Kód aplikace

```
from asyncio.windows_events import NULL
from operator import contains
from statistics import quantiles
from tkinter import *
from tkinter import ttk
import tkinter as tk
from tkinter import messagebox
from PIL import ImageTk,Image
from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit, execute
from numpy import pi
from qiskit import IBMQ, Aer, assemble, transpile

def pulPryc(pole):
    if len(pole)<6:
        qr = QuantumRegister(1, 'q')
        cr = ClassicalRegister(1, 'c')
    else:
        qr = QuantumRegister(3, 'q')
        cr = ClassicalRegister(3, 'c')

    obvod = QuantumCircuit(qr, cr)

    if len(pole)<6:
        obvod.h(qr[0])
    else:
        obvod.h(qr[2])
        obvod.h(qr[1])
        obvod.h(qr[0])

    if len(pole)<6:
        obvod.u((-pole[0])*pi, 0, 0, qr[0])
        obvod.u((pole[1])*pi, 0, 0, qr[0])
    else:
        obvod.u((-pole[0])*pi, 0, 0, qr[2])
        obvod.u((-pole[2])*pi, 0, 0, qr[1])
        obvod.u((-pole[4])*pi, 0, 0, qr[0])
        obvod.u((pole[1])*pi, 0, 0, qr[2])
        obvod.u((pole[3])*pi, 0, 0, qr[1])
        obvod.u((pole[5])*pi, 0, 0, qr[0])

    if len(pole)<6:
        obvod.measure(qr[0], cr[0])
    else:
        obvod.measure(qr[2], cr[2])
        obvod.measure(qr[1], cr[1])
        obvod.measure(qr[0], cr[0])

    simulator = Aer.get_backend('qasm_simulator')
    job = execute(obvod, simulator, shots=2000)
    pocty = job.result().get_counts()

    return pocty

def nejPravdepodobnejsi(pocty):
    pomoc =0
    index=""
    for x in pocty:
        if pocty[x] > pomoc:
            pomoc = pocty[x]
            index = x
    return index
```

```
def najdiMensi(index, pole):
    poleP = []
    neco=[i for i in index]
    i=0
    for x in range(0,6,2):
        if len(neco)<=i:
            break
        else:
            if  int(neco[i]) == 0:
                poleP.append(pole[x])
            else:
                poleP.append(pole[x+1])
            i=i+1
    return poleP

def odebrat(totopole, zpole, cesta):
    for x in totopole:
        cesta.remove(cesta [zpole.index(x)])
        zpole.remove(x)

def loopOdebrat(vzdalenosti, vstup):
    while len(vzdalenosti)>1:
        odebrat(najdiMensi(nejPravdepodobnejsi(pulPryc(vzdalenosti)), vzdalenosti), vzdalenosti, vstup)

def uprava (vzdalenosti):
    for x in range(len(vzdalenosti)):
        vzdalenosti[x] = vzdalenosti[x]/1000

def odchodoveOkno():
    odchod = Toplevel()
    odchod.title("KVNav")
    odchod.geometry("200x100")
    otazka = Label(odchod, text = "Opravdu chcete odejit?")
    Ano = Button(odchod, text = "Ano", command=lambda: [odchod.destroy(), apka.destroy()])
    Ne  = Button(odchod, text = "Ne", command = odchod.destroy)
    otazka.grid(row = 0, column = 0, columnspan = 2, rowspan = 1, pady = 5)
    Ano.grid(row = 1, column = 0, columnspan = 1, rowspan = 1)
    Ne.grid(row = 1, column = 1, columnspan = 1, rowspan = 1)
    odchod.mainloop()

def vysledneOkno():
    vysledek = Toplevel()
    vysledek.title("KVNav")
    vysledek.geometry("894x720")

    vstup =
[Checkbutton1.get(),Checkbutton2.get(),Checkbutton3.get(),Checkbutton4.get(),Checkbutton5.get(),Checkbutton6
.get(),Checkbutton7.get()]
    start = vyberStart.get()
    vzdalenosti= []

    canvas = Canvas(
        vysledek,
        width = 854,
        height = 500
        )

    informace = Canvas(
        vysledek,
        width = 854,
        height = 200
        )
```

```
    nejkratsiCesta(vstup, start, vzdalenosti)
    canvas.pack()
    informace.pack()
    informace.create_text(10, 10, anchor=NW, text= "Trasa: " + vypisCestu(vstup), fill="black", font=('Helvetica 12'))
    informace.create_text(10, 30, anchor=NW, text= "Vzdalenost: " + str( int(vzdalenosti[0]*1000)) + " km",
fill="black", font=('Helvetica 12'))
    poleCesta = str(vstup[0])
    mapa = ImageTk.PhotoImage(Image.open('img/mapaPrototyp.png'))

    canvas.create_image(
          10,
          10,
          anchor=NW,
          image=mapa
          )

    for x in range(len(poleCesta)-1):
        cesta = poleCesta[x] + poleCesta[x+1]
        if "01" == cesta or "10" == cesta:
            prhCb = ImageTk.PhotoImage(Image.open('img/prhCb.png'))
            canvas.create_image(
                10,
                10,
                anchor=NW,
                image=prhCb
