

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Отчет по лабораторной работе №2

Основы работы с PySide2

По дисциплине «Объектно-ориентированное программирование»

Выполнил студент группы ИВТ-б-о-20-1

Новикова В.И. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р. А. _____

(подпись)

Ставрополь 2022

Цель работы: приобретение навыков построения графического интерфейса пользователя GUI с помощью пакета PySide2 языка программирования Python версии 3.x

Ход работы:

1. Создала общедоступный репозиторий и клонировала на его на локальный сервер.

https://github.com/Valentina1502/4.7_Pyside.git

2. После изучения теоретического материала и методических рекомендаций приступила к выполнению заданий.

Задание 1. Напишите простейший калькулятор, состоящий из двух текстовых полей, куда пользователь вводит числа, и четырех кнопок "+", "-", "*", "/". Результат вычисления должен отображаться в метке. Если арифметическое действие выполнить невозможно (например, если были введены буквы, а не числа), то в метке должно появляться слово "ошибка"

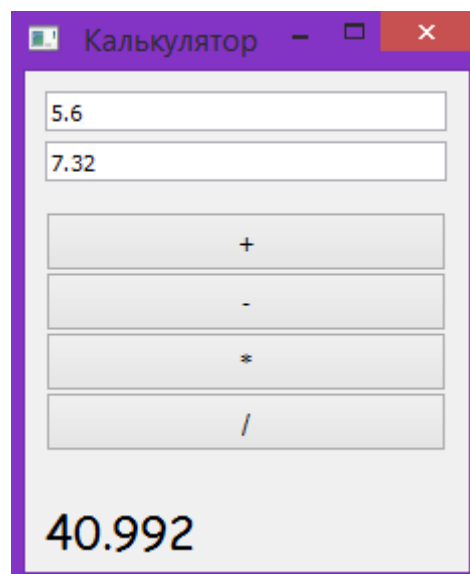


Рисунок 1 – Задание 1

Код:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

from PySide2.QtWidgets import QApplication, QLabel, \
    QWidget, QLineEdit, QPushButton
import sys
from PySide2.QtGui import QFont
```

```

class MainWindow(QWidget):
    def __init__(self):
        super().__init__() # вызываем конструктор базового класса
        self.line_edit1 = QLineEdit(self)
        self.line_edit2 = QLineEdit(self)
        self.button1 = QPushButton('+', self)
        self.button2 = QPushButton('-', self)
        self.button3 = QPushButton('*', self)
        self.button4 = QPushButton('/', self)
        self.label = QLabel(self)
        self.initializeUI()

    def initializeUI(self):
        self.setGeometry(400, 200, 250, 250)
        self.setWindowTitle("Калькулятор")

        self.line_edit1.move(10, 10)
        self.line_edit1.resize(200, 20)
        self.line_edit2.move(10, 35)
        self.line_edit2.resize(200, 20)
        self.button1.move(10, 70)
        self.button1.resize(200, 30)
        self.button2.move(10, 100)
        self.button2.resize(200, 30)
        self.button3.move(10, 130)
        self.button3.resize(200, 30)
        self.button4.move(10, 160)
        self.button4.resize(200, 30)
        self.answer_label = QLabel(self)
        self.answer_label.move(10, 215)
        self.answer_label.resize(200, 30)
        self.answer_label.setFont(QFont('Comic Sans MS', 17))

        self.button1.clicked.connect(self.Calculator)
        self.button2.clicked.connect(self.Calculator)
        self.button3.clicked.connect(self.Calculator)
        self.button4.clicked.connect(self.Calculator)
        self.show()

    def Calculator(self):
        try:
            if float(self.line_edit1.text()) and
float(self.line_edit2.text()):
                n1 = float(self.line_edit1.text())
                n2 = float(self.line_edit2.text())
            except ValueError:
                self.answer_label.setText("Введите числа")

        sender = self.sender()
        if sender.text() == "+":
            self.answer_label.setText(str(n1 + n2))
        elif sender.text() == "-":
            self.answer_label.setText(str(n1 - n2))
        elif sender.text() == "*":
            self.answer_label.setText(str(n1 * n2))
        elif sender.text() == "/":
            self.answer_label.setText(str(n1 / n2))

