|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** \_***ИУК «Информатика и управление»*\_\_**\_\_\_\_\_\_\_\_\_\_\_\_

**КАФЕДРА** \_\_***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

**ДОМАШНЯЯ РАБОТА №1**

**«Разработка графического интерфейса с помощью библиотеки PyQt языка Python»**

**ДИСЦИПЛИНА: «Высокоуровневое программирование»**

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК4-32Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Карельский М.К. )  (Подпись) (Ф.И.О.) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Ерохин И.И. )  (Подпись) (Ф.И.О.) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |
| Калуга , 2021 | | |

**Цель:** приобретение практических навыков разработки графического интерфейса с помощью библиотеки PyQt средствами языка Python.

**Задачи:**

1. Ознакомиться с конструкцией библиотеки PyQt;
2. Изучить способы создания мини-приложений с помощью PyQt;
3. Изучить возможности данной платформы;
4. Закрепить полученные в ходе выполнения домашней работы навыки.

**Вариант 10**

**Задание:**

*Задача 1.*

В созданном окне вам необходимо построить график функций на осях x и y по заданной формуле:

*Задача 2.*

Используйте предыдущее окно с нарисованным графиком, вам необходимо:

1. Поставить надпись «График (формула) был построен» напротив графика.
2. Создать Toggle button, содержащий варианты выбора цвета светофора. При выборе рядом стоящий квадрат должен появляться окрашенным в выбранный цвет. Создайте цветное текстовое сообщение «Cветофор».
3. Создать Button с названием «Справка», при наведении на которую всплывает подсказка, из которой можно узнать об истории появления светофора. При нажатии на кнопку должно появляться сообщение «Обращение за справкой».

**Листинг:**

*Interface.py:*

from PyQt5 import QtCore, QtGui, QtWidgets

class Ui\_MainWindow(object):

def setupUi(self, MainWindow):

MainWindow.setObjectName("MainWindow")

MainWindow.setFixedSize(264, 313)

self.centralwidget = QtWidgets.QWidget(MainWindow)

self.centralwidget.setObjectName("centralwidget")

self.GraphButton = QtWidgets.QPushButton(self.centralwidget)

self.GraphButton.setGeometry(QtCore.QRect(20, 260, 121, 28))

self.GraphButton.setToolTip("")

self.GraphButton.setObjectName("GraphButton")

self.RedButton = QtWidgets.QRadioButton(self.centralwidget)

self.RedButton.setGeometry(QtCore.QRect(40, 80, 95, 20))

font = QtGui.QFont()

font.setPointSize(10)

self.RedButton.setFont(font)

self.RedButton.setObjectName("RedButton")

self.YellowButton = QtWidgets.QRadioButton(self.centralwidget)

self.YellowButton.setGeometry(QtCore.QRect(40, 140, 95, 20))

font = QtGui.QFont()

font.setPointSize(10)

self.YellowButton.setFont(font)

self.YellowButton.setObjectName("YellowButton")

self.GreenButton = QtWidgets.QRadioButton(self.centralwidget)

self.GreenButton.setGeometry(QtCore.QRect(40, 200, 95, 20))

font = QtGui.QFont()

font.setPointSize(10)

self.GreenButton.setFont(font)

self.GreenButton.setObjectName("GreenButton")

self.TrafficLightsLabel = QtWidgets.QLabel(self.centralwidget)

self.TrafficLightsLabel.setGeometry(QtCore.QRect(50, 10, 151, 31))

font = QtGui.QFont()

font.setPointSize(16)

self.TrafficLightsLabel.setFont(font)

self.TrafficLightsLabel.setAlignment(QtCore.Qt.AlignCenter)

self.TrafficLightsLabel.setObjectName("TrafficLightsLabel")

self.HelpButton = QtWidgets.QPushButton(self.centralwidget)

self.HelpButton.setGeometry(QtCore.QRect(150, 260, 93, 28))

self.HelpButton.setObjectName("HelpButton")

self.HelpButton.setToolTip(

"Первый светофор был установлен 10 декабря 1868 года в Лондоне возле \n\

здания Британского парламента. Его изобретатель — Джон Пик Найт — был \n\

специалистом по железнодорожным семафорам.")

