont()

font.setFamily("Segoe Print")

font.setPointSize(11)

font.setBold(True)

font.setUnderline(False)

font.setWeight(75)

self.groupBox2.setFont(font)

self.groupBox2.setTitle("Настройки")

self.combo = QComboBox(self.groupBox2)

self.combo.addItem("SSJ-100")

self.combo.addItem("Airbus A310")

self.combo.addItem("ИЛ-96М")

self.combo.addItem("Airbus A320")

self.combo.addItem("Boeing-737")

self.qlabel = QLabel(self.groupBox2)

self.qlabel.move(10,37)

self.qlabel.setText('Самолет:')

self.qlabel2 = QLabel(self.groupBox2)

self.qlabel2.move(10,77)

self.qlabel2.setText('Месяц:')

self.combo2 = QComboBox(self.groupBox2)

self.combo2.addItem("Январь")

self.combo2.addItem("Февраль")

self.combo2.addItem("Март")

self.combo2.addItem("Апрель")

self.combo2.addItem("Май")

self.combo2.addItem("Июнь")

self.combo2.addItem("Июль")

self.combo2.addItem("Август")

self.combo2.addItem("Сентябрь")

self.combo2.addItem("Октябрь")

self.combo2.addItem("Ноябрь")

self.combo2.addItem("Декабрь")

self.combo.move(110, 35)

self.combo2.move(80, 77)

self.qlabel3 = QLabel(self.groupBox2)

self.qlabel3.move(10,125)

self.qlabel3.setText('Год:')

self.line = QLineEdit(self.groupBox2)

self.line.move(65, 127)

self.pushButton = QtWidgets.QPushButton(self.groupBox2)

self.pushButton.setGeometry(QtCore.QRect(10, 180, 231, 31))

self.pushButton.clicked.connect(self.tab)

self.pushButton.setText('ПРОСМОТР')

def tab(self):#ункция заполнения таблицы в табеле

year=self.line.text()

mmm = str(self.combo2.currentIndex()+1)

samin = self.combo.currentIndex()+1

chty = (self.combo2.currentIndex()+1)%2 #тут я четность месяца проверяю, четный - 31 день, нечетный - 30

if chty == 0:#если номер месяца делится без отстатка

msx = 31#то дней 31

else:#если нет

msx = 30#то 30

if mmm == 2:#если месяц имеет номер 2 (февраль)

msx = 28#то 28 дней

try:

for i in range(msx):

xhasi = 0 #это переменная нужна для формулы, чтобы высчитать проценты, я не помню что она означает :(

i = i+1

ii = i

if i<10:

dddd = "0"+str(i)

with open("base/"+dddd+"-0"+mmm+"-"+year, 'r') as f:

text = f.read().split("\n")

vsego = 0

sp = text[samin].split(' ')

for i in sp:

if i!='0' and i!='':

vsego+=1

vsego = str((vsego\*10)//60)+"ч. "+str((vsego\*10)%60)+"м."

vsispr = 0

for i in sp:

if i=='1' or i=='2'or i=='3'or i=='4'or i=='5':

vsispr+=1

vsispr = str((vsispr\*10)//60)+"ч. "+str((vsispr\*10)%60)+"м."

vpol = 0

for i in sp:

if i=='1':

vpol+=1

fgpol = vpol

vpol = str((vpol\*10)//60)+"ч. "+str((vpol\*10)%60)+"м."

obsp = 0

for i in sp:

if i=='2':

obsp+=1

xhasi =+ obsp

obsp = str((obsp\*10)//60)+"ч. "+str((obsp\*10)%60)+"м."

meteo = 0

for i in sp:

if i=='3':

meteo+=1

xhasi =+ meteo

meteo = str((meteo\*10)//60)+"ч. "+str((meteo\*10)%60)+"м."

rezrv = 0

for i in sp:

if i=='4':

rezrv+=1

xhasi =+ rezrv

rezrv = str((rezrv\*10)//60)+"ч. "+str((rezrv\*10)%60)+"м."

isneis = 0

for i in sp:

if i=='5':

isneis+=1

xhasi =+ isneis

isneis = str((isneis\*10)//60)+"ч. "+str((isneis\*10)%60)+"м."

vsneispr = 0

for i in sp:

if i!='1' and i!='2'and i!='3'and i!='4'and i!='5'and i!='0'and i!='':

vsneispr+=1

vsneispr = str((vsneispr\*10)//60)+"ч. "+str((vsneispr\*10)%60)+"м."

ustrn = 0

for i in sp:

if i=='6':

ustrn+=1

ustrn = str((ustrn\*10)//60)+"ч. "+str((ustrn\*10)%60)+"м."

ozhob = 0

for i in sp:

if i=='7':

ozhob+=1

ozhob= str((ozhob\*10)//60)+"ч. "+str((ozhob\*10)%60)+"м."

obsl = 0

for i in sp:

if i=='8':

obsl+=1

obsl = str((obsl\*10)//60)+"ч. "+str((obsl\*10)%60)+"м."

ozhper = 0

for i in sp:

if i=='9':

ozhper+=1

kozh = ozhper

ozhper = str((ozhper\*10)//60)+"ч. "+str((ozhper\*10)%60)+"м."

