1. **ЛИСТИНГ ПРОГРАММЫ**

import sys

from PyQt5 import QtCore, QtGui, QtWidgets

from PyQt5.QtCore import QPropertyAnimation, QRect,QDate

from PyQt5.QtWidgets import QTableWidgetItem as QWT

from uis2 import\*

import traceback

import sqlite3

from PyQt5.QtWidgets import QMessageBox

import time

def log\_uncaught\_exceptions(ex\_cls, ex, tb):

text = '{}: {}:\n'.format(ex\_cls.\_\_name\_\_, ex)

text += ''.join(traceback.format\_tb(tb))

print(text)

sys.exit()

sys.excepthook = log\_uncaught\_exceptions

class MyWin(QtWidgets.QMainWindow):

def \_\_init\_\_(self, parent=None):

QtWidgets.QWidget.\_\_init\_\_(self, parent)

self.ui = Ui\_MainWindow()

self.ui.setupUi(self)

self.ui.Btn\_Toggle.clicked.connect(lambda:self.toggleMenu( 400, True))

self.ui.pushButton\_3.clicked.connect(lambda:self.toggleMenu( 400, True))

self.ui.pushButton\_6.clicked.connect(lambda:self.ui.stackedWidget.setCurrentIndex(0))

self.ui.pushButton.clicked.connect(lambda:self.ui.stackedWidget.setCurrentIndex(1))

self.ui.pushButton\_2.clicked.connect(lambda:self.ui.stackedWidget.setCurrentIndex(2))

self.ui.listWidget.currentItemChanged.connect(self.cool) #при нажатие на листвиджет

self.ui.tableWidget.cellClicked.connect(self.tab)

#self.ui.tableWidget.cellEntered.connect(self.tab)

self.ui.pushButton\_4.clicked.connect(self.save)

self.ui.comboBox.activated.connect(self.tabels)

dats = self.ui.calendarWidget.selectedDate()

self.ds=dats.toString('dd-MM-yyyy')

yyy=int(self.ds[6:])

m=int(self.ds[3:5])

d=int(self.ds[:2])

self.ui.calendarWidget.clicked.connect(self.Datt)

#self.ui.calendarWidget.clicked.connect(self.tabels)

self.ui.calendarWidget.setMaximumDate(QDate(yyy, m, d))#максимальная дата

#DateTime

self.timer = QtCore.QTimer()

self.timer.timeout.connect(self.tick)

self.timer.start(1000)

self.Tablo()

#Удаление заголовков

self.ui.tableWidget.verticalHeader().hide()

self.ui.tableWidget.horizontalHeader().hide()

self.ui.tableWidget\_2.verticalHeader().hide()

self.ui.tableWidget\_2.horizontalHeader().hide()

self.ui.tableWidget\_3.verticalHeader().hide()

self.ui.tableWidget\_3.horizontalHeader().hide()

self.Datt()

self.da = self.ui.calendarWidget.selectedDate().toString('MM')

self.Mo(self.da)

self.ui.calendarWidget.clicked.connect(self.Mo)

def Mo(self, x):

self.junuary=[]

self.february=[]

self.marth=[]

self.april=[]

self.may=[]

self.juny=[]

self.july=[]

self.august=[]

self.semtember=[]

self.october=[]

self.november=[]

self.december=[]

x=self.ui.calendarWidget.selectedDate().toString('MM')

if x == '01':

self.ui.label\_6.setText('январь')

self.ui.label\_7.setText('январь')

for i in range(1,31):

if len(str(i))==1:

i='0'+str(i)

i=str(i)

i+='-01-2020'

self.junuary.append(str(i))

self.mes = 1

if x == '02':

self.ui.label\_6.setText('февраль')

self.ui.label\_7.setText('февраль')

for i in range(1,31):

if len(str(i))==1:

i='0'+str(i)

i=str(i)

i+='-02-2020'

self.february.append(str(i))

self.mes = 2

if x == '03':

self.ui.label\_6.setText('март')

self.ui.label\_7.setText('март')

for i in range(1,31):

if len(str(i))==1:

i='0'+str(i)

i=str(i)

i+='-03-2020'

self.marth.append(str(i))

self.mes = 3

if x == '04':

self.ui.label\_6.setText('апрель')

self.ui.label\_7.setText('апрель')