MainWindow.setCentralWidget(self.centralwidget)

self.menubar = QtWidgets.QMenuBar(MainWindow)

self.menubar.setGeometry(QtCore.QRect(0, 0, 264, 26))

self.menubar.setObjectName("menubar")

MainWindow.setMenuBar(self.menubar)

self.statusbar = QtWidgets.QStatusBar(MainWindow)

self.statusbar.setObjectName("statusbar")

MainWindow.setStatusBar(self.statusbar)

self.retranslateUi(MainWindow)

QtCore.QMetaObject.connectSlotsByName(MainWindow)

def retranslateUi(self, MainWindow):

\_translate = QtCore.QCoreApplication.translate

MainWindow.setWindowTitle(\_translate("MainWindow", "MainWindow"))

self.GraphButton.setText(\_translate("MainWindow", "Построить график"))

self.RedButton.setText(\_translate("MainWindow", "Красный"))

self.YellowButton.setText(\_translate("MainWindow", "Желтый"))

self.GreenButton.setText(\_translate("MainWindow", "Зеленый"))

self.TrafficLightsLabel.setText(\_translate("MainWindow", "<span style=\" color:#0055ff;\">Светофор</span>"))

self.HelpButton.setText(\_translate("MainWindow", "Справка"))

*HW 1.py:*

from PyQt5 import QtCore, QtGui, QtWidgets

from PyQt5.QtGui import QPainter, QColor, QBrush

import sys

from Interface import \*

import matplotlib.pyplot as plt

import math

class MyWin(QtWidgets.QMainWindow):

def \_\_init\_\_(self, parent=None):

QtWidgets.QWidget.\_\_init\_\_(self, parent)

self.activatedLight = 0

self.ui = Ui\_MainWindow()

self.ui.setupUi(self)

self.ui.GraphButton.clicked.connect(self.PressGraph)

self.ui.RedButton.toggled.connect(self.PressRed)

self.ui.YellowButton.toggled.connect(self.PressYellow)

self.ui.GreenButton.toggled.connect(self.PressGreen)

self.ui.HelpButton.clicked.connect(self.PressHelp)

def paintEvent(self, e):

qp = QPainter()

qp.begin(self)

self.drawRectangles(qp)

qp.end()

def drawRectangles(self, qp):

col = QColor(0, 0, 0)

col.setNamedColor('#d4d4d4')

qp.setPen(col)

if self.activatedLight == 0:

qp.setBrush(QColor(255, 255, 255))

qp.drawRect(160, 60, 60, 60)

qp.setBrush(QColor(255, 255, 255))

qp.drawRect(160, 120, 60, 60)

qp.setBrush(QColor(255, 255, 255))

qp.drawRect(160, 180, 60, 60)

elif self.activatedLight == 1:

qp.setBrush(QColor(230, 0, 0))

qp.drawRect(160, 60, 60, 60)

qp.setBrush(QColor(255, 255, 255))

qp.drawRect(160, 120, 60, 60)

qp.setBrush(QColor(255, 255, 255))

qp.drawRect(160, 180, 60, 60)

elif self.activatedLight == 2:

qp.setBrush(QColor(255, 255, 255))

qp.drawRect(160, 60, 60, 60)

qp.setBrush(QColor(247, 231, 0))

qp.drawRect(160, 120, 60, 60)

qp.setBrush(QColor(255, 255, 255))

qp.drawRect(160, 180, 60, 60)

elif self.activatedLight == 3:

qp.setBrush(QColor(255, 255, 255))

qp.drawRect(160, 60, 60, 60)

qp.setBrush(QColor(255, 255, 255))

qp.drawRect(160, 120, 60, 60)

qp.setBrush(QColor(0, 171, 17))

qp.drawRect(160, 180, 60, 60)

def PressGraph(self):

x1 = [n/10 for n in range(-40, 0)]