per = 0

for i in sp:

if i=='10':

per+=1

per = str((per\*10)//60)+"ч. "+str((per\*10)%60)+"м."

noch = 0

for i in sp:

if i=='11':

noch+=1

noch = str((noch\*10)//60)+"ч. "+str((noch\*10)%60)+"м."

ozhr = 0

for i in sp:

if i=='12':

ozhr+=1

ozhr = str((ozhr\*10)//60)+"ч. "+str((ozhr\*10)%60)+"м."

vremnt = 0

for i in sp:

if i=='13':

vremnt+=1

vremnt = str((vremnt\*10)//60)+"ч. "+str((vremnt\*10)%60)+"м."

otszp = 0

for i in sp:

if i=='14':

otszp+=1

otszp = str((otszp\*10)//60)+"ч. "+str((otszp\*10)%60)+"м."

otsdv = 0

for i in sp:

if i=='15':

otsdv+=1

otsdv = str((otsdv\*10)//60)+"ч. "+str((otsdv\*10)%60)+"м."

dorbul = 0

for i in sp:

if i=='16':

dorbul+=1

dorbul = str((dorbul\*10)//60)+"ч. "+str((dorbul\*10)%60)+"м."

rekpr = 0

for i in sp:

if i=='17':

rekpr+=1

rekpr = str((rekpr\*10)//60)+"ч. "+str((rekpr\*10)%60)+"м."

rekremz = 0

for i in sp:

if i=='18':

rekremz+=1

rekremz = str((rekremz\*10)//60)+"ч. "+str((rekremz\*10)%60)+"м."

rasspr = 0

for i in sp:

if i=='19':

rasspr+=1

rasspr = str((rasspr\*10)//60)+"ч. "+str((rasspr\*10)%60)+"м."

vosspr = 0

for i in sp:

if i=='20':

vosspr+=1

vosspr = str((vosspr\*10)//60)+"ч. "+str((vosspr\*10)%60)+"м."

ozhsp = 0

for i in sp:

if i=='21':

ozhsp+=1

ozhsp = str((ozhsp\*10)//60)+"ч. "+str((ozhsp\*10)%60)+"м."

xhasi = xhasi\*10 #переводим непонятную переменную для формулы в минуты

fgpol = fgpol\*10 #переводим непонятную переменную для формулы в минуты

kpi = xhasi/fgpol #чето делим, так надо

kispr = (720-(kozh\*10))/720 #высчитываем коэффицент исправности

kisp = (720\*kispr-fgpol\*kpi)/720#высчитываем коэффицент используемости

#вставляем всё добро по ячейкам

self.tableWidget2.setItem(1, 2+ii, QTableWidgetItem(vsego))

self.tableWidget2.setItem(2, 2+ii, QTableWidgetItem(vsispr))

self.tableWidget2.setItem(3, 2+ii, QTableWidgetItem(vpol))

self.tableWidget2.setItem(4, 2+ii, QTableWidgetItem(obsp))

self.tableWidget2.setItem(5, 2+ii, QTableWidgetItem(meteo))

self.tableWidget2.setItem(6, 2+ii, QTableWidgetItem(rezrv))

self.tableWidget2.setItem(7, 2+ii, QTableWidgetItem(isneis))

self.tableWidget2.setItem(8, 2+ii, QTableWidgetItem(vsneispr))

self.tableWidget2.setItem(9, 2+ii, QTableWidgetItem(ustrn))

self.tableWidget2.setItem(10, 2+ii, QTableWidgetItem(ozhob))

self.tableWidget2.setItem(11, 2+ii, QTableWidgetItem(obsl))

self.tableWidget2.setItem(12, 2+ii, QTableWidgetItem(ozhper))

self.tableWidget2.setItem(13, 2+ii, QTableWidgetItem(per))

self.tableWidget2.setItem(14, 2+ii, QTableWidgetItem(noch))

self.tableWidget2.setItem(15, 2+ii, QTableWidgetItem(ozhr))

self.tableWidget2.setItem(16, 2+ii, QTableWidgetItem(vremnt))

self.tableWidget2.setItem(17, 2+ii, QTableWidgetItem(otszp))

self.tableWidget2.setItem(18, 2+ii, QTableWidgetItem(otsdv))

self.tableWidget2.setItem(19, 2+ii, QTableWidgetItem(dorbul))

self.tableWidget2.setItem(20, 2+ii, QTableWidgetItem(rekpr))

self.tableWidget2.setItem(21, 2+ii, QTableWidgetItem(rekremz))

self.tableWidget2.setItem(22, 2+ii, QTableWidgetItem(rasspr))

self.tableWidget2.setItem(23, 2+ii, QTableWidgetItem(vosspr))

self.tableWidget2.setItem(24, 2+ii, QTableWidgetItem(ozhsp))

self.tableWidget2.setItem(25, 2+ii, QTableWidgetItem(str(round(kispr,2))[2:]+'%'))

self.tableWidget2.setItem(26, 2+ii, QTableWidgetItem(str(round(kisp,2))[2:]+'%'))

self.tableWidget2.resizeColumnsToContents()

except:

QMessageBox.about(self, "Ошибка", "Ошибка данных!")

