# OOP2 Final - 2. Soru Tüm Uygulama Rehberi (Qt Designer + Kodlama)

Bu rehber, OOP2 final sınavındaki 2. sorunun (liste işlemleri GUI uygulaması) baştan sona tüm adımlarıyla açıklamasını içerir. Hiç kod bilmeyen bir öğrenci bile bu rehber sayesinde Qt Designer'dan Python kodlamaya kadar tüm süreci tamamlayabilir.

## 1️⃣ Qt Designer ile Arayüz Oluşturma

Soru bizden bir GUI (görsel arayüz) uygulaması istiyor. Bu uygulama PyQt5 + Qt Designer ile oluşturulmalı.  
İstenenler:  
- Sayı üretmek için 1 buton  
- Sonuçları göstermek için 2 tane Text Edit  
- Durumu yazmak için 1 adet QLabel  
- Kullanıcının işlem seçmesi için 4 adet Toggle Button (QRadioButton)  
- Sıralama yönünü seçmek için 1 ComboBox

|  |  |  |
| --- | --- | --- |
| Bileşen | Görev | objectName |
| QPushButton | Sayı üretmek | generateButton |
| QTextEdit | Üretilen sayıları göstermek | numbersTextEdit |
| QTextEdit | Sonuçları göstermek | resultTextEdit |
| QLabel | Durum bilgisi (boş) | statusLabel |
| QRadioButton | Toplama işlemi | sumRadio |
| QRadioButton | Ortalama işlemi | avgRadio |
| QRadioButton | Maksimum bulma | maxRadio |
| QRadioButton | Sıralama işlemi | sortRadio |
| QComboBox | Sıralama yönü (Ascending, Descending) | sortComboBox |

Tüm bu bileşenler Qt Designer'dan Widget Box kısmından sürüklenip bırakılır. ComboBox içine 2 seçenek eklenmelidir: Ascending ve Descending.  
ObjectName'ler kodda erişim sağlamak için zorunludur.

## 2️⃣ logic.py – Sayı ve Liste İşlemleri

Bu dosya sadece işlemleri yapar, ekrana hiçbir şey yazmaz. Fonksiyonlar şunlardır:  
- generate\_random\_numbers(): 40 adet 0–99 arasında rastgele sayı üretir  
- calculate\_sum(numbers): listedeki sayıları toplar  
- calculate\_avg(numbers): ortalama hesaplar  
- calculate\_max(numbers): en büyük sayıyı verir  
- sort\_numbers(numbers, ascending): sayıları sıralar

## 3️⃣ app\_logic.py – Arayüz ve İşlem Bağlantıları

Bu dosya, Qt Designer'da çizdiğimiz arayüzü 'uic.loadUi("list\_ui.ui", self)' ile yükler. Arayüzdeki butonlara tıklanınca ne olacağını burada belirleriz.  
- generateButton'a tıklanırsa: sayı üretilir ve numbersTextEdit'e yazılır  
- RadioButton seçilirse: uygun işlem yapılır (sum, avg, max, sort)  
- ComboBox aktifleşir sadece 'sort' seçildiyse  
- Sonuçlar resultTextEdit'e yazılır, durumlar statusLabel'da görünür

## 4️⃣ main.py – Uygulamayı Başlatan Dosya

Bu dosya sadece uygulamayı başlatır. Kodu kısadır:  
from PyQt5.QtWidgets import QApplication  
from app\_logic import ListOperationApp  
app = QApplication([])  
window = ListOperationApp()  
window.show()  
app.exec\_()

## 5️⃣ Neden Böyle Yaptık? (Sınav Açıklamaları)

- GUI istendiği için Qt Designer kullanıyoruz  
- Sayılarla işlem yapılacağı için ayrı bir logic.py dosyasında matematik yazdık  
- Arayüz kodu karışmasın diye app\_logic.py içine bağlantılar yazıldı  
- Her widget’a objectName verdik çünkü koddan erişim için gerekli  
- ComboBox başta kapalı çünkü sadece sıralamada kullanılacak  
- Kod sade, anlaşılır ve modüler yapıldı çünkü sınav bunu puanlıyor

## 📌 SONUÇ

Bu yapı sayesinde hem sınavdaki liste uygulaması sorusunu hatasız yapabilir, hem de benzer tüm GUI işlemlerini başarıyla geçebilirsin. Tek yapman gereken:  
1. Qt Designer ile list\_ui.ui dosyasını çizmek  
2. Python dosyalarını yerleştirmek (logic.py, app\_logic.py, main.py)  
3. 'python main.py' komutu ile uygulamayı çalıştırmak