

## Vežbe 12 - Zadaci

Instalirati virtuelno okruženje:

***sudo apt install python3-venv***

Napraviti virtuelno okruženje:

***python3 -m venv ime\_okruzenja***

Aktivirati virtuelno okruženje:

***source ime\_okruzenja/bin/activate***

Instalirati PySide6 komandom:

***pip3 install PySide6***

Otvoriti folder u kom je napravljeno virtuelno okruženje u VScode. Prilikom realizacije .py fajla, VScode bi trebao prepoznati interpreter u donjem desnom uglu (Python) a pored se nalazi verzija pajtona gde u zagradi pored broja verzije mora pisati (*'ime\_okruzenja':venv*). Ako to nije slučaj, kliknuti na verziju i iz padajućeg menija odabrati interpreter virtuelnog okruženja. Ako to nije moguće aktivirati ga manuelno kao što je objašnjeno u koracima iznad.

Dizajner bi trebao biti dostupan sa instalacijom pyside6. Pokreće se iz terminala komandom:

***pyside6-designer***

Za prevođenje ui dizajna u py fajl koristiti komandu: ***pyside6-uic ime\_fajla.ui > ime\_fajla.py***

Ako se pojavi greška:

***qt.qpa.plugin: Could not load the Qt platform plugin 'xcb' in " even though it was found. This application failed to start because no Qt platform plugin could be initialized. Reinstalling the application may fix this problem.***

***Available platform plugins are: eglfs, linuxfb, minimal, minimalegl, offscreen, vnc, xcb. Aborted (core dumped)***

Rešenje:

***sudo apt install '^libxcb.\*-dev'***

---

**Zadatak 1.** Realizovati aplikaciju koja koristeći QCharts iscrtava liniju na ekranu. Takođe, iscrtati stubičaste dijagrame.

```
1 import sys
2 from PySide6.QtWidgets import (QApplication, QMainWindow, QVBoxLayout, QWidget)
3 from PySide6 import QtCharts
4 from PySide6.QtCore import Qt
5
6
7 class MainWindow(QMainWindow):
8     def __init__(self):
9         super(MainWindow, self).__init__()
10
11         self.setGeometry(100, 100, 900, 700)
12
13         self.centralWidget = QWidget(self)
14         self.setCentralWidget(self.centralWidget)
15         self.layout = QVBoxLayout(self.centralWidget)
16
17         self.chart_view = QtCharts.QChartView(self)
18         self.layout.addWidget(self.chart_view)
19
20         chart = QtCharts.QChart()
21         chart.setTitle("Bar i Line grafik")
22
23         bar_series = QtCharts.QBarSeries()
24         bar_set = QtCharts.QBarSet("Podaci")
25         bar_set.append([1, 3, 5, 2, 4])
26         bar_series.append(bar_set)
27         chart.addSeries(bar_series)
28
29         line_series = QtCharts.QLineSeries()
30         line_series.setName('Linija')
31         line_series.append(1, 1)
32         line_series.append(2, 3)
33         line_series.append(3, 5)
34         line_series.append(4, 2)
35         line_series.append(5, 4)
36         chart.addSeries(line_series)
37
38         axis_x = QtCharts.QBarCategoryAxis()
39         axis_x.append(["Prvi", "Drugi", "Treci", "Cetvrti", "Peti"])
40         chart.addAxis(axis_x, Qt.AlignmentFlag.AlignBottom)
41         bar_series.attachAxis(axis_x)
42
43         axis_x_secondary = QtCharts.QValueAxis()
44         axis_x_secondary.setTickCount(5)
45         axis_x_secondary.setRange(1, 5)
```

```

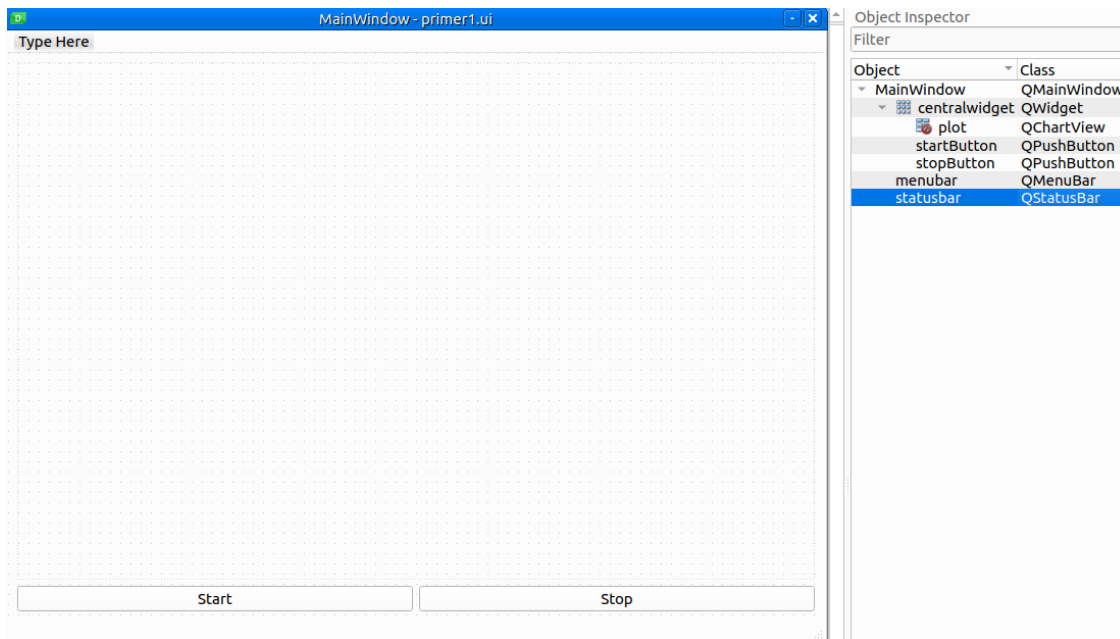
46     axis_x_secondary.setLabelFormat("%.0f")
47     chart.addAxis(axis_x_secondary, Qt.AlignmentFlag.AlignTop)
48     line_series.attachAxis(axis_x_secondary)
49
50     axis_y = QtCharts.QValueAxis()
51     axis_y.setRange(0, 6)
52     chart.addAxis(axis_y, Qt.AlignmentFlag.AlignLeft)
53     bar_series.attachAxis(axis_y)
54     line_series.attachAxis(axis_y)
55
56     self.chart_view.setChart(chart)
57
58
59 if __name__ == "__main__":
60     app = QApplication(sys.argv)
61     window = MainWindow()
62     window.show()
63     sys.exit(app.exec())