class Ui\_MainWindow(object):#класс основного нашего окна

def show\_window\_1(self):#функция вызова первого окна (табеля)

self.w1 = Window1()

self.w1.show()

def show\_window\_2(self):#функция вызова второго окна (отчета)

self.w2 = Window2()

self.w2.show()

###ФУНКЦИИ КОНТЕКСТНОГО МЕНЮ

#номер функции соответствует порядку в контекстном меню, опишу один, а остальные такие же по аналогии

def daction1(self):

for index in self.tableWidget.selectedIndexes():#определяем индекс выбранной ячейки (ее строку и столбец)

pic = QtGui.QPixmap("data/1.bmp")#в переменную загоняем нашу картинку, с номером, соответвующим порядку в конт. меню

self.label = QtWidgets.QLabel(MainWindow)#создаем пустой лейбл

self.label.setPixmap(pic)#вставляем в лейбл нашу картинку

self.tableWidget.setCellWidget(index.row(),index.column(), self.label)#вставляем в ячейку лейбл, в котором наша картинка

self.tableWidget.setItem(index.row(),index.column(), QTableWidgetItem("1"))#под картинку вставляем её обозначающий номер (как в названии)

#остальные по такому же принципу, только номер картинки отличается

def daction2(self):

for index in self.tableWidget.selectedIndexes():

pic = QtGui.QPixmap("data/2.bmp")

self.label = QtWidgets.QLabel(MainWindow)

self.label.setPixmap(pic)

self.tableWidget.setCellWidget(index.row(),index.column(), self.label)

self.tableWidget.setItem(index.row(),index.column(), QTableWidgetItem("2"))

def daction3(self):

for index in self.tableWidget.selectedIndexes():

pic = QtGui.QPixmap("data/3.bmp")

self.label = QtWidgets.QLabel(MainWindow)

self.label.setPixmap(pic)

self.tableWidget.setCellWidget(index.row(),index.column(), self.label)

self.tableWidget.setItem(index.row(),index.column(), QTableWidgetItem("3"))

def daction4(self):

for index in self.tableWidget.selectedIndexes():

pic = QtGui.QPixmap("data/4.bmp")

self.label = QtWidgets.QLabel(MainWindow)

self.label.setPixmap(pic)

self.tableWidget.setCellWidget(index.row(),index.column(), self.label)

self.tableWidget.setItem(index.row(),index.column(), QTableWidgetItem("4"))

def daction5(self):

for index in self.tableWidget.selectedIndexes():

pic = QtGui.QPixmap("data/5.bmp")

self.label = QtWidgets.QLabel(MainWindow)

self.label.setPixmap(pic)

self.tableWidget.setCellWidget(index.row(),index.column(), self.label)

self.tableWidget.setItem(index.row(),index.column(), QTableWidgetItem("5"))

def daction6(self):

for index in self.tableWidget.selectedIndexes():

pic = QtGui.QPixmap("data/6.bmp")

self.label = QtWidgets.QLabel(MainWindow)

self.label.setPixmap(pic)

self.tableWidget.setCellWidget(index.row(),index.column(), self.label)

self.tableWidget.setItem(index.row(),index.column(), QTableWidgetItem("6"))

def daction7(self):

for index in self.tableWidget.selectedIndexes():

pic = QtGui.QPixmap("data/7.bmp")

self.label = QtWidgets.QLabel(MainWindow)

self.label.setPixmap(pic)

self.tableWidget.setCellWidget(index.row(),index.column(), self.label)

self.tableWidget.setItem(index.row(),index.column(), QTableWidgetItem("7"))

def daction8(self):

for index in self.tableWidget.selectedIndexes():

pic = QtGui.QPixmap("data/8.bmp")

self.label = QtWidgets.QLabel(MainWindow)

self.label.setPixmap(pic)

self.tableWidget.setCellWidget(index.row(),index.column(), self.label)

self.tableWidget.setItem(index.row(),index.column(), QTableWidgetItem("8"))

def daction9(self):

for index in self.tableWidget.selectedIndexes():

pic = QtGui.QPixmap("data/9.bmp")

self.label = QtWidgets.QLabel(MainWindow)

self.label.setPixmap(pic)

self.tableWidget.setCellWidget(index.row(),index.column(), self.label)

self.tableWidget.setItem(index.row(),index.column(), QTableWidgetItem("9"))

def daction10(self):

for index in self.tableWidget.selectedIndexes():

pic = QtGui.QPixmap("data/10.bmp")

self.label = QtWidgets.QLabel(MainWindow)

self.label.setPixmap(pic)

self.tableWidget.setCellWidget(index.row(),index.column(), self.label)

self.tableWidget.setItem(index.row(),index.column(), QTableWidgetItem("10"))

def daction11(self):

for index in self.tableWidget.selectedIndexes():

pic = QtGui.QPixmap("data/11.bmp")

self.label = QtWidgets.QLabel(MainWindow)

self.label.setPixmap(pic)

self.tableWidget.setCellWidget(index.row(),index.column(), self.label)

self.tableWidget.setItem(index.row(),index.column(), QTableWidgetItem("11"))

def daction12(self):

for index in self.tableWidget.selectedIndexes():

pic = QtGui.QPixmap("data/12.bmp")

