

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ И. С. ТУРГЕНЕВА»

Кафедра программной инженерии

ОТЧЕТ

по лабораторной работе № 4
на тему: «Разработка графического интерфейса»
по дисциплине: «Программирование на языке Python»

Выполнил: Евдокимов Н.А.

Институт приборостроения, автоматизации и информационных технологий

Направление: 09.03.04 «Программная инженерия»

Группа: 71-ПГ

Проверила: Захарова О.В.

Отметка о зачете:

Дата: « ____ » _____ 2019 г.

Орёл, 2019

Задание

Разработать и реализовать графический интерфейс для задания лабораторной работы № 3 (вариант в соответствии со списком студентов).

Обязательные элементы графического интерфейса: надписи, кнопки, текстовые поля, выпадающий список, чекбоксы и/или радиокнопки, всплывающее окно с сообщением.

Код

```
from PyQt5 import QtWidgets, uic
import sys
from lab3.Book import Book

class UI(QtWidgets.QDialog):
    def __init__(self):
        super(UI, self).__init__()
        uic.loadUi('lab4.ui', self)
        self.show()

book_list = []

def add_book(name, publication_year, n_pages, isbn=None, authors=None, publisher=None, price=None):
    book = Book(name, publication_year, n_pages, isbn, authors, publisher, price)
    book_list.append(book)
    repaint_table()

def btn_delete_clicked(event):
    if len(tvBooks.selectedIndexes()) <= 0:
        return

    index = [idx.row() for idx in tvBooks.selectionModel().selectedRows()][0]
    book_name = book_list[index].name
    remove_book(book_name)

def repaint_table():
    tvBooks.setRowCount(0)
    tvBooks.setColumnCount(7)

    for book in book_list:
        row_pos = tvBooks.rowCount()
        tvBooks.insertRow(row_pos)
        tvBooks.setItem(row_pos, 0, QtWidgets.QTableWidgetItem(book.name))
        tvBooks.setItem(row_pos, 1, QtWidgets.QTableWidgetItem(str(book.publication_year)))
        tvBooks.setItem(row_pos, 2, QtWidgets.QTableWidgetItem(str(book.n_pages)))
        tvBooks.setItem(row_pos, 3, QtWidgets.QTableWidgetItem(book.isbn))
        tvBooks.setItem(row_pos, 4, QtWidgets.QTableWidgetItem(','.join(book.authors)))
        tvBooks.setItem(row_pos, 5, QtWidgets.QTableWidgetItem(book.publisher))
        tvBooks.setItem(row_pos, 6, QtWidgets.QTableWidgetItem(str(book.price)))

def btn_message_clicked(event):
    message_type = str(cbMessageType.currentText())

    is_sad = message_type == 'Грустное сообщение'
    if is_sad:
```

```

        message_str = ':'
    else:
        message_str = ':'

    message = QtWidgets.QMessageBox()
    message.setText(message_str)
    message.setWindowTitle('Сообщение')

    if cbWithIcon.isChecked() and is_sad:
        message.setIcon(QtWidgets.QMessageBox.Critical)
    elif cbWithIcon.isChecked() and not is_sad:
        message.setIcon(QtWidgets.QMessageBox.Information)

    message.exec()

def remove_book(book_name):
    index = -1
    for i, book in enumerate(book_list):
        if book.name == book_name:
            index = i
            break

    if index < 0:
        return

    book_list.pop(index)
    repaint_table()

def btn_add_clicked(event):
    name = leName.text()
    pub_year = sbPublicationYear.value()
    n_pages = sbNPages.value()

    if len(name) <= 0 or n_pages <= 0:
        message = QtWidgets.QMessageBox()
        message.setText('Недостаточно данных о книге. Нужны: название, год издания, кол-во страниц')
        message.setIcon(QtWidgets.QMessageBox.Critical)
        message.setWindowTitle('Книга не добавлена')
        message.exec()

    return

for book in book_list:
    if book.name == name:
        message = QtWidgets.QMessageBox()
        message.setText('Такая книга уже есть')
        message.setIcon(QtWidgets.QMessageBox.Critical)
        message.setWindowTitle('Книга не добавлена')
        message.exec()

    return

isbn = leIsbn.text()
authors = leAuthors.text().replace(' ', '').split(',')
publisher = lePublisher.text()
price = sbPrice.value()
add_book(name, pub_year, n_pages, isbn, authors, publisher, price)

def btn_clear_clicked(event):
    leName.setText("")

```

```

lePublisher.setText("")
leAuthors.setText("")
leIsbn.setText("")
sbPublicationYear.setValue(0)
sbPrice.setValue(0)
sbNPages.setValue(0)

app = QtWidgets.QApplication(sys.argv)
window = UI()

btnClear = window.findChild(QtWidgets.QPushButton, 'btnClear')
btnAdd = window.findChild(QtWidgets.QPushButton, 'btnAdd')
btnDelete = window.findChild(QtWidgets.QPushButton, 'btnDelete')
btnMessage = window.findChild(QtWidgets.QPushButton, 'btnMessage')
btnSearch = window.findChild(QtWidgets.QPushButton, 'btnSearch')

leName = window.findChild(QtWidgets.QLineEdit, 'leName')
leIsbn = window.findChild(QtWidgets.QLineEdit, 'leIsbn')
leAuthors = window.findChild(QtWidgets.QLineEdit, 'leAuthors')
lePublisher = window.findChild(QtWidgets.QLineEdit, 'lePublisher')
leSearch = window.findChild(QtWidgets.QLineEdit, 'leSearch')

sbNPages = window.findChild(QtWidgets.QSpinBox, 'sbNPages')
sbPrice = window.findChild(QtWidgets.QDoubleSpinBox, 'dsbPrice')
sbPublicationYear = window.findChild(QtWidgets.QSpinBox, 'sbPublicationYear')

cbMessageType = window.findChild(QtWidgets.QComboBox, 'cbMessageType')

cbWithIcon = window.findChild(QtWidgets.QCheckBox, 'cbWithIcon')

tvBooks = window.findChild(QtWidgets.QTableView, 'twBooks')
tvBooks.setSelectionBehavior(QtWidgets.QAbstractItemView.SelectRows)

btnAdd.clicked.connect(btn_add_clicked)
btnClear.clicked.connect(btn_clear_clicked)
btnDelete.clicked.connect(btn_delete_clicked)
btnMessage.clicked.connect(btn_message_clicked)

app.exec()

```