                )
        if "12" == cesta or "21" == cesta:
            prhBrno= ImageTk.PhotoImage(Image.open('img/prhBrno.png'))
            canvas.create_image(
                  10,
                  10,
                  anchor=NW,
                  image=prhBrno
                  )
        if "15" == cesta or "51" == cesta:
            prhHk = ImageTk.PhotoImage(Image.open('img/prhHk.png'))
            canvas.create_image(
                  10,
                  10,
                  anchor=NW,
                  image=prhHk
                  )
        if "16" == cesta or "61" == cesta:
            prhPlz = ImageTk.PhotoImage(Image.open('img/prhPlz.png'))
            canvas.create_image(
                  10,
                  10,
                  anchor=NW,
                  image=prhPlz
                  )
        if "14" == cesta or "41" == cesta:
            prhOv = ImageTk.PhotoImage(Image.open('img/prhOv.png'))
            canvas.create_image(
                 10,
                  10,
                  anchor=NW,
                  image=prhOv
                  )
```

```
if "13" == cesta or "31" == cesta:
    prhLbc = ImageTk.PhotoImage(Image.open('img/prhLbc.png'))
    canvas.create_image(
            10,
            10,
            anchor=NW,
            image=prhLbc
            )
if "02" == cesta or "20" == cesta:
    cbBrno = ImageTk.PhotoImage(Image.open('img/cbBrno.png'))
    canvas.create_image(
            10,
            10,
            anchor=NW,
            image=cbBrno
            )
if "06" == cesta or "60" == cesta:
    cbPlz = ImageTk.PhotoImage(Image.open('img/cbPlz.png'))
    canvas.create_image(
            10,
            10,
            anchor=NW,
            image=cbPlz
            )
if "03" == cesta or "30" == cesta:
    cbLbc= ImageTk.PhotoImage(Image.open('img/cbLbc.png'))
    canvas.create_image(
            10,
            10,
            anchor=NW,
            image=cbLbc
            )
if "04" == cesta or "40" == cesta:
    cbOv = ImageTk.PhotoImage(Image.open('img/cbOv.png'))
    canvas.create_image(
            10,
            10,
            anchor=NW,
            image=cbOv
            )
if "05" == cesta or "50" == cesta:
    cbHk = ImageTk.PhotoImage(Image.open('img/cbHk.png'))
    canvas.create_image(
            10,
            10,
            anchor=NW,
            image=cbHk
            )
if "24" == cesta or "42" == cesta:
    brnoOv = ImageTk.PhotoImage(Image.open('img/brnoOv.png'))
    canvas.create_image(
            10,
            10,
            anchor=NW,
            image=brnoOv
            )
```

```
if "52" == cesta or "25" == cesta:
    brnoHk = ImageTk.PhotoImage(Image.open('img/brnoHk.png'))
    canvas.create_image(
            10,
            10,
            anchor=NW,
            image=brnoHk
            )
if "62" == cesta or "26" == cesta:
    brnoPlz = ImageTk.PhotoImage(Image.open('img/brnoPlz.png'))
    canvas.create_image(
            10,
            10,
            anchor=NW,
            image=brnoPlz
            )
if "32" == cesta or "23" == cesta:
    brnoLbc = ImageTk.PhotoImage(Image.open('img/brnoLbc.png'))
    canvas.create_image(
            10,
            10,
            anchor=NW,
            image=brnoLbc
            )
if "43" == cesta or "34" == cesta:
    ovLbc = ImageTk.PhotoImage(Image.open('img/ovLbc.png'))
    canvas.create_image(
            10,
            10,
            anchor=NW,
            image=ovLbc
            )
if "46" == cesta or "64" == cesta:
    ovPlz = ImageTk.PhotoImage(Image.open('img/ovPlz.png'))
    canvas.create_image(
            10,
            10,
            anchor=NW,
            image=ovPlz
            )
if "45" == cesta or "54" == cesta:
    ovHk = ImageTk.PhotoImage(Image.open('img/ovHk.png'))
    canvas.create_image(
            10,
            10,
            anchor=NW,
            image=ovHk
            )
if "35" == cesta or "53" == cesta:
    lbcHk = ImageTk.PhotoImage(Image.open('img/lbcHk.png'))
    canvas.create_image(
            10,
            10,
            anchor=NW,
            image=lbcHk
            )
```

```
        if "36" == cesta or "63" == cesta:
            lbcPlz = ImageTk.PhotoImage(Image.open('img/lbcPlz.png'))
            canvas.create_image(
                    10,