if __name__ == '__main__':
    app = QApplication(sys.argv)

```

```
window = MainWindow()
sys.exit(app.exec_()) #зацикливание
```

Задание 2. Напишите программу, состоящую из семи кнопок, цвета которых соответствуют цветам радуги. При нажатии на ту или иную кнопку в текстовое поле должен вставляться код цвета, а в метку – название цвета.

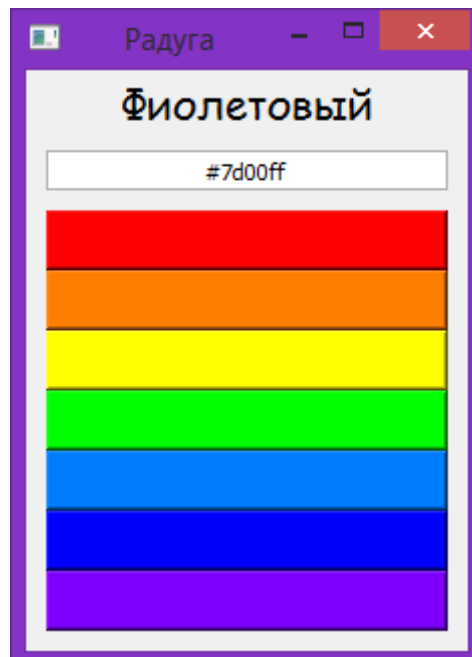


Рисунок 2 – Задание 2

Код:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

from PySide2.QtWidgets import QApplication, QLabel, \
    QWidget, QLineEdit, QPushButton
import sys
from PySide2.QtGui import QFont
from PySide2.QtCore import Qt

class MainWindow(QWidget):
    def __init__(self):
        super().__init__() # вызываем конструктор базового класса
        self.label = QLabel(self)
        self.line_edit = QLineEdit(self)
        self.button1 = QPushButton(self)
        self.button2 = QPushButton(self)
        self.button3 = QPushButton(self)
        self.button4 = QPushButton(self)
        self.button5 = QPushButton(self)
        self.button6 = QPushButton(self)
        self.button7 = QPushButton(self)
```

```

self.initializeUI()

def initializeUI(self):
    self.setGeometry(200, 200, 220, 290)
    self.setWindowTitle("Радуга")

    self.label = QLabel(self)
    self.label.move(10, 1)
    self.label.resize(200, 30)
    self.label.setFont(QFont('Comic Sans MS', 16))
    self.label.setAlignment(Qt.AlignCenter)
    self.line_edit.move(10, 40)
    self.line_edit.setAlignment(Qt.AlignCenter)
    self.line_edit.resize(200, 20)
    self.button1.move(10, 70)
    self.button1.resize(200, 30)
    self.button1.setStyleSheet("background: #ff0000")
    self.button2.move(10, 100)
    self.button2.resize(200, 30)
    self.button2.setStyleSheet("background: #ff7d00")
    self.button3.move(10, 130)
    self.button3.resize(200, 30)
    self.button3.setStyleSheet("background: #ffff00")
    self.button4.move(10, 160)
    self.button4.resize(200, 30)
    self.button4.setStyleSheet("background: #00ff00")
    self.button5.move(10, 190)
    self.button5.resize(200, 30)
    self.button5.setStyleSheet("background: #007dff")
    self.button6.move(10, 220)
    self.button6.resize(200, 30)
    self.button6.setStyleSheet("background: #0000ff")
    self.button7.move(10, 250)
    self.button7.resize(200, 30)
    self.button7.setStyleSheet("background: #7d00ff")

    self.button1.clicked.connect(self.Red)
    self.button2.clicked.connect(self.Orange)
    self.button3.clicked.connect(self.Yellow)
    self.button4.clicked.connect(self.Green)
    self.button5.clicked.connect(self.LBlue)
    self.button6.clicked.connect(self.Blue)
    self.button7.clicked.connect(self.Purple)
    self.show()

def Red(self):
    self.label.setText("Красный")
    self.line_edit.setText("#ff0000")

def Orange(self):
    self.label.setText("Оранжевый")
    self.line_edit.setText("#ff7d00")

def Yellow(self):
    self.label.setText("Желтый")
    self.line_edit.setText("#ffff00")

def Green(self):
    self.label.setText("Зеленый")
    self.line_edit.setText("#00ff00")