for i in range(1,31):

if len(str(i))==1:

i='0'+str(i)

i=str(i)

i+='-04-2020'

self.april.append(str(i))

self.mes = 4

if x == '05':

self.ui.label\_6.setText('май')

self.ui.label\_7.setText('май')

for i in range(1,31):

if len(str(i))==1:

i='0'+str(i)

i=str(i)

i+='-05-2020'

self.may.append(str(i))

self.mes = 5

if x == '06':

self.ui.label\_6.setText('июнь')

self.ui.label\_7.setText('июнь')

for i in range(1,31):

if len(str(i))==1:

i='0'+str(i)

i=str(i)

i+='-06-2020'

self.juny.append(str(i))

self.mes = 6

if x == '07':

self.ui.label\_6.setText('июль')

self.ui.label\_7.setText('июль')

for i in range(1,31):

if len(str(i))==1:

i='0'+str(i)

i=str(i)

i+='-07-2020'

self.july.append(str(i))

self.mes = 7

if x == '08':

self.ui.label\_6.setText('август')

self.ui.label\_7.setText('август')

for i in range(1,31):

if len(str(i))==1:

i='0'+str(i)

i=str(i)

i+='-08-2020'

self.august.append(str(i))

self.mes = 8

if x == '09':

self.ui.label\_6.setText('сентябрь')

self.ui.label\_7.setText('сентябрь')

for i in range(1,31):

if len(str(i))==1:

i='0'+str(i)

i=str(i)

i+='-09-2020'

self.semtember.append(str(i))

self.mes = 9

if x == '10':

self.ui.label\_6.setText('октябрь')

self.ui.label\_7.setText('октябрь')

for i in range(1,31):

if len(str(i))==1:

i='0'+str(i)

i=str(i)

i+='-10-2020'

self.october.append(str(i))

self.mes = 10

if x == '11':

self.ui.label\_6.setText('ноябрь')

self.ui.label\_7.setText('ноябрь')

for i in range(1,31):

if len(str(i))==1:

i='0'+str(i)

i=str(i)

i+='-11-2020'

self.november.append(str(i))

self.mes = 11

if x == '12':

self.ui.label\_6.setText('декабрь')

self.ui.label\_7.setText('декабрь')

for i in range(1,31):

if len(str(i))==1:

i='0'+str(i)

i=str(i)

i+='-12-2020'

self.december.append(str(i))

self.mes = 12

def Tablo(self):

#Отчет

for i in range(20):

self.ui.tableWidget\_2.setSpan(0,i,7,1)

self.ui.tableWidget\_2.setSpan(0,4,2,1)

for i in range(5,10):

self.ui.tableWidget\_2.setSpan(0,i,1,1)

for i in range(10,20):

self.ui.tableWidget\_2.setSpan(0,i,1,1)

self.ui.tableWidget\_2.setSpan(1,i,2,1)

self.ui.tableWidget\_2.setSpan(0,4,1,1)

self.ui.tableWidget\_2.setSpan(0,4,1,20)

self.ui.tableWidget\_2.setSpan(1,4,1,6)

self.ui.tableWidget\_2.setSpan(2,5,1,5)

self.ui.tableWidget\_2.setSpan(1,10,2,10)

self.ui.tableWidget\_2.setSpan(2,4,5,1)

for i in range(4,20):

self.ui.tableWidget\_2.setSpan(3,i,4,1)

#Табель

self.ui.tableWidget\_3.setSpan(0,0,2,1)

self.ui.tableWidget\_3.setSpan(0,1,2,2)

self.ui.tableWidget\_3.setSpan(0,3,2,1)

self.ui.tableWidget\_3.setSpan(0,4,1,30)

self.ui.tableWidget\_3.setSpan(0,34,2,1)

for i in range(2,30):

self.ui.tableWidget\_3.setSpan(i,1,1,2)