                    10,
                    anchor=NW,
                    image=lbcPlz
                    )
        if "65" == cesta or "56" == cesta:
            plzHk = ImageTk.PhotoImage(Image.open('img/plzHk.png'))
            canvas.create_image(
                    10,
                    10,
                    anchor=NW,
                    image=plzHk
                    )

    vysledek.mainloop()

vzdalenosti =[
                [0, 148, 213, 250, 380, 253, 134],
                [148, 0, 208, 110 , 373, 117, 92],
                [213,208, 0, 309, 170, 171, 298],
                [250,   110, 309, 0, 437, 101, 203],
                [380,   373, 170, 437, 0, 242, 466],
                [253,   117, 171, 101, 242, 0, 208],
                [134,   92,          298, 203, 466, 208, 0]
               ]

def naMesto(cislo):
        menic = {
            0 : "Ceske Budejovice",
            1  : "Praha",
            2 : "Brno",
            3 : "Liberec",
            4 : "Ostrava",
            5 : "Hradec Kralove",
            6 : "Plzen"
                }
        return menic[cislo]

def naCislo(mesto):
        menic = {
            "Ceske Budejovice" : 0,
            "Praha" : 1,
            "Brno" : 2,
            "Liberec" : 3,
            "Ostrava" : 4,
            "Hradec Kralove" : 5,
            "Plzen" : 6
                }
        return menic[mesto]

def naIndex1 (vstup):
    for x in range(len(vstup)):
        vstup[x] = vstup[x]-1
```

```
def vycistit(vstup):
    pole = []
    for x in range(len(vstup)):
        if vstup[x] != 0:
            pole.append(vstup[x])
    vstup.clear()
    for x in pole:
        vstup.append(x)

def naIndex (vstup):
    if vstup!= []:
        for x in range(len(vstup)):
            vstup[x] = vstup[x]-1
    else:
        messagebox.showwarning("showwarning", "Warning")
def duplicita (vstup , start):
    if start in vstup:
        vstup.remove(start)

def permutace(cisla):
    seznam = []
    for x in cisla:
        if len(cisla)==1:
            seznam.append(str(x))
            break
        else:
         for y in cisla:
             if len(cisla)==2:
                if x == y:
                    continue
                else:
                    seznam.append(str(x)+str(y))
                    break
             for z in cisla:
                 if len(cisla)==3:
                    if x == y or x == z or y == z:
                        continue
                    else:
                        seznam.append(str(x)+str(y)+str(z))
                        break
                 for w in cisla:
                     if len(cisla)==4:
                        if x == y or x == z or x==w or y == z or y==w or z==w:
                            continue
                        else:
                            seznam.append(str(x)+str(y)+str(z)+str(w))
                            break
                     for v in cisla:
                         if len(cisla)==5:
                            if x == y or x == z or x==w or x==v or y == z or y==w or y==v or z==w or z==v or w==v:
                                continue
                            else:
                                seznam.append(str(x)+str(y)+str(z)+str(w)+str(v))
                                break
                         for f in cisla:
                             if len(cisla)==6:
                                if x==y or x==z or x==w or x==v or x==f or y == z or y==w or y==v or y==f or z==w or z==v or
z==f or w==v or w==f or v==f:
                                    continue
                                else:
                                    seznam.append(str(x)+str(y)+str(z)+str(w)+str(v)+str(f))
                                    break
```

```
    cisla.clear()
    for x in seznam:
        cisla.append(x)

def zacKon(permutace, start):
    for x in range(len(permutace)):
        permutace[x] = str(start) + str(permutace[x]) + str(start)

def vzdalenost(trasa):
    delka=0
    for x in range(len(trasa)-1):
        delka = delka + vzdalenosti[int(trasa[int(x)])] [int(trasa[int(x)+1])]
    return delka

def shoda(vstup ,vzdalenosti):
    pole = []
    pole1 = []
    for x in vzdalenosti:
        if  x in pole:
            continue
        else:
            pole.append(x)
            pole1.append( vstup[ vzdalenosti.index( x ) ] )