self.label = QtWidgets.QLabel(MainWindow)

self.label.setPixmap(pic)

self.tableWidget.setCellWidget(index.row(),index.column(), self.label)

self.tableWidget.setItem(index.row(),index.column(), QTableWidgetItem("12"))

def daction13(self):

for index in self.tableWidget.selectedIndexes():

pic = QtGui.QPixmap("data/13.bmp")

self.label = QtWidgets.QLabel(MainWindow)

self.label.setPixmap(pic)

self.tableWidget.setCellWidget(index.row(),index.column(), self.label)

self.tableWidget.setItem(index.row(),index.column(), QTableWidgetItem("13"))

def daction14(self):

for index in self.tableWidget.selectedIndexes():

pic = QtGui.QPixmap("data/14.bmp")

self.label = QtWidgets.QLabel(MainWindow)

self.label.setPixmap(pic)

self.tableWidget.setCellWidget(index.row(),index.column(), self.label)

self.tableWidget.setItem(index.row(),index.column(), QTableWidgetItem("14"))

def daction15(self):

for index in self.tableWidget.selectedIndexes():

pic = QtGui.QPixmap("data/15.bmp")

self.label = QtWidgets.QLabel(MainWindow)

self.label.setPixmap(pic)

self.tableWidget.setCellWidget(index.row(),index.column(), self.label)

self.tableWidget.setItem(index.row(),index.column(), QTableWidgetItem("15"))

def daction16(self):

for index in self.tableWidget.selectedIndexes():

pic = QtGui.QPixmap("data/16.bmp")

self.label = QtWidgets.QLabel(MainWindow)

self.label.setPixmap(pic)

self.tableWidget.setCellWidget(index.row(),index.column(), self.label)

self.tableWidget.setItem(index.row(),index.column(), QTableWidgetItem("16"))

def daction17(self):

for index in self.tableWidget.selectedIndexes():

pic = QtGui.QPixmap("data/17.bmp")

self.label = QtWidgets.QLabel(MainWindow)

self.label.setPixmap(pic)

self.tableWidget.setCellWidget(index.row(),index.column(), self.label)

self.tableWidget.setItem(index.row(),index.column(), QTableWidgetItem("17"))

def daction18(self):

for index in self.tableWidget.selectedIndexes():

pic = QtGui.QPixmap("data/18.bmp")

self.label = QtWidgets.QLabel(MainWindow)

self.label.setPixmap(pic)

self.tableWidget.setCellWidget(index.row(),index.column(), self.label)

self.tableWidget.setItem(index.row(),index.column(), QTableWidgetItem("18"))

def daction19(self):

for index in self.tableWidget.selectedIndexes():

pic = QtGui.QPixmap("data/19.bmp")

self.label = QtWidgets.QLabel(MainWindow)

self.label.setPixmap(pic)

self.tableWidget.setCellWidget(index.row(),index.column(), self.label)

self.tableWidget.setItem(index.row(),index.column(), QTableWidgetItem("19"))

def daction20(self):

for index in self.tableWidget.selectedIndexes():

pic = QtGui.QPixmap("data/20.bmp")

self.label = QtWidgets.QLabel(MainWindow)

self.label.setPixmap(pic)

self.tableWidget.setCellWidget(index.row(),index.column(), self.label)

self.tableWidget.setItem(index.row(),index.column(), QTableWidgetItem("20"))

def daction21(self):

for index in self.tableWidget.selectedIndexes():

pic = QtGui.QPixmap("data/21.bmp")

self.label = QtWidgets.QLabel(MainWindow)

self.label.setPixmap(pic)

self.tableWidget.setCellWidget(index.row(),index.column(), self.label)

self.tableWidget.setItem(index.row(),index.column(), QTableWidgetItem("21"))

#функция сохранения графика

def savegr(self):

name = self.calendarWidget.selectedDate().toString("dd-MM-yyyy")#создаем отдельную переменную для выбраной даты в календаре

with open("base/"+name, 'w', encoding='UTF-8') as out\_file:#создаем файл с названием, как у выбранной ранее датой

for row in range(self.tableWidget.rowCount()):#цикл перебора строк

for column in range(self.tableWidget.columnCount()):#цикл перебора столбцов

if row == 0 or column == 0 or column ==1:#если строка равняется 0, а колонка 0 или 1, то мы их не сохраняем, так как там находятся просто названия

p = 0#заглушка

else:

item = self.tableWidget.item(row, column)#элемент таблицы (там может быть цифра от 0 до 21, вспоминай конт. меню)с определенной строкой и столбцем, мы загоняем в отдельную переменную

print(item.text() if item else "0", end=' ', file=out\_file)#записываем в файл это элемент

print('', file=out\_file)#записываем в конце просто пробел как отступ

#функция загрузки графика

def gruz(self):

name = self.calendarWidget.selectedDate().toString("dd-MM-yyyy")#создаем отдельную переменную для выбраной даты в календаре

num = 0 #переменная для счетчика индекса, чтобы брать числа из файла по индексу

with open("base/"+name, 'r') as f:#открываем файл с названием, как у выбранной ранее датой