#График

for i in range(1,145,6):

self.ui.tableWidget.setSpan(0,i,1,6)

self.ui.tableWidget.setSpan(0,0,2,1)

def tick(self):

self.time2 = time.strftime('%Y-%m-%d/%H:%M:%S')#получаем текущее время и дату

self.ui.label\_2.setText(self.time2)

def func(self,bo1,bo):

for i in bo1:

for j in i:

if j==self.d:

pass

else:

bo.append(j)

def Datt(self):#функция при нажатии на календарь

labell = QtWidgets.QLabel()

date = self.ui.calendarWidget.selectedDate()

self.dd=date.toString('dd-MM-yyyy')

if self.dd==self.ds: #если выбранная дата = сегодняшнее число, заполняем все ячейки пустотой

for i in range (self.ui.tableWidget.columnCount()):

for j in range (self.ui.tableWidget.rowCount()):

self.ui.tableWidget.setCellWidget(j,i,labell)

self.d=str(self.dd)#выводим записи с бд по дате

connection = sqlite3.connect('iotat.db')

cursor = connection.cursor()

cursor.execute("SELECT \* FROM Ту134 WHERE Дата=?",(self.d,))

resultstu = cursor.fetchall()

tu1 = resultstu

tu=[]

self.func(tu1,tu)

cursor.execute("SELECT \* FROM Ил96 WHERE Дата=?",(self.d,))

resultsil = cursor.fetchall()

il1= resultsil

il=[]

self.func(il1,il)

cursor.execute("SELECT \* FROM Боинг767 WHERE Дата=?",(self.d,))

resultsil = cursor.fetchall()

bo1= resultsil

bo=[]

self.func(bo1,bo)

cursor.execute("SELECT \* FROM Боинг777 WHERE Дата=?",(self.d,))

resultsil = cursor.fetchall()

bo71= resultsil

bo7=[]

self.func(bo71,bo7)

cursor.execute("SELECT \* FROM Ту154 WHERE Дата=?",(self.d,))

resultsil = cursor.fetchall()

tu11= resultsil

tu1=[]

self.func(tu11,tu1)

if not tu and not il and not bo and not bo7 and not tu1 and self.dd!=self.ds:#если все ячейки пустые

qmsgBox = QMessageBox()

qmsgBox.setStyleSheet('QMessageBox {background-color: white;}')

QMessageBox.about(qmsgBox,"Предупреждение ","Таблица не заполнена")

else:

self.insTb(tu,il,bo,bo7,tu1,self.ui.tableWidget)

connection.commit()

connection.close()

def tabels(self):

connection = sqlite3.connect('iotat.db')

cursor = connection.cursor()

currText = self.ui.comboBox.currentText()

currText=currText.replace('-', '')

self.spismes = [self.junuary,self.february, self.marth, self.april, self.may, self.juny, self.july, self.august, self.semtember, self.october, self.november, self.december]

col=4

self.vlenk=0

self.vlene=0

self.vlenm=0

self.vleng=0

self.vlena=0

self.vleny=0

self.vlenob=0

self.vlentb=0

self.vlenop=0

self.vlentp=0

self.vlendv=0

self.vlensh=0

self.vlenor=0

self.vlenz=0

self.vlenr=0

self.vlend=0

self.vlenj=0

self.vlenjr=0

self.vlenl=0

self.vlenv=0

self.vlens=0

vii=0

vni=0

if self.mes==1:

poop = self.spismes[0]

if self.mes==2:

poop = self.spismes[1]

if self.mes==3:

poop = self.spismes[2]

if self.mes==4:

poop = self.spismes[3]

if self.mes==5:

poop = self.spismes[4]

if self.mes==6:

poop = self.spismes[5]

if self.mes==7:

poop = self.spismes[6]

if self.mes==8:

poop = self.spismes[7]

if self.mes==9:

poop = self.spismes[8]

if self.mes==10:

poop = self.spismes[9]

if self.mes==11:

poop = self.spismes[10]

if self.mes==12:

poop = self.spismes[11]

for i in poop:

self.lenk=0

self.lene=0

self.lenm=0

self.leng=0

self.lena=0

self.leny=0

self.lenob=0

self.lentb=0

self.lenop=0

self.lentp=0

self.lendv=0

self.lensh=0

self.lenor=0

self.lenz=0

self.lenr=0

self.lend=0

self.lenj=0

self.lenjr=0

self.lenl=0

self.lenv=0

self.lens=0

cursor.execute(f"SELECT \* FROM {currText} WHERE Дата=?",(i,))

resultstu = cursor.fetchall()

tu1 = resultstu

sp=[]

for j in tu1:

for i in j:

if i=='К':

self.lenk+=1

self.vlenk+=1

if i=='K':

