**AMALIY MASHG‘ULOT UCHUN O‘QUV MATERIALLARI**

**2-Mavzu:** PyQt5 paketi va QtDesigner dasturi yordamida GUI dasturlarini yaratish.

**7-mashg‘ulot.** Matn muharriri dasturini yozish.

**O‘quv savollari:**

1. PyQt5 da matn muharriri dizaynidagi elementlarning funksiyalari uchun dastur yozish.
2. Dasturni testlash.

**1. PyQt5 da matn muharriri dizaynidagi elementlarning funksiyalari uchun dastur yozish**

Ro‘yxat ob’ektining nomi  listWidget bo‘lib , bu holda yaxshi. Dizaynni loyiha papkasida design.ui sifatida saqlang.

**Dizaynni kodga aylantirish**

Albatta foydalanishingiz mumkin. ui fayllarini to‘g‘ridan-to‘g‘ri Python kodidan olish mumkin, ammo osonroq ko‘rinishi mumkin bo‘lgan yana bir usul mavjud. Siz kodni o‘zgartirishingiz mumkin. ui faylini Python fayliga kiritamiz, keyin uni import qilishimiz va ishlatishimiz mumkin. Buning uchun terminal/buyruqlar satridan **pyuic5** buyrug‘idan foydalanamiz.

Konvertatsiya qilish uchun. ui faylini Python faylida design.py deb nomlagan holda quyidagi buyruqdan foydalaning:

$ pyuic5 path/to/ design.ui -o output/path/to/design.py

**Kodni yozishni boshlaymi**

Endi bizda QtDesigner dasturida yaratilgan dizayn faylining python faylga o‘girilgan versiyasi design.py nomli fayli bor va biz uning mantiqiy qismini ya’ni asosiy dastur qismini yozish ustida ishlashni boshlaymiz.

Birinchi navbatda **design.py** joylashgan papkada main.py faylini yarating.

**QtDesigner dasturida yaratilgan dizayndan foydalanish**

Python GUI ilovasi uchun sizga quyidagi modullar kerak bo‘ladi:

import sys

from PyQt5 import QtWidgets

Bizga avval yaratilgan dizayn kodi ham kerak, shuning uchun biz uni ham import qilamiz:

import design # Bu bizning o‘zgartirilgan dizayn faylimiz

Dizayn fayli har safar dizayn o‘zgartirilganda butunlay qayta yozilishi sababli, biz uni o‘zgartirmaymiz. Buning o‘rniga biz yangi klass yaratamiz,  ExampleApp , uni barcha funksiyalaridan foydalanish uchun dizayn kodi bilan birlashtiramiz:

Class ExampleApp(QtWidgets.QMainWindow, design.Ui\_MainWindow):

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

# Bu o‘zgaruvchilarga, usullarga kirish uchun bu erda kerak

# va hokazo. design.py faylida

        super().\_\_init\_\_()

self.setupUi(self)

# Bu dizaynimizni ishga tushirish uchun kerak

Ushbu sinfda biz interfeys elementlari bilan o‘zaro aloqada bo‘lamiz, ulanishlar va bizga kerak bo‘lgan barcha narsalarni qo‘shamiz. Lekin birinchi navbatda, biz kodni ishga tushirganimizda sinfni ishga tushirishimiz kerak. Buni asosiy ( ) funksiyada hal qilamiz  :

def main():

app = QtWidgets.QApplication(sys.argv)

window = ExampleApp()

window.show()

app.exec\_()

Va bu funktsiyani bajarish uchun biz tanish qurilishdan foydalanamiz:

if \_\_name\_\_ == '\_\_main\_\_':

main()

Natijada main.py quyidagicha ko‘rinadi:

import sys

from PyQt5 import QtWidgets

import design

class ExampleApp(QtWidgets.QMainWindow, design.Ui\_MainWindow):

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

super().\_\_init\_\_()

self.setupUi(self)

def main():

app = QtWidgets.QApplication(sys.argv)

window = ExampleApp()

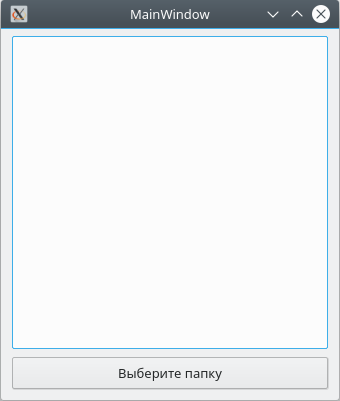
window.show()

app.exec\_()

if \_\_name\_\_ == '\_\_main\_\_':

main()

Agar biz ushbu kodni ishga tushirsak: $ python3 main.py ilovamiz ishga tushadi!



Lekin tugmani bosish hech narsa qilmaydi, shuning uchun biz u bilan shug‘ullanishimiz kerak.