text = f.read().replace("\n", " ").split()#считываем данные из файла, убираем все отступы и пробелы, и делим на отдельные элементы (кажда цифра в строке, будет отдельно от других)

for row in range(self.tableWidget.rowCount()):#цикл перебора строк

for column in range(self.tableWidget.columnCount()):#цикл перебора столбцов

if row == 0 or column == 0 or column ==1:#если строка равняется 0, а колонка 0 или 1, то мы их не перезаписываем, так как там находятся просто названия

p = 0#заглушка

else:

pic = QtGui.QPixmap("data/"+text[num]+".bmp")#в переменную загоняем нашу картинку, с соответсвующим номером элемента из списка

self.label = QtWidgets.QLabel(MainWindow)#делаем пустой лейбл

self.label.setPixmap(pic)#вставляем туда картинку

self.tableWidget.setCellWidget(row,column, self.label)#лейбл в ячейку таблицы

self.tableWidget.setItem(row,column, QTableWidgetItem(text[num]))#под лейбл соответсвующий номер

num+=1#прибавляем +1 к счетчику

def och(self):#функция очистки, тут в принципе по анлогии, только во все ячейки вставляю просто картинку с белым фоном

for row in range(self.tableWidget.rowCount()):

for column in range(self.tableWidget.columnCount()):

if row == 0 or column == 0 or column ==1:

p = 0

else:

pic = QtGui.QPixmap("data/0.bmp")

self.label = QtWidgets.QLabel(MainWindow)

self.label.setPixmap(pic)

self.tableWidget.setCellWidget(row,column, self.label)

self.tableWidget.setItem(row,column, QTableWidgetItem("0"))

def setupUi(self, MainWindow):#интерфейс основного окна, ну можно по прошлым комментам понять что к чему

MainWindow.setObjectName("MainWindow")

MainWindow.resize(520, 535)

font = QtGui.QFont()

MainWindow.setFont(font)

MainWindow.setCursor(QtGui.QCursor(QtCore.Qt.CrossCursor))#меняем курсор на прицел

MainWindow.setFocusPolicy(QtCore.Qt.NoFocus)

icon = QtGui.QIcon()#переменная иконки

icon.addPixmap(QtGui.QPixmap("ИОТЭАТ МОЛОКО и СЕРКАН/sam.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)#меняем иконку на наш самолетик

MainWindow.setWindowIcon(icon)#вставляем иконку

MainWindow.setLayoutDirection(QtCore.Qt.LeftToRight)

self.centralwidget = QtWidgets.QWidget(MainWindow)

self.centralwidget.setObjectName("centralwidget")

#лейбл с фоном задним

self.label\_4 = QtWidgets.QLabel(self.centralwidget)

self.label\_4.setGeometry(QtCore.QRect(0, 0, 520, 519))

self.label\_4.setText("")

self.label\_4.setPixmap(QtGui.QPixmap("data/fno.jpg"))

self.label\_4.setScaledContents(True)

#мега супер пупер таблица

self.tableWidget = QtWidgets.QTableWidget(self.centralwidget)

self.tableWidget.setEditTriggers(QtWidgets.QTableWidget.NoEditTriggers)

self.tableWidget.setGeometry(QtCore.QRect(10, 10, 501, 191))

self.tableWidget.setRowCount(6)

self.tableWidget.setColumnCount(146)

self.tableWidget.horizontalHeader().hide()

self.tableWidget.verticalHeader().hide()

self.tableWidget.setObjectName("tableWidget")

self.tableWidget.setVerticalScrollBarPolicy(Qt.ScrollBarAlwaysOff)

self.tableWidget.setItem(0, 0, QTableWidgetItem("ТИП САМОЛЕТА"))

self.tableWidget.setItem(0, 1, QTableWidgetItem("БОРТОВОЙ №"))

self.tableWidget.setItem(1, 0, QTableWidgetItem("SSJ-100"))

self.tableWidget.setItem(1, 1, QTableWidgetItem("43214"))

self.tableWidget.setItem(2, 0, QTableWidgetItem("Airbus A310"))

self.tableWidget.setItem(2, 1, QTableWidgetItem("43206"))

self.tableWidget.setItem(3, 0, QTableWidgetItem("ИЛ-96М"))

self.tableWidget.setItem(3, 1, QTableWidgetItem("22814"))

self.tableWidget.setItem(4, 0, QTableWidgetItem("Airbus A320"))

self.tableWidget.setItem(4, 1, QTableWidgetItem("14888"))

self.tableWidget.setItem(5, 0, QTableWidgetItem("Airbus A320"))

self.tableWidget.setItem(5, 1, QTableWidgetItem("43612"))

self.tableWidget.setContextMenuPolicy(Qt.ActionsContextMenu)#ставим свойство таблицы, чтобы при нажатии на правую кнопку, выскакивало контексное меню