self.lenk+=1

self.vlenk+=1

if i=='Е':

self.lene+=1

self.vlene+=1

if i=='М':

self.lenm+=1

self.vlenm+=1

if i=='Г':

self.leng+=1

self.vleng+=1

if i=='А':

self.lena+=1

self.vlena+=1

if i=='У':

self.leny+=1

self.vleny+=1

if i=='Об':

self.lenob+=1

self.vlenob+=1

if i=='Тб':

self.lentb+=1

self.vlentb+=1

if i=='Оп':

self.lenop+=1

self.vlenop+=1

if i=='Тп':

self.lentp+=1

self.vlentp+=1

if i=='Ш':

self.lensh+=1

self.vlensh+=1

if i=='Ор':

self.lenor+=1

self.vlenor+=1

if i=='Р':

self.lenr+=1

self.vlenr+=1

if i=='З':

self.lenz+=1

self.vlenz+=1

if i=='Дв':

self.lendv+=1

self.vlendv+=1

if i=='Д':

self.lend+=1

self.vlend+=1

if i=='Ж':

self.lenj+=1

self.vlenj+=1

if i=='Жр':

self.lenjr+=1

self.vlenjr+=1

if i=='Л':

self.lenl+=1

self.vlenl+=1

if i=='В':

self.lenv+=1

self.vlenv+=1

if i=='С':

self.lens+=1

self.vlens+=1

#print(self.vlenz)

self.vsego=[self.vlenk,self.vlene, self.vlenm,self.vleng,self.vlena,self.vleny,self.vlenob,self.vlentb,self.vlenop,self.vlentp,self.vlendv,self.vlensh,self.vlenor,self.vlenz,self.vlenr,self.vlend,self.vlenj,self.vlenjr,self.vlenl,self.vlenv,self.vlens]

sp.append(self.lenk)

sp.append(self.lene)

sp.append(self.lenm)

sp.append(self.leng)

sp.append(self.lena)

sp.append(self.leny)

sp.append(self.lenob)

sp.append(self.lentb)

sp.append(self.lenop)

sp.append(self.lentp)

sp.append(self.lendv)

sp.append(self.lensh)

sp.append(self.lenor)

sp.append(self.lenr)

sp.append(self.lenz)

sp.append(self.lend)

sp.append(self.lenj)

sp.append(self.lenjr)

sp.append(self.lenl)

sp.append(self.lenv)

sp.append(self.lens)

s=0

#print(self.vsego)

ii=0#испрвные

ni=0#неисправные

for k in range(5,10):

a = str(sp[s]\*10//60)

b = str(sp[s]\*10%60)

if len(a) == 1:

a='0'+a

if len(b) == 1:

b='0'+b

volk = a+':'+b

ii+=sp[s]

vii+=sp[s]

self.ui.tableWidget\_3.setItem(k,col, QWT(str(volk)))

vsa = str(self.vsego[s]\*10//60)

vsb = str(self.vsego[s]\*10%60)

if len(vsa) == 1:

vsa = '0'+vsa

if len(vsb) == 1:

vsb = '0'+vsb

vse = vsa+':'+vsb

self.ui.tableWidget\_3.setItem(k,34, QWT(str(vse)))

s+=1

vsa = str(ii\*10//60)

vsb = str(ii\*10%60)

if len(vsa) == 1:

vsa = '0'+vsa

if len(vsb) == 1:

vsb = '0'+vsb

vseii = vsa+':'+vsb

self.ui.tableWidget\_3.setItem(3,col, QWT(str(vseii)))

s=5

for l in range(12,22):

a = str(sp[s]\*10//60)

b = str(sp[s]\*10%60)

if len(a) == 1:

a='0'+a

if len(b) == 1:

b='0'+b

volk = a+':'+b

ni+=sp[s]

vni+=sp[s]

self.ui.tableWidget\_3.setItem(l,col, QWT(str(volk)))

vsa = str(self.vsego[s]\*10//60)

vsb = str(self.vsego[s]\*10%60)

if len(vsa) == 1:

vsa = '0'+vsa

if len(vsb) == 1:

vsb = '0'+vsb

vse = vsa+':'+vsb

self.ui.tableWidget\_3.setItem(l,34, QWT(str(vse)))