```

**Zadatak 2.** Realizovati aplikaciju koja prima podatke poslate preko serijskog porta i te podatke prikazuje na grafiku (QChart). Pokrenuti dizajner iz terminala komandom

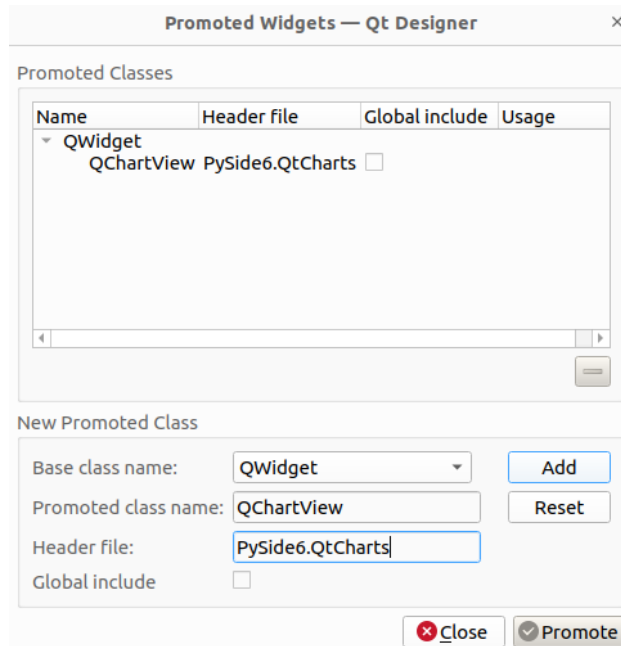
***pyside6-designer***

U MainWidnow dodati QWidget, i preimenovati ga u plot, dodati dva QPushButton-a i dati im imena startButton i stopButton. Sve elemente postaviti da budu poravnati u GridLayout-u. Izgled bi trebao biti kao što je prikazan na sledećoj slici.