#вставляем в меню пункты по очереди

self.action1 = QAction("В рейсе(К)", MainWindow)

self.action2 = QAction("Обеспечение рейса(Е)", MainWindow)

self.action3 = QAction("Задержки по метео условиям и запретам(М)", MainWindow)

self.action4 = QAction("В резерве(Г)", MainWindow)

self.action5 = QAction("Исправный - неиспользуемый(А)", MainWindow)

self.action6 = QAction("Устранение неиспр. при оперативном ТО(У)", MainWindow)

self.action7 = QAction("Ожидание ТО по Ф-Б(Об)", MainWindow)

self.action8 = QAction("ТО по Ф-Б(Тб)", MainWindow)

self.action9 = QAction("Ожидание переодического ТО(Оп)", MainWindow)

self.action10 = QAction("ТО по переодическим формам(Тп)", MainWindow)

self.action11 = QAction("Межсменные или ночные перерывы(Ш)", MainWindow)

self.action12 = QAction("Ожидание ремонта(Ор)", MainWindow)

self.action13 = QAction("В ремонте(Р)", MainWindow)

self.action14 = QAction("Отсутствие запчастей(З)", MainWindow)

self.action15 = QAction("Отсутствие двигателей(Дв)", MainWindow)

self.action16 = QAction("Доработки по бюллютеням(Д)", MainWindow)

self.action17 = QAction("Рекламация промышленности(Ж)", MainWindow)

self.action18 = QAction("Рекламация ремзаводом(Жр)", MainWindow)

self.action19 = QAction("Расследование летных происшествий(Л)", MainWindow)

self.action20 = QAction("Восстановление самолета(В)", MainWindow)

self.action21 = QAction("Ожидание списания(С)", MainWindow)

#каждый пункт подключаем к контекстному меню

self.tableWidget.addAction(self.action1)

self.tableWidget.addAction(self.action2)

self.tableWidget.addAction(self.action3)

self.tableWidget.addAction(self.action4)

self.tableWidget.addAction(self.action5)

self.tableWidget.addAction(self.action6)

self.tableWidget.addAction(self.action7)

self.tableWidget.addAction(self.action8)

self.tableWidget.addAction(self.action9)

self.tableWidget.addAction(self.action10)

self.tableWidget.addAction(self.action11)

self.tableWidget.addAction(self.action12)

self.tableWidget.addAction(self.action13)

self.tableWidget.addAction(self.action14)

self.tableWidget.addAction(self.action15)

self.tableWidget.addAction(self.action16)

self.tableWidget.addAction(self.action17)

self.tableWidget.addAction(self.action18)

self.tableWidget.addAction(self.action19)

self.tableWidget.addAction(self.action20)

self.tableWidget.addAction(self.action21)

#каждый пункт подключаем к собсвенным функциям, которые были наверху

self.action1.triggered.connect(self.daction1)

self.action2.triggered.connect(self.daction2)

self.action3.triggered.connect(self.daction3)

self.action4.triggered.connect(self.daction4)

self.action5.triggered.connect(self.daction5)

self.action6.triggered.connect(self.daction6)

self.action7.triggered.connect(self.daction7)

self.action8.triggered.connect(self.daction8)

self.action9.triggered.connect(self.daction9)

self.action10.triggered.connect(self.daction10)

self.action11.triggered.connect(self.daction11)

self.action12.triggered.connect(self.daction12)

self.action13.triggered.connect(self.daction13)

self.action14.triggered.connect(self.daction14)

self.action15.triggered.connect(self.daction15)

self.action16.triggered.connect(self.daction16)

self.action17.triggered.connect(self.daction17)

self.action18.triggered.connect(self.daction18)

self.action19.triggered.connect(self.daction19)

self.action20.triggered.connect(self.daction20)

self.action21.triggered.connect(self.daction21)

#часть кода отвечающее за заполнение заголовка сверху временем (0:10, 0:20 и тд)

hour = 0#переменная часов по умолчанию

mint = 10#переменная минут по умолчанию

for j in range(146):#цикл

if j>=2:#если j больше либо равно двум, то работаем, просто первые два столбца (0 и 1) заняты уже под заголовки

if mint == 60:#если переменная минуты равна 60

mint = 0#минуты обнуляем

hour+=1#добавляем единицу к часам

self.tableWidget.setItem(0, j, QTableWidgetItem(" "+str(hour)+":"+str(mint)+"0 "))#заполняем ячейку

mint += 10#добавляем 10 к минутам

else:

self.tableWidget.setItem(0, j, QTableWidgetItem(" "+str(hour)+":"+str(mint)+" "))#заполняем ячейку

mint += 10#добавляем 10 к минутам

self.tableWidget.item(0, j).setBackground(QtGui.QColor(219,235,255))#закрашиваем всё это время

for j in range(6):#цикл, чтобы закрасить заголовки вначале

self.tableWidget.item(j, 0).setBackground(QtGui.QColor(219,235,255))

self.tableWidget.item(j, 1).setBackground(QtGui.QColor(219,235,255))

self.tableWidget.resizeColumnsToContents()#подгоняем ширину колонок к содердимому внутри

