# OOP2 Final - 2. Soru (Liste İşlemleri) Uygulama + Kod Rehberi

Bu belge, OOP2 final sınavındaki 2. sorunun Qt Designer arayüzü oluşturma sürecinden Python kodlamasına kadar tüm adımlarını içerir. Kodlama bilmeyen biri için adım adım, hatasız uygulama rehberidir.

## 1️⃣ Qt Designer - Arayüz Çizimi ve Yapılandırma

1. Qt Designer'ı aç → 'Main Window' seç → OK  
2. Widget Box panelinden aşağıdaki bileşenleri sürükleyip yerleştir:

|  |  |  |
| --- | --- | --- |
| Widget | Görev | objectName |
| QPushButton | Sayı üretme | generateButton |
| QTextEdit | Sayıları gösterme | numbersTextEdit |
| QTextEdit | Sonuçları gösterme | resultTextEdit |
| QLabel | Durum mesajı (boş) | statusLabel |
| QRadioButton | Toplama | sumRadio |
| QRadioButton | Ortalama | avgRadio |
| QRadioButton | Maksimum | maxRadio |
| QRadioButton | Sıralama | sortRadio |
| QComboBox | Sıralama yönü | sortComboBox |

3. ComboBox içerisine şu iki seçeneği ekle:  
 - Ascending  
 - Descending

4. Grid layout kullanımı zorunlu değildir ama önerilir. Widget’ları düzenli yerleştirin.

5. Dosyayı 'list\_ui.ui' olarak kaydet. Sonraki adım: bu .ui dosyasını Python ile bağlamak.

## 2️⃣ Python Dosyaları ve Kodlar

### 🔹 logic.py

import random  
  
def generate\_random\_numbers():  
 return [random.randint(0, 99) for \_ in range(40)]  
  
def calculate\_sum(numbers):  
 return sum(numbers)  
  
def calculate\_avg(numbers):  
 return sum(numbers) / len(numbers)  
  
def calculate\_max(numbers):  
 return max(numbers)  
  
def sort\_numbers(numbers, ascending=True):  
 return sorted(numbers, reverse=not ascending)

### 🔹 app\_logic.py

from PyQt5.QtWidgets import QMainWindow  
from PyQt5 import uic  
from logic import generate\_random\_numbers, calculate\_sum, calculate\_avg, calculate\_max, sort\_numbers  
  
class ListOperationApp(QMainWindow):  
 def \_\_init\_\_(self):  
 super().\_\_init\_\_()  
 uic.loadUi("list\_ui.ui", self)  
 self.numbers = []  
  
 self.sumRadio.setEnabled(False)  
 self.avgRadio.setEnabled(False)  
 self.maxRadio.setEnabled(False)  
 self.sortRadio.setEnabled(False)  
 self.sortComboBox.setEnabled(False)  
  
 self.generateButton.clicked.connect(self.generate\_numbers)  
 self.sumRadio.toggled.connect(self.perform\_operation)  
 self.avgRadio.toggled.connect(self.perform\_operation)  
 self.maxRadio.toggled.connect(self.perform\_operation)  
 self.sortRadio.toggled.connect(self.perform\_operation)  
 self.sortComboBox.currentIndexChanged.connect(self.perform\_operation)  
  
 def generate\_numbers(self):  
 self.numbers = generate\_random\_numbers()  
 self.numbersTextEdit.setPlainText(", ".join(map(str, self.numbers)))  
 self.statusLabel.setText("Numbers are generated")  
 self.enable\_operations()  
  
 def enable\_operations(self):  
 self.sumRadio.setEnabled(True)  
 self.avgRadio.setEnabled(True)  
 self.maxRadio.setEnabled(True)  
 self.sortRadio.setEnabled(True)  
 self.sortComboBox.setEnabled(False)  
  
 def perform\_operation(self):  
 if self.sumRadio.isChecked():  
 result = calculate\_sum(self.numbers)  
 self.resultTextEdit.setPlainText(str(result))  
 self.statusLabel.setText("Sum calculated")  
  
 elif self.avgRadio.isChecked():  
 result = calculate\_avg(self.numbers)  
 self.resultTextEdit.setPlainText(str(result))  
 self.statusLabel.setText("Average calculated")  
  
 elif self.maxRadio.isChecked():  
 result = calculate\_max(self.numbers)  
 self.resultTextEdit.setPlainText(str(result))  
 self.statusLabel.setText("Maximum calculated")  
  
 elif self.sortRadio.isChecked():  
 self.sortComboBox.setEnabled(True)  
 if self.sortComboBox.currentText() == "Ascending":  
 result = sort\_numbers(self.numbers, ascending=True)  
 else:  
 result = sort\_numbers(self.numbers, ascending=False)  
 self.resultTextEdit.setPlainText(", ".join(map(str, result)))  
 self.statusLabel.setText("Sorted")

### 🔹 main.py

from PyQt5.QtWidgets import QApplication  
from app\_logic import ListOperationApp  
import sys  
  
app = QApplication(sys.argv)  
window = ListOperationApp()  
window.show()  
sys.exit(app.exec\_())

## 📌 Önemli Notlar

- .ui dosyasını çizmek yeterli değildir. objectName'ler doğru verilmelidir.  
- Widget'lar yerleştirildikten sonra `File > Save As` ile .ui dosyası olarak kaydedilmeli.  
- Python tarafında butonlara `.clicked.connect(...)` ile bağlantı yapılmalıdır.  
- layout kullanmak kodda zorunlu değil ama görsel düzen sağlar.  
- Kodda `.setPlainText(...)` ile kutulara yazı yazılır, `.setText(...)` ile QLabel güncellenir.