    vstup.clear()
    vzdalenosti.clear()
    for x in pole:
        vzdalenosti.append(x)
    for x in pole1:
        vstup.append(x)

def vypisCestu(vstup):
    cesta = ""
    for y in vstup:
        vstup1 = [i for i in y]
        for x in vstup1:
            if cesta=="":
                cesta = naMesto(int(x))
            else:
                cesta = cesta + " => " + naMesto(int(x))
    return cesta

def uka():
    vstup =
[Checkbutton1.get(),Checkbutton2.get(),Checkbutton3.get(),Checkbutton4.get(),Checkbutton5.get(),Checkbutton6
.get(),Checkbutton7.get()]
    start = vyberStart.get()
    vzdalenosti = []
    vycistit(vstup)
    naIndex(vstup)
    duplicita(vstup, naCislo(start))
    permutace(vstup)
    zacKon(vstup,naCislo(start))
    for x in vstup:
        vzdalenosti.append(vzdalenost(x))
    shoda(vstup, vzdalenosti)
    uprava(vzdalenosti)
    loopOdebrat(vzdalenosti, vstup)
```

```
def nejkratsiCesta(vstup, start, vzdalenosti):
    vycistit(vstup)
    naIndex(vstup)
    duplicita(vstup, naCislo(start))
    permutace(vstup)
    zacKon(vstup,naCislo(start))
    for x in vstup:
        vzdalenosti.append(vzdalenost(x))
    shoda(vstup, vzdalenosti)
    uprava(vzdalenosti)
    loopOdebrat(vzdalenosti, vstup)

apka = Tk()
apka.geometry('250x400')
apka.title('KVNav')

ZacatekText = Label(apka, text = "Zacatecni destinace")

n = tk.StringVar()
vyberStart = ttk.Combobox(apka, width = 27, textvariable = n)

vyberStart['values'] = ("Ceske Budejovice",
          "Praha",
          "Brno",
          "Liberec",
          "Ostrava",
          "Hradec Kralove",
          "Plzen")
vyberStart.current(0)

Checkbutton1 = IntVar()
Checkbutton2 = IntVar()
Checkbutton3 = IntVar()
Checkbutton4 = IntVar()
Checkbutton5 = IntVar()
Checkbutton6 = IntVar()
Checkbutton7 = IntVar()

zaskrtnuti1 = Checkbutton(apka, text = "Ceske Budejovice",
                  variable = Checkbutton1,
                  onvalue = 1,
                  offvalue = 0,
                  height = 2,
                  width = 15)
zaskrtnuti2 = Checkbutton(apka, text = "Praha",
                  variable = Checkbutton2,
                  onvalue = 2,
                  offvalue = 0,
                  height = 2,
                  width = 10)
zaskrtnuti3 = Checkbutton(apka, text = "Brno",
                  variable = Checkbutton3,
                  onvalue = 3,
                  offvalue = 0,
                  height = 2,
                  width = 10)
zaskrtnuti4 = Checkbutton(apka, text = "Liberec",
                  variable = Checkbutton4,
                  onvalue = 4,
                  offvalue = 0,
                  height = 2,
                  width = 10)
```

```
zaskrtnuti5 = Checkbutton(apka, text = "Ostrava",
                    variable = Checkbutton5,
                    onvalue = 5,
                    offvalue = 0,
                    height = 2,
                    width = 10)
zaskrtnuti6 = Checkbutton(apka, text = "Hradec Kralove",
                    variable = Checkbutton6,
                    onvalue = 6,
                    offvalue = 0,
                    height = 2,
                    width = 10)
zaskrtnuti7 = Checkbutton(apka, text = "Plzen",
                    variable = Checkbutton7,
                    onvalue = 7,
                    offvalue = 0,
                    height = 2,
                    width = 10)

tlacitko = Button(apka, text = "Naviguj", bd = 5, command = vysledneOkno )
tlacitkoKonec = Button(apka, text = "Konec", command = odchodoveOkno )

ZacatekText.grid(row = 0, column = 0, columnspan = 2, pady = 20)
vyberStart.grid(row = 1, column = 0, columnspan = 2, rowspan = 1)
zaskrtnuti1.grid(row = 2, column = 0, columnspan = 1, rowspan = 1)
zaskrtnuti2.grid(row = 3, column = 0, columnspan = 1, rowspan = 1)
zaskrtnuti3.grid(row = 4, column = 0, columnspan = 1, rowspan = 1)
zaskrtnuti4.grid(row = 5, column = 0, columnspan = 1, rowspan = 1)
zaskrtnuti5.grid(row = 2, column = 1, columnspan = 1, rowspan = 1)
zaskrtnuti6.grid(row = 3, column = 1, columnspan = 1, rowspan = 1)
zaskrtnuti7.grid(row = 4, column = 1, columnspan = 1, rowspan = 1)
tlacitko.grid(row = 6, column = 0, columnspan = 2, rowspan = 1, pady = 20)
tlacitkoKonec.grid(row = 7, column = 0, columnspan = 2, rowspan = 1)

apka.mainloop()
```