x2 = [n/10 for n in range(1, 41)]

y1 = [abs(math.cos(2\*n)\*math.sin(3\*n))/math.sin(n) for n in x1]

y2 = [abs(math.cos(2\*n)\*math.sin(3\*n))/math.sin(n) for n in x2]

ax = [0, 0]

ay = [-4, 4]

plt.text(0.5, 3.5, "y = |cos(2x)sin(2x)|/sin(x)")

plt.plot(x1, y1, x2, y2, ax, ay, '--')

plt.show()

def PressRed(self):

button = self.sender()

if button.isChecked():

self.activatedLight = 1

self.update()

def PressYellow(self):

button = self.sender()

if button.isChecked():

self.activatedLight = 2

self.update()

def PressGreen(self):

button = self.sender()

if button.isChecked():

self.activatedLight = 3

self.update()

def PressHelp(self):

message = QtWidgets.QMessageBox()

message.setIcon(QtWidgets.QMessageBox.Information)

message.setText("Обращение за справкой")

message.setWindowTitle("Справка")

message.exec\_()

if \_\_name\_\_=="\_\_main\_\_":

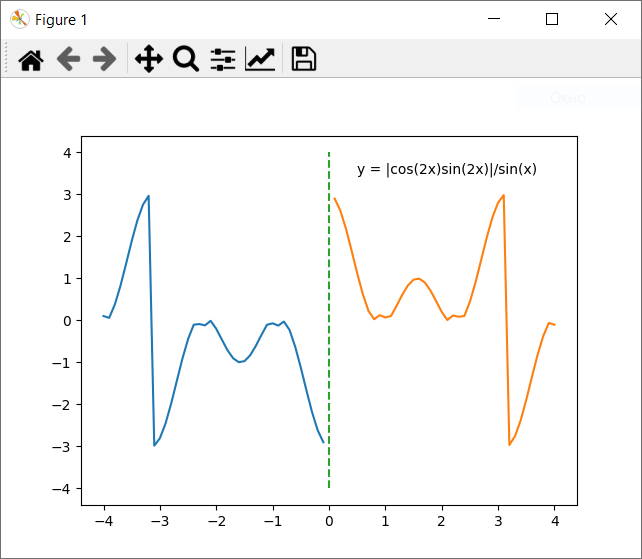
app = QtWidgets.QApplication(sys.argv)

myapp = MyWin()

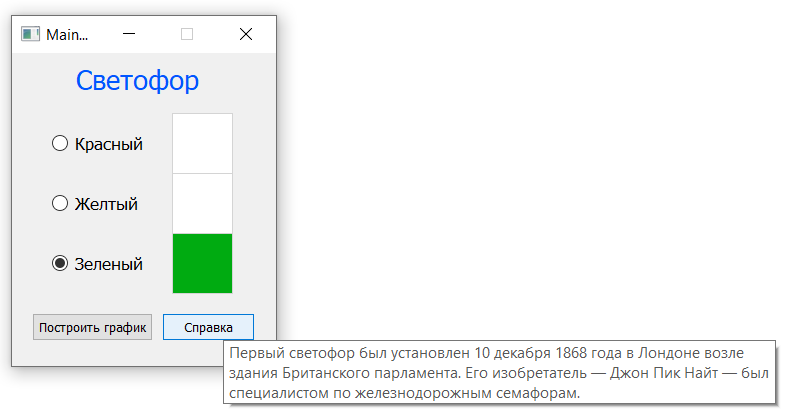
myapp.show()

sys.exit(app.exec\_())

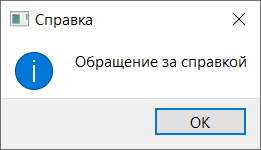
**Демонстрация:**



**Рисунок 1.** Окно с графиком



**Рисунок 2.** Главное окно



**Рисунок 3.** Окно справки

**Вывод:** в ходе выполнения лабораторной работы были получены навыки работы с PyQt Designer, библиотеками PyQt5, matplotlib, создания оконного приложения.