Pošto ne postoji ugrađeni QChart widget u dizajneru, neophodno je izvršiti promociju widgeta u QChartWidget. Desnim klikom na widget odabrati opciju *Promote to...* nakon čega se otvara novi prozor. U njemu uneti *Base class name: QWidget, Promoted class name: QChartView, Header*

file: *PySide6.QtCharts*, *Global Include* se ne mora odabrati. Kliknuti na *Add*, a zatim na *Promote*. Podešavanja su data na sledećoj slici.



Sačuvati templejt kao *primer2.ui*, u folderu gde se želi realizovati aplikacija. Ponovo otvoriti terminal i izvršiti konverziju u python kod pozivom komande

***pyside6-uic primer2.ui -o primer2\_uic.py***

Kao izlaz dobija se novi python fajl *primer2\_uic.py* koji će biti učitani u glavni program.

```
1 import sys
2 from PySide6.QtCore import QThread, Signal, Slot
3 from PySide6.QtWidgets import QApplication, QMainWindow
4 from PySide6.QtCharts import QChart, QCategoryAxis, QValueAxis, QLineSeries
5 from primer2_ui import Ui_MainWindow
6 import numpy as np
7 import serial
8
9 class MainWindow(QMainWindow):
10     def __init__(self):
11         super(MainWindow, self).__init__()
12         self.ui = Ui_MainWindow()
13         self.ui.setupUi(self)
14         self.ui.startButton.clicked.connect(self.start_com)
15         self.ui.stopButton.clicked.connect(self.close_com)
16         self.th = Worker()
17         self.th.signalData.connect(self.update_chart)
18
19         self.chart = QChart()
20         self.ui.plot.setChart(self.chart)
```

```

21
22     self.series1 = QLineSeries()
23     self.series1.setName('Prvi kanal')
24     self.series2 = QLineSeries()
25     self.series2.setName('Drugi kanal')
26
27     self.chart.addSeries(self.series1)
28     self.chart.addSeries(self.series2)
29
30     axis_x = QCategoryAxis()
31     axis_x.setLabelsVisible(True)
32     axis_x.setRange(0, 100)
33     self.chart.setAxisX(axis_x, self.series1)
34     self.chart.setAxisX(axis_x, self.series2)
35
36     self.axis_y = QValueAxis()
37     self.axis_y.setLabelsVisible(True)
38     self.axis_y.setRange(0, 65535)
39     self.chart.setAxisY(self.axis_y, self.series1)
40     self.chart.setAxisY(self.axis_y, self.series2)
41
42
43     @Slot(np.ndarray, np.ndarray)
44     def update_chart(self, y1, y2):
45         self.series1.clear()
46         self.series2.clear()
47         #self.axis_y.setRange(min(min(y1), min(y2)), max(max(y1), max(y2)))
48         x = np.arange(len(y1))
49         for i in range(len(y1)):
50             self.series1.append(x[i], y1[i])
51             self.series2.append(x[i], y2[i])
52
53     def start_com(self):
54         self.th.start()
55         self.th.running = True
56         self.ui.startButton.setDisabled(True)
57         self.ui.stopButton.setDisabled(False)
58
59     def close_com(self):
60         self.th.running = False
61         self.th.serial.close()
62         self.ui.stopButton.setDisabled(True)
63         self.ui.startButton.setDisabled(False)
64
65     class Worker(QThread):
66         signalData = Signal(np.ndarray, np.ndarray)
67         def __init__(self):
68             super().__init__()

```

```

69         self.running = True
70
71     def run(self):
72         self.serial = serial.Serial('/dev/ttyUSB0', baudrate=115200)
73         y1 = np.zeros(100)
74         y2 = np.zeros(100)
75         while self.running:
76             try:
77                 text = self.serial.readline().decode()
78                 text = text.rstrip('\n').split(',')
79                 if len(text) == 2:
80                     y1 = np.roll(y1, -1)
81                     y1[-1] = float(text[0])
82                     y2 = np.roll(y2, -1)
83                     y2[-1] = float(text[1])
84                     self.signalData.emit(y1, y2)
85             except:
86                 print(text)
87
88 if __name__ == "__main__":
89     app = QApplication(sys.argv)
90     window = MainWindow()
91     window.show()
92     sys.exit(app.exec())