s+=1

a = str(sp[9]\*10//60)

b = str(sp[9]\*10%60)

s=10

if len(a) == 1:

a='0'+a

if len(b) == 1:

b='0'+b

volk = a+':'+b

self.ui.tableWidget\_3.setItem(22,col, QWT(str(volk)))

vsa = str(self.vsego[s]\*10//60)

vsb = str(self.vsego[s]\*10%60)

if len(vsa) == 1:

vsa = '0'+vsa

if len(vsb) == 1:

vsb = '0'+vsb

vse = vsa+':'+vsb

self.ui.tableWidget\_3.setItem(22,34, QWT(str(vse)))

s=15

for i in range(23,29):

a = str(sp[s]\*10//60)

b = str(sp[s]\*10%60)

if len(a) == 1:

a='0'+a

if len(b) == 1:

b='0'+b

volk = a+':'+b

ni+=sp[s]

vni+=sp[s]

self.ui.tableWidget\_3.setItem(i,col, QWT(str(volk)))

vsa = str(self.vsego[s]\*10//60)

vsb = str(self.vsego[s]\*10%60)

if len(vsa) == 1:

vsa = '0'+vsa

if len(vsb) == 1:

vsb = '0'+vsb

vse = vsa+':'+vsb

self.ui.tableWidget\_3.setItem(i,34, QWT(str(vse)))

s+=1

nia = str(ni\*10//60)

nib = str(ni\*10%60)

if len(nia) == 1:

nia = '0'+nia

if len(nib) == 1:

nia = '0'+nib

vseni = nia+':'+nib

self.ui.tableWidget\_3.setItem(10,col, QWT(str(vseni)))

self.ui.tableWidget\_3.setItem(2,col, QWT(str('24:00')))

self.ui.tableWidget\_3.setItem(3,34, QWT(str(vii)))

self.ui.tableWidget\_3.setItem(10,34, QWT(str(vni)))

self.ui.tableWidget\_3.setItem(2,34, QWT(str(vni+vii)))

col+=1

self.ui.tableWidget\_2.setItem(7,4, QWT(str(vii)))

def save(self):

value=''

count=self.ui.tableWidget.columnCount()

connection = sqlite3.connect('iotat.db')

cursor = connection.cursor()

cursor.execute("SELECT \* FROM Ту134")

cursor.execute("SELECT \* FROM Ил96")

cursor.execute("SELECT \* FROM Боинг767")

cursor.execute("SELECT \* FROM Боинг777")

cursor.execute("SELECT \* FROM Ту154"

il96=[]

tu134=[]

bo767=[]

bo777=[]

tu154=[]

for i in range(self.ui.tableWidget.columnCount()+1):

bo767.append('0')

tu134.append('0')

il96.append('0')

tu154.append('0')

bo777.append('0')

bo767[0]=self.ds

il96[0]=self.ds

tu134[0]=self.ds

tu154[0]=self.ds

bo777[0]=self.ds

for i in range (145):

value+='?,'

value+='?'

try:

for i in range (self.ui.tableWidget.columnCount()):

bo777[i+1]=self.ui.tableWidget.item(4,i+1).text()

except:

pass

try:

for i in range (self.ui.tableWidget.columnCount()):

bo767[i+1]=self.ui.tableWidget.item(3,i+1).text()

except:

pass

try:

for i in range (self.ui.tableWidget.columnCount()):

il96[i+1]=self.ui.tableWidget.item(2,i+1).text()

except:

pass

try:

for i in range (self.ui.tableWidget.columnCount()):

tu134[i+1]=self.ui.tableWidget.item(5,i+1).text()

except:

pass

try:

for i in range (self.ui.tableWidget.columnCount()):

tu154[i+1]=(self.ui.tableWidget.item(6,i+1).text())

except:

pass

try:

print(self.ds)

cursor.execute(f'INSERT INTO Ил96 VALUES({value})',tuple(il96))

except:

cursor.execute ('DELETE FROM Ил96 where Дата =?',(self.ds,))

cursor.execute(f'INSERT INTO Ил96 VALUES({value})',tuple(il96))

try:

cursor.execute(f'INSERT INTO Боинг777 VALUES({value})',tuple(bo777))

except:

cursor.execute ('DELETE FROM Боинг777 where Дата =?',(self.ds,))