#пошел просто интерфейс, всё то же самое, что в окнах выше, можно догнать

self.groupBox = QtWidgets.QGroupBox(self.centralwidget)

self.groupBox.setGeometry(QtCore.QRect(10, 210, 251, 141))

font = QtGui.QFont()

font.setFamily("Segoe Print")

font.setPointSize(14)

font.setBold(True)

font.setWeight(75)

self.groupBox.setFont(font)

self.groupBox.setObjectName("groupBox")

self.pushButton = QtWidgets.QPushButton(self.groupBox)

self.pushButton.setGeometry(QtCore.QRect(10, 30, 231, 31))

font = QtGui.QFont()

font.setFamily("Segoe Print")

font.setPointSize(12)

self.pushButton.setFont(font)

self.pushButton.setLayoutDirection(QtCore.Qt.RightToLeft)

self.pushButton.setDefault(True)#чтобы кнопка была активной и сияла синим

self.pushButton.setObjectName("pushButton")

self.pushButton\_2 = QtWidgets.QPushButton(self.groupBox)

self.pushButton\_2.setGeometry(QtCore.QRect(10, 65, 231, 31))

font = QtGui.QFont()

font.setFamily("Segoe Print")

font.setPointSize(12)

self.pushButton\_2.setFont(font)

self.pushButton\_2.setDefault(True)

self.pushButton\_2.setObjectName("pushButton\_2")

self.pushButton\_3 = QtWidgets.QPushButton(self.groupBox)

self.pushButton\_3.setGeometry(QtCore.QRect(10, 100, 231, 31))

font = QtGui.QFont()

font.setFamily("Segoe Print")

font.setPointSize(12)

self.pushButton\_3.setFont(font)

self.pushButton\_3.setDefault(True)

self.pushButton\_3.setObjectName("pushButton\_3")

self.label = QtWidgets.QLabel(self.groupBox)

self.label.setGeometry(QtCore.QRect(23, 32, 28, 28))

self.label.setText("")

self.label.setPixmap(QtGui.QPixmap("data/babl.png"))

self.label.setScaledContents(True)

self.label.setObjectName("label")

self.label\_2 = QtWidgets.QLabel(self.groupBox)

self.label\_2.setGeometry(QtCore.QRect(24, 64, 28, 28))

self.label\_2.setText("")

self.label\_2.setPixmap(QtGui.QPixmap("data/kovid.png"))

self.label\_2.setScaledContents(True)

self.label\_2.setObjectName("label\_2")

self.label\_3 = QtWidgets.QLabel(self.groupBox)

self.label\_3.setGeometry(QtCore.QRect(24, 102, 28, 28))

self.label\_3.setText("")

self.label\_3.setPixmap(QtGui.QPixmap("data/jopa.png"))

self.label\_3.setScaledContents(True)

self.label\_3.setObjectName("label\_3")

self.groupBox\_2 = QtWidgets.QGroupBox(self.centralwidget)

self.groupBox\_2.setGeometry(QtCore.QRect(10, 350, 191, 141))

font = QtGui.QFont()

font.setFamily("Segoe Print")

font.setPointSize(14)

font.setBold(True)

font.setWeight(75)

self.groupBox\_2.setFont(font)

self.groupBox\_2.setObjectName("groupBox\_2")

self.pushButton\_4 = QtWidgets.QPushButton(self.groupBox\_2)

self.pushButton\_4.setGeometry(QtCore.QRect(10, 30, 171, 31))

font = QtGui.QFont()

font.setFamily("Segoe Print")

font.setPointSize(12)

self.pushButton\_4.setFont(font)

self.pushButton\_4.setLayoutDirection(QtCore.Qt.RightToLeft)

self.pushButton\_4.setDefault(True)

self.pushButton\_4.setObjectName("pushButton\_4")

self.pushButton\_5 = QtWidgets.QPushButton(self.groupBox\_2)

self.pushButton\_5.setGeometry(QtCore.QRect(10, 65, 171, 31))

font = QtGui.QFont()

font.setFamily("Segoe Print")

font.setPointSize(12)

self.pushButton\_5.setFont(font)

self.pushButton\_5.setDefault(True)

self.pushButton\_5.setObjectName("pushButton\_5")

self.pushButton\_6 = QtWidgets.QPushButton(self.groupBox\_2)

self.pushButton\_6.setGeometry(QtCore.QRect(10, 100, 171, 31))

font = QtGui.QFont()

font.setFamily("Segoe Print")

font.setPointSize(12)

self.pushButton\_6.setFont(font)

self.pushButton\_6.setDefault(True)

self.pushButton\_6.setObjectName("pushButton\_6")

self.label\_4 = QtWidgets.QLabel(self.groupBox\_2)

self.label\_4.setGeometry(QtCore.QRect(27, 33, 25, 25))

self.label\_4.setText("")

self.label\_4.setPixmap(QtGui.QPixmap("data/1f4d5.png"))

self.label\_4.setScaledContents(True)

self.label\_4.setObjectName("label\_4")

self.label\_5 = QtWidgets.QLabel(self.groupBox\_2)

self.label\_5.setGeometry(QtCore.QRect(28, 68, 27, 25))

self.label\_5.setText("")

self.label\_5.setPixmap(QtGui.QPixmap("data/kybok.png"))

self.label\_5.setScaledContents(True)

self.label\_5.setObjectName("label\_5")

self.label\_6 = QtWidgets.QLabel(self.groupBox\_2)

self.label\_6.setGeometry(QtCore.QRect(30, 104, 24, 24))

self.label\_6.setText("")

self.label\_6.setPixmap(QtGui.QPixmap("data/274c.png"))

self.label\_6.setScaledContents(True)

self.label\_6.setObjectName("label\_6")

self.calendarWidget = QtWidgets.QCalendarWidget(self.centralwidget)

self.calendarWidget.setGeometry(QtCore.QRect(270, 210, 241, 141))

self.calendarWidget.setObjectName("calendarWidget")

self.label\_7 = QtWidgets.QLabel(self.centralwidget)

self.label\_7.setGeometry(QtCore.QRect(210, 360, 301, 131))

self.label\_7.setText("")

self.label\_7.setPixmap(QtGui.QPixmap("data/plane-2.png"))

self.label\_7.setScaledContents(True)

self.label\_7.setWordWrap(False)

self.label\_7.setOpenExternalLinks(False)

self.label\_7.setObjectName("label\_7")