**Python GUI ilovamizga funksionallikni qo‘shish**

Eslatma  Qo‘shimcha barcha kodlar ExampleApp sinfida yozilgan  .

*Keling, “****papkani tanlash****”* tugmasidan boshlaylik. Siz hodisani tugmani bosish kabi funksiyaga bog‘lashingiz mumkin:

            self.btnBrowse.clicked.connect(self.browse\_folder)

Ushbu qatorni ilova ExampleApp sinfining \_\_ init \_\_  metodiga qo‘shib qo‘ying. Endi buni batafsil ko‘rib chiqaylik:

* self.btnBrowse : bu yerda  btnBrowse - biz Qt Designer’da  aniqlagan ob’ektning nomi . o‘z  o‘zi uchun gapiradi va hozirgi sinfga tegishli bo‘lganligini anglatadi;
* cliked - biz bog‘lamoqchi bo‘lgan voqea . Turli elementlarda turli hodisalar mavjud, masalan, ro‘yxat vidjetlarida  itemSelectionChanged  va boshqalar mavjud;
* connect ( ) - o‘tkazilgan funksiyaga qo‘ng‘iroq bilan hodisani bog‘laydigan usul;
* self.browse\_folder        \_folder bu biz ExampleApp sinfida  tasvirlangan funksiya (metod) xolos.

Papka tanlash dialogini ochish uchun biz o‘rnatilgan QtWidgets.QFileDialog.getExistingDirectory metodidan foydalanishimiz mumkin :

            katalog = QtWidgets.QFileDialog.getExistingDirectory (self, "Papkani tanlash")

Agar foydalanuvchi katalogni tanlasa,  katalog o‘zgaruvchisi  tanlangan katalogning mutlaq yo‘liga o‘rnatiladi, aks holda u  Yo‘q ga o‘rnatiladi . Agar foydalanuvchi dialog oynasini yopsa, kodni boshqa bajarmaslik uchun biz  if katalogidan foydalanamiz : buyrug‘i.

Katalog tarkibini ko‘rsatish uchun biz  os ni import qilishimiz kerak :

            import os

Va shunday tarkib ro‘yxatini oling:

            os.listdir (path)

listWidget- ga elementlar qo‘shish uchun  addItem ( ) usulidan foydalanamiz va barcha elementlarni o‘chirish uchun bizda  self.listWidget.clear () mavjud.

Natijada,  browse\_folder funksiyasi  quyidagicha ko‘rinishi kerak:

def browse\_folder(self):

self.listWidget.clear()

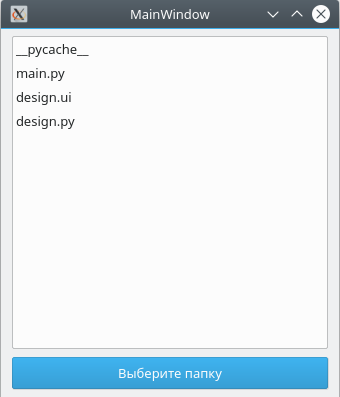
directory = QtWidgets.QFileDialog.getExistingDirectory (self, "papkani tanlash")

if directory:

for file\_name in os.listdir(directory):

self.listWidget.addItem(file\_name)

Endi, agar biz dasturni ishga tushirsak, tugmani bosing va katalogni tanlang, biz ko‘ramiz:



Python GUI ilovamizning butun kodi shunday ko‘rinadi:

import sys

import os

from PyQt5 import QtWidgets

import design

class ExampleApp(QtWidgets.QMainWindow, design.Ui\_MainWindow):

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

super().\_\_init\_\_()

self.setupUi(self)

self.btnBrowse.clicked.connect(self.browse\_folder)

def browse\_folder(self):

self.listWidget.clear()

directory = QtWidgets.QFileDialog.getExistingDirectory(self, "tanlash")

if directory:

for file\_name in os.listdir(directory):

self.listWidget.addItem(file\_name)

def main():

app = QtWidgets.QApplication(sys.argv)

window = ExampleApp()

window.show()

app.exec\_()

if \_\_name\_\_ == '\_\_main\_\_':

main()

Bular Python GUI ilovasini ishlab chiqish uchun Qt Designer va PyQt dan foydalanish asoslari edi. Endi siz dastur dizaynini o‘zingining ideyalaringiz bilan o‘zgartirishingiz va kodini o‘zgartirish uchun **pyuic5** buyrug‘idan qanday foydalanishni bilasiz.

**Nazorat savollari:**

1. Pythonda gui bilan ishlash uchun qanday kutubxonalari mavjud?
2. PyQT5 paketini qanday o‘rnatiladi va chaqirib ishlatish qanday amalga oshiriladi?
3. Pyuic5 buyrug‘i nima uchun foydalaniladi?
4. Papkalarni ochish uchun qanday kutubxonadan foydalaniladi?
5. Kataloglar ustida qanday amallar bajarish mumkin?
6. Faylni saqlash uchun qanday amal bajariladi?