cursor.execute(f'INSERT INTO Боинг777 VALUES({value})',tuple(bo777))

try:

cursor.execute(f'INSERT INTO Боинг767 VALUES({value})',tuple(bo767))

except:

cursor.execute ('DELETE FROM Боинг767 where Дата =?',(self.ds,))

cursor.execute(f'INSERT INTO Боинг767 VALUES({value})',tuple(bo767))

try:

cursor.execute(f'INSERT INTO Ту134 VALUES({value})',tuple(tu134))

except:

cursor.execute ('DELETE FROM Ту134 where Дата =?',(self.ds,))

cursor.execute(f'INSERT INTO Ту134 VALUES({value})',tuple(tu134))

try:

cursor.execute(f'INSERT INTO Ту154 VALUES({value})',tuple(tu154))

except:

cursor.execute ('DELETE FROM Ту154 where Дата =?',(self.ds,))

cursor.execute(f'INSERT INTO Ту154 VALUES({value})',tuple(tu154))

connection.commit()

connection.close()

def tab(self):#при нажатие на ячейку таблицы

date = self.ui.calendarWidget.selectedDate()

self.dd=date.toString('dd-MM-yyyy')

if self.dd==self.ds:#если дата=сегдняшня дата, то разрешено ситывать индекс и в него заносить картинку

self.label = QtWidgets.QLabel()

self.label.setPixmap(self.pic)

for index in self.ui.tableWidget.selectedIndexes():

self.ui.tableWidget.setItem(index.row(),index.column(), QWT(self.spis))

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

else:

pass

def cool(self):

self.spisimame=["1.bmp","2.bmp","3.bmp","4.bmp","5.bmp","6.bmp","7.bmp","8.bmp","9.bmp","10.bmp","11.bmp","12.bmp","13.bmp","14.bmp","15.bmp","16.bmp","17.bmp","18.bmp","19.bmp","20.bmp","21.bmp"]

numberlist = self.ui.listWidget.currentRow()

self.spis=''

if numberlist==0:

self.pic = QtGui.QPixmap(self.spisimame[0])

self.spis='К'

if numberlist==1:

self.pic = QtGui.QPixmap(self.spisimame[1])

self.spis='Е'

if numberlist==2:

self.pic = QtGui.QPixmap(self.spisimame[2])

self.spis='М'

if numberlist==3:

self.pic = QtGui.QPixmap(self.spisimame[3])

self.spis='Г'

if numberlist==4:

self.pic = QtGui.QPixmap(self.spisimame[4])

self.spis='А'

if numberlist==5:

self.pic = QtGui.QPixmap(self.spisimame[5])

self.spis='У'

if numberlist==6:

self.pic = QtGui.QPixmap(self.spisimame[6])

self.spis='Об'

if numberlist==7:

self.pic = QtGui.QPixmap(self.spisimame[7])

self.spis='Тб'

if numberlist==8:

self.pic = QtGui.QPixmap(self.spisimame[8])

self.spis='Оп'

if numberlist==9:

self.pic = QtGui.QPixmap(self.spisimame[9])

self.spis='Тп'

if numberlist==10:

self.pic = QtGui.QPixmap(self.spisimame[10])

self.spis='Ш'

if numberlist==11:

self.pic = QtGui.QPixmap(self.spisimame[11])

self.spis='Ор'

if numberlist==12:

self.pic = QtGui.QPixmap(self.spisimame[12])

self.spis='Р'

if numberlist==13:

self.pic = QtGui.QPixmap(self.spisimame[13])

self.spis='З'

if numberlist==14:

self.pic = QtGui.QPixmap(self.spisimame[14])

self.spis='Дв'

if numberlist==15:

self.pic = QtGui.QPixmap(self.spisimame[15])

self.spis='Д'

if numberlist==16:

self.pic = QtGui.QPixmap(self.spisimame[16])

self.spis='Ж'

if numberlist==17:

self.pic = QtGui.QPixmap(self.spisimame[17])

self.spis='Жр'

if numberlist==18:

self.pic = QtGui.QPixmap(self.spisimame[18])

self.spis='Л'

if numberlist==19:

self.pic = QtGui.QPixmap(self.spisimame[19])

self.spis='В'

if numberlist==20:

self.pic = QtGui.QPixmap(self.spisimame[20])