MainWindow.setCentralWidget(self.centralwidget)

self.menubar = QtWidgets.QMenuBar(MainWindow)

self.menubar.setGeometry(QtCore.QRect(0, 0, 520, 21))

self.menubar.setObjectName("menubar")

MainWindow.setMenuBar(self.menubar)

self.statusbar = QtWidgets.QStatusBar(MainWindow)

self.statusbar.setObjectName("statusbar")

MainWindow.setStatusBar(self.statusbar)

#подключаем кнопки к функциям

self.pushButton\_6.clicked.connect(sys.exit)

self.pushButton.clicked.connect(self.savegr)

self.pushButton\_2.clicked.connect(self.gruz)

self.pushButton\_3.clicked.connect(self.och)

self.menubar = QtWidgets.QMenuBar(MainWindow)

self.menubar.setGeometry(QtCore.QRect(0, 0, 800, 22))

self.menubar.setObjectName("menubar")

self.menu2 = QtWidgets.QMenu(self.menubar)

self.menu2.setObjectName("menu2")

self.menu3 = QtWidgets.QMenu(self.menubar)

self.menu3.setObjectName("menu3")

MainWindow.setMenuBar(self.menubar)

self.action1 = QtWidgets.QAction(MainWindow)

self.action1.setObjectName("action1")

self.action4 = QtWidgets.QAction(MainWindow)

self.action4.setObjectName("action4")

self.action5 = QtWidgets.QAction(MainWindow)

self.action5.setObjectName("action4")

self.menu2.addAction(self.action1)

self.menu2.addAction(self.action4)

self.menu2.addAction(self.action5)

self.menubar.addAction(self.menu2.menuAction())

self.menubar.addAction(self.menu3.menuAction())

self.action1.triggered.connect(self.ddaction1)

self.action4.triggered.connect(self.ddaction2)

self.action5.triggered.connect(self.ddaction3)

self.retranslateUi(MainWindow)

QtCore.QMetaObject.connectSlotsByName(MainWindow)

MainWindow.setWindowIcon(QtGui.QIcon('data/sam.png'))

def ddaction1(self):

QMessageBox.about(MainWindow, "О программе", "Программный комплекс «Автоматизация учета исправности и использования самолетов в эксплуатационном авиапредприятии»\n\nПрограмма обеспечивает работу диспетчера производственно- диспетчерского отдела АТБ по регистрации состояний парка ВС на диспетчерском графике использования и технического обслуживания самолетов.")

def ddaction2(self):

QMessageBox.about(MainWindow, "Об авторах", "Авторы:\n\nШеметов Даниил Александрович\nЖабакова Джамиля Сержановна\n\nКурсанты 432 группы.")

def ddaction3(self):

QMessageBox.about(MainWindow, "Инструкция", "1. Пользователь заполняет вручную график на день;\n2. Далее выбирает структурировать данные в виде табеля или отчета;\n3. Выбирает месяц и год, и нажимает кнопку.")

def retranslateUi(self, MainWindow):

\_translate = QtCore.QCoreApplication.translate

MainWindow.setWindowTitle(\_translate("MainWindow", "DM"))

self.groupBox.setTitle(\_translate("MainWindow", "Действия"))

self.pushButton.setText(\_translate("MainWindow", " Сохранение графика"))

self.pushButton\_2.setText(\_translate("MainWindow", " Загрузка графика"))

self.pushButton\_3.setText(\_translate("MainWindow", " Очистка графика"))

self.groupBox\_2.setTitle(\_translate("MainWindow", "Переход"))

self.pushButton\_4.setText(\_translate("MainWindow", "Табель"))

self.pushButton\_5.setText(\_translate("MainWindow", "Отчёт"))

self.pushButton\_6.setText(\_translate("MainWindow", "Выход"))

self.pushButton\_4.clicked.connect(self.show\_window\_1)

self.pushButton\_5.clicked.connect(self.show\_window\_2)

self.menu2.setTitle(\_translate("MainWindow", "Справка"))

self.action1.setText(\_translate("MainWindow", "О программе"))

self.action4.setText(\_translate("MainWindow", "Об авторах"))

self.action5.setText(\_translate("MainWindow", "Инструкция"))

#компиляция окна

if \_\_name\_\_ == "\_\_main\_\_":

import sys

app = QtWidgets.QApplication(sys.argv)

MainWindow = QtWidgets.QMainWindow()

ui = Ui\_MainWindow()

ui.setupUi(MainWindow)

MainWindow.show()

sys.exit(app.exec\_())