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

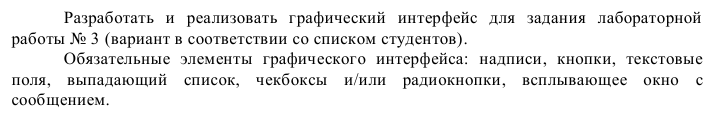
Проверила: Захарова О.В.

Отметка о зачете:

Дата: «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2019 г.

Орёл, 2019

Задание



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()