self.spis='С'

def picture(self,k,j):

self.labels = QtWidgets.QLabel()

if str(k[j])=='0':

pics = QtGui.QPixmap("22.png")

self.labels.setPixmap(pics)

if str(k[j])=='К':

pics = QtGui.QPixmap("1.bmp")

self.labels.setPixmap(pics)

if str(k[j])=='K':

pics = QtGui.QPixmap("1.bmp")

self.labels.setPixmap(pics)

if str(k[j])=='Е':

pics = QtGui.QPixmap("2.bmp")

self.labels.setPixmap(pics)

if str(k[j])=='М':

pics = QtGui.QPixmap("3.bmp")

self.labels.setPixmap(pics)

if str(k[j])=='Г':

pics = QtGui.QPixmap("4.bmp")

self.labels.setPixmap(pics)

if str(k[j])=='А':

pics = QtGui.QPixmap("5.bmp")

self.labels.setPixmap(pics)

if str(k[j])=='У':

pics = QtGui.QPixmap("6.bmp")

self.labels.setPixmap(pics)

if str(k[j])=='Об':

pics = QtGui.QPixmap("7.bmp")

self.labels.setPixmap(pics)

if str(k[j])=='Тб':

pics = QtGui.QPixmap("8.bmp")

self.labels.setPixmap(pics)

if str(k[j])=='Оп':

pics = QtGui.QPixmap("9.bmp")

self.labels.setPixmap(pics)

if str(k[j])=='Тп':

pics = QtGui.QPixmap("10.bmp")

self.labels.setPixmap(pics)

if str(k[j])=='Ш':

pics = QtGui.QPixmap("11.bmp")

self.labels.setPixmap(pics)

if str(k[j])=='Ор':

pics = QtGui.QPixmap("12.bmp")

self.labels.setPixmap(pics)

if str(k[j])=='Р':

pics = QtGui.QPixmap("13.bmp")

self.labels.setPixmap(pics)

if str(k[j])=='З':

pics = QtGui.QPixmap("14.bmp")

self.labels.setPixmap(pics)

if str(k[j])=='Дв':

pics = QtGui.QPixmap("15.bmp")

self.labels.setPixmap(pics)

if str(k[j])=='Д':

pics = QtGui.QPixmap("16.bmp")

self.labels.setPixmap(pics)

if str(k[j])=='Ж':

pics = QtGui.QPixmap("17.bmp")

self.labels.setPixmap(pics)

if str(k[j])=='Жр':

pics = QtGui.QPixmap("18.bmp")

self.labels.setPixmap(pics)

if str(k[j])=='Л':

pics = QtGui.QPixmap("19.bmp")

self.labels.setPixmap(pics)

if str(k[j])=='В':

pics = QtGui.QPixmap("20.bmp")

self.labels.setPixmap(pics)

if str(k[j])=='С':

pics = QtGui.QPixmap("21.bmp")

self.labels.setPixmap(pics)

def insTb(self,k,il,bo,bo7,tu1,tbl):

s=''

for j in range(len(k)):

self.picture(k,j)

tbl.setCellWidget(5,j+1,self.labels)

for j in range(len(bo)):

self.picture(bo,j)

tbl.setCellWidget(3,j+1,self.labels)

for j in range(len(bo7)):

self.picture(bo7,j)

tbl.setCellWidget(4,j+1,self.labels)

for j in range(len(tu1)):

self.picture(tu1,j)

tbl.setCellWidget(6,j+1,self.labels)

for j in range(len(il)):

self.picture(il,j)

tbl.setCellWidget(2,j+1,self.labels)

def toggleMenu(self, maxWidth, enable):

if enable:

width = self.ui.frame\_left\_menu.width()

maxExtend = maxWidth

standard = 70

if width ==70:

widthExtended = maxExtend

else:

widthExtended = standard

# ANIMATION

self.animation = QPropertyAnimation(self.ui.frame\_left\_menu, b"minimumWidth")

self.animation.setDuration(800)

self.animation.setStartValue(width)

self.animation.setEndValue(widthExtended)

self.animation.setEasingCurve(QtCore.QEasingCurve.InOutQuart)

self.animation.start()

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

app = QtWidgets.QApplication(sys.argv)

myapp = MyWin()

myapp.show()

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