```

**Zadatak 3.** Realizovati aplikaciju koja koristeći pyqtgraph iscrtava liniju na ekranu. Takođe, realizovati i iscrtavanje samo pojedinih tačaka.

```

1  from PySide6.QtWidgets import QApplication, QMainWindow
2  import pyqtgraph as pg
3  import sys
4
5  class MainWindow(QMainWindow):
6
7      def __init__(self):
8          super(MainWindow, self).__init__()
9
10         self.graphWidget = pg.PlotWidget()
11         self.setCentralWidget(self.graphWidget)
12
13         hour = [1,2,3,4,5,6,7,8,9,10]
14         temperature = [19,22,25,27,27,25,23,20,17,14]
15
16         self.graphWidget.setBackground('w')
17
18         pen = pg.mkPen(color=(255, 0, 0), width = 3)
19         self.graphWidget.plot(hour, temperature, pen=pen)
20

```

```

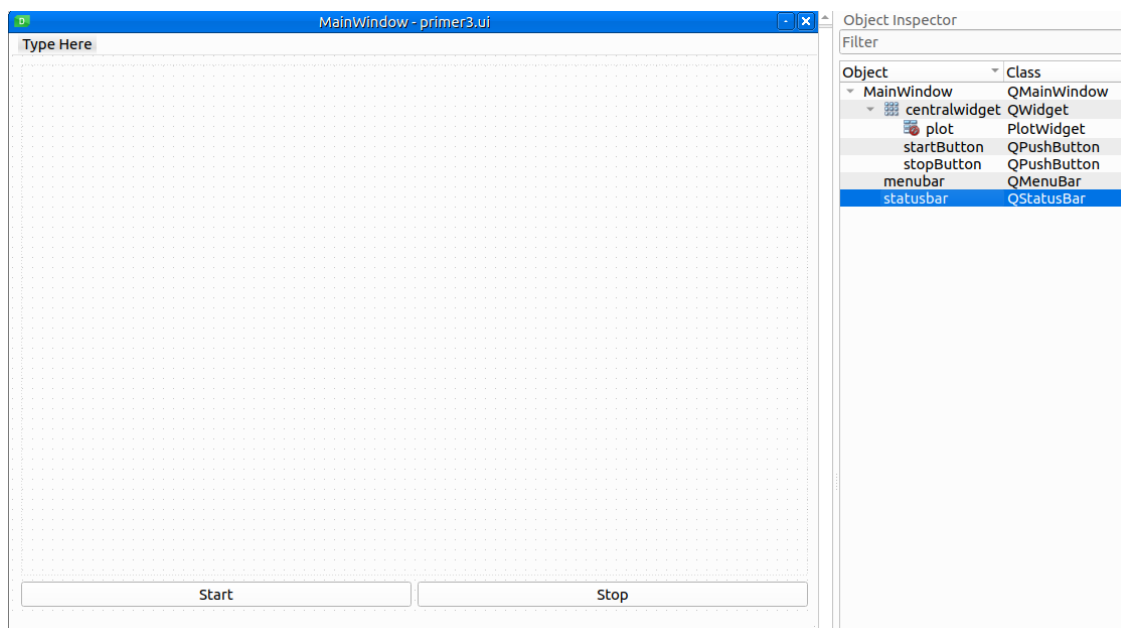
21         #scatter = pg.ScatterPlotItem(hour, temperature, brush='b')
22         #self.graphWidget.addItem(scatter)
23
24 app = QApplication(sys.argv)
25 w = MainWindow()
26 w.show()
27 app.exec()

