**Algorithm: Instructors**

Import 🡪 QtWidgets, QtGui 🡪 PyQt5

Import 🡪 Database as db, Timetable 🡪 components

import 🡪 Instructor as Parent 🡪 py\_ui

import 🡪 json

Class 🡨 Instructor

Function 🡨 \_init\_ (self, id)

self.id 🡨 id

self.dialog 🡨 dialog 🡨 QtWidgets.QDialog()

self.parent 🡨 parent 🡨 Parent.Ui\_Dialog()

parent 🡨 setupUi (dialog)

IF id

self.fillForm()

ELSE

# Create a new instance of timetable

self.table 🡨 Timetable.Timetable(parent.tableSchedule)

parent.btnFinish.clicked.connect(self.finish)

parent.btnCancel.clicked.connect(self.dialog.close)

dialog.exec\_()

def function 🡨 fillForm(self)

conn 🡨 db.getConnection()

cursor 🡨 conn.cursor()

cursor.execute('SELECT name, hours, schedule FROM instructors WHERE id = ?', [self.id])

result 🡨 cursor.fetchone()

conn.close()

self.parent.lineEditName.setText(str(result[0]))

self.parent.lineEditHours.setText(str(result[1]))

# Generate timetable from custom schedule

self.table 🡨 Timetable.Timetable(self.parent.tableSchedule, json.loads(result[2]))

def function 🡨 finish(self)

# Verification of input

IF not self.parent.lineEditName.text():

return 🡨 False

name 🡨 self.parent.lineEditName.text()

TRY

hours 🡨 int(self.parent.lineEditHours.text())

if hours <= 0 or hours > 100 or hours % .5 != 0:

return 🡨 False

EXCEPT

return 🡨 False

data 🡨 [name, hours, json.dumps(self.table.getData()), self.id]

IF not self.id

data.pop()

self.insertInstructor(data)

self.dialog.close()

def function 🡨 insertInstructor(data):

conn 🡨 db.getConnection()

cursor 🡨 conn.cursor()

IF len(data) > 3

cursor.execute('UPDATE instructors SET name = ?, hours = ?, schedule = ? WHERE id=?', data)

ELSE

cursor.execute('INSERT INTO instructors (name, hours, schedule) VALUES (?, ?, ?)', data)

conn.commit()

conn.close()

return True

class 🡨 Tree

def function 🡨 \_\_init\_\_(self, tree)

self.tree 🡨 tree

self.model 🡨 model 🡨 QtGui.QStandardItemModel()

model.setHorizontalHeaderLabels(['ID', 'Available', 'Name', 'Hours', 'Operation'])

tree.setModel(model)

tree.setColumnHidden(0, True)

model.itemChanged.connect(lambda item: self.toggleAvailability(item))

self.display()

def function 🡨 toggleAvailability(self, item)

# Get ID of toggled instructor

id 🡨 self.model.data(self.model.index(item.row(), 0))

newValue 🡨 1 if item.checkState() == 2 else 0

conn 🡨 db.getConnection()

cursor 🡨 conn.cursor()

cursor.execute('UPDATE instructors SET active = ? WHERE id = ?', [newValue, id])

conn.commit()

conn.close()

def function 🡨 display(self)

# Clear model

self.model.removeRows(0, self.model.rowCount())

conn 🡨 db.getConnection()

cursor 🡨 conn.cursor()

cursor.execute('SELECT id, active, hours, name FROM instructors')

result 🡨 cursor.fetchall()

conn.close()

for instr in result:

# ID Item

id 🡨QtGui.QStandardItem(str(instr[0]))

id.setEditable(False)

# Availability Item

availability 🡨 QtGui.QStandardItem()

availability.setCheckable(True)

availability.setCheckState(2 if instr[1] == 1 else 0)

availability.setEditable(False)

# Hours Item

hours 🡨 QtGui.QStandardItem(str(instr[2]))

hours.setEditable(False)

# Name Item

name 🡨 QtGui.QStandardItem(instr[3])

name.setEditable(False)

# Edit Item / Container for operation buttons

edit 🡨 QtGui.QStandardItem()

edit.setEditable(False)

# Append items to model

self.model.appendRow([id, availability, name, hours, edit])

# Create a widget group for edit and delete buttons

frameEdit 🡨 QtWidgets.QFrame()

btnEdit 🡨 QtWidgets.QPushButton('Edit', frameEdit)

btnEdit.clicked.connect(lambda state, id=instr[0]: self.edit(id))

btnDelete 🡨 QtWidgets.QPushButton('Delete', frameEdit)

btnDelete.clicked.connect(lambda state, id=instr[0]: self.delete(id))

frameLayout 🡨 QtWidgets.QHBoxLayout(frameEdit)

frameLayout.setContentsMargins(0, 0, 0, 0)

frameLayout.addWidget(btnEdit)

frameLayout.addWidget(btnDelete)

# Append the widget group to edit item

self.tree.setIndexWidget(edit.index(), frameEdit)

def function 🡨 edit(self, id)

Instructor(id)

self.display()

def function 🡨 delete(self, id)

# Show confirm model

confirm 🡨 QtWidgets.QMessageBox()

confirm.setIcon(QtWidgets.QMessageBox.Warning)

confirm.setText('Are you sure you want to delete this entry?')

confirm.setWindowTitle('Confirm Delete')

confirm.setStandardButtons(QtWidgets.QMessageBox.Yes | QtWidgets.QMessageBox.No)

result 🡨 confirm.exec\_()

# 16384 == Confirm

if result ==16384:

conn 🡨 db.getConnection()

cursor 🡨 conn.cursor()

cursor.execute('DELETE FROM instructors WHERE id = ?', [id])

conn.commit()

conn.close()

self.display()