```

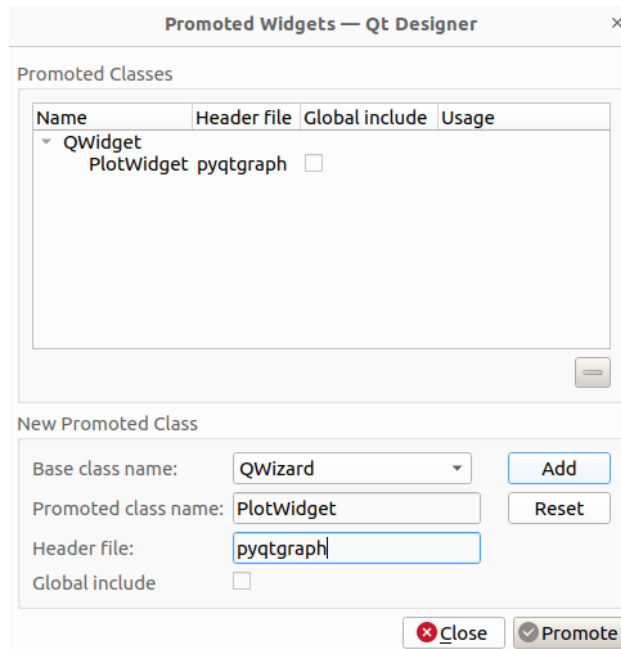
**Zadatak 4.** Realizovati aplikaciju koja prima podatke poslate preko serijskog porta i te podatke prikazuje na grafiku (pyqtgraph). Pokrenuti dizajner iz terminala komandom

*pySide6-designer*

U MainWidnow dodati QWidget, preimenovati ga u plot, dodati dva QPushButton-a i dati imena startButton i stopButton. Sve elemente postaviti da budu poravnani u GridLayout-u. Izgled bi trebao biti kao što je prikazan na sledećoj slici.



Pošto ne postoji ugrađeni pyqtgraph widget u dizajneru, neophodno je izvršiti promociju widgeta u PlotWidget. Desnim klikom na widget odabrati opciju *Promote to...* nakon čega se otvara novi prozor. U njemu uneti *Base class name: QWidget*, *Promoted class name: PlotWidget*, *Header file: pyqtgraph*, *Global Include* se ne mora odabrati. Izgled bi trebao biti kao što je prikazan na sledećoj slici.



Sačuvati templejt kao primer4.ui, u folderu gde se želi realizovati aplikacija. Ponovo otvoriti terminal i izvršiti konverziju u python kod pozivom komande

***pyside6-uc primer4.ui -o primer4\_uc.py***

Kao izlaz dobija se novi python fajl *primer4\_uc.py* koji će biti učitani u glavni program.

```

1 import sys
2 from PySide6 import QtCore, QtSerialPort
3 from PySide6.QtWidgets import QApplication, QMainWindow
4 from primer4_ui import Ui_MainWindow
5 import numpy as np
6
7 class MainWindow(QMainWindow):
8     def __init__(self):
9         super(MainWindow, self).__init__()
10        self.ui = Ui_MainWindow()
11        self.ui.setupUi(self)
12        self.x = np.zeros(100)
13        self.y1 = np.zeros(100)
14        self.y2 = np.zeros(100)
15        self.ui.startButton.clicked.connect(self.start_com)
16        self.ui.stopButton.clicked.connect(self.close_com)
17        self.serial = QtSerialPort.QSerialPort('/dev/ttyUSB0',
18                                             baudRate=QtSerialPort.QSerialPort.BaudRate.Baud115200,
19                                             readyRead=self.receive)
20
21    def receive(self):
22        while self.serial.canReadLine():
23            self.ui.plot.clear()
24            text = self.serial.readLine().data().decode()

```



```

25         text = text.rstrip('\n').split(',')
26         if len(text) == 2:
27             self.y1 = np.roll(self.y1, -1)
28             self.y1[-1] = float(text[0])
29             self.y2 = np.roll(self.y2, -1)
30             self.y2[-1] = float(text[1])
31         self.x = np.arange(len(self.y1))
32         #self.y = np.roll(self.y, -1)
33
34         self.ui.plot.plot(self.x,self.y1, pen='g', symbolSize=1,
35                           symbolPen='g', symbol='x')
36         self.ui.plot.plot(self.x,self.y2, pen='b', symbolSize=1,
37                           symbolPen='g', symbol='o')
38     def start_com(self):
39         if not self.serial.isOpen():
40             self.ui.startButton.setDisabled(True)
41             self.ui.stopButton.setDisabled(False)
42             self.serial.open(QIODevice.OpenModeFlag.ReadWrite)
43
44     def close_com(self):
45         self.serial.close()
46         self.ui.stopButton.setDisabled(True)
47         self.ui.startButton.setDisabled(False)
48
49 if __name__ == "__main__":
50     app = QApplication(sys.argv)
51     window = MainWindow()
52     window.show()
53     sys.exit(app.exec())

```