```

```

def LBlue(self):
    self.label.setText("Голубой")
    self.line_edit.setText("#007dff")

def Blue(self):
    self.label.setText("Синий")
    self.line_edit.setText("#0000ff")

def Purple(self):
    self.label.setText("Фиолетовый")
    self.line_edit.setText("#7d00ff")

if __name__ == '__main__':
    app = QApplication(sys.argv)
    window = MainWindow()
    sys.exit(app.exec_())

```

Задание 3. Перепишите программу

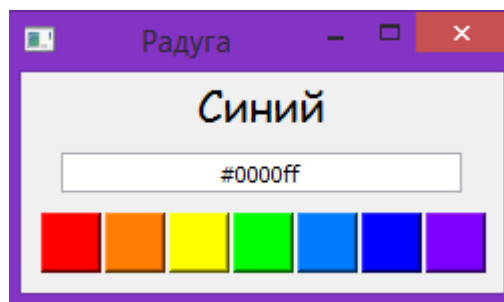


Рисунок 3 – Задание 3

Код:

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

from PySide2.QtWidgets import QApplication, QLabel, \
    QWidget, QLineEdit, QPushButton
import sys
from PySide2.QtGui import QFont
from PySide2.QtCore import Qt

class MainWindow(QWidget):
    def __init__(self):
        super().__init__() # вызываем конструктор базового класса
        self.label = QLabel(self)
        self.line_edit = QLineEdit(self)
        self.button1 = QPushButton(self)
        self.button2 = QPushButton(self)
        self.button3 = QPushButton(self)
        self.button4 = QPushButton(self)
        self.button5 = QPushButton(self)
        self.button6 = QPushButton(self)
        self.button7 = QPushButton(self)

```

```

self.initializeUI()

def initializeUI(self):
    self.setGeometry(200, 200, 240, 110)
    self.setWindowTitle("Радуга")

    self.label = QLabel(self)
    self.label.move(20, 1)
    self.label.resize(200, 30)
    self.label.setFont(QFont('Comic Sans MS', 16))
    self.label.setAlignment(Qt.AlignCenter)
    self.line_edit.move(20, 40)
    self.line_edit.setAlignment(Qt.AlignCenter)
    self.line_edit.resize(200, 20)
    self.button1.move(10, 70)
    self.button1.resize(30, 30)
    self.button1.setStyleSheet("background: #ff0000")
    self.button2.move(42, 70)
    self.button2.resize(30, 30)
    self.button2.setStyleSheet("background: #ff7d00")
    self.button3.move(74, 70)
    self.button3.resize(30, 30)
    self.button3.setStyleSheet("background: #ffff00")
    self.button4.move(106, 70)
    self.button4.resize(30, 30)
    self.button4.setStyleSheet("background: #00ff00")
    self.button5.move(138, 70)
    self.button5.resize(30, 30)
    self.button5.setStyleSheet("background: #007dff")
    self.button6.move(170, 70)
    self.button6.resize(30, 30)
    self.button6.setStyleSheet("background: #0000ff")
    self.button7.move(202, 70)
    self.button7.resize(30, 30)
    self.button7.setStyleSheet("background: #7d00ff")

    self.button1.clicked.connect(self.Red)
    self.button2.clicked.connect(self.Orange)
    self.button3.clicked.connect(self.Yellow)
    self.button4.clicked.connect(self.Green)
    self.button5.clicked.connect(self.LBlue)
    self.button6.clicked.connect(self.Blue)
    self.button7.clicked.connect(self.Purple)
    self.show()

def Red(self):
    self.label.setText("Красный")
    self.line_edit.setText("#ff0000")

def Orange(self):
    self.label.setText("Оранжевый")
    self.line_edit.setText("#ff7d00")

def Yellow(self):
    self.label.setText("Желтый")
    self.line_edit.setText("#ffff00")

def Green(self):
    self.label.setText("Зеленый")
    self.line_edit.setText("#00ff00")

```

```

def LBlue(self):
    self.label.setText("Голубой")
    self.line_edit.setText("#007dff")

def Blue(self):
    self.label.setText("Синий")
    self.line_edit.setText("#0000ff")

def Purple(self):
    self.label.setText("Фиолетовый")
    self.line_edit.setText("#7d00ff")

if __name__ == '__main__':
    app = QApplication(sys.argv)
    window = MainWindow()
    sys.exit(app.exec_())

```

Задание 4. Напишите программу, состоящую из однострочного и многострочного текстовых полей и двух кнопок "Открыть" и "Сохранить". При клике на первую должен открываться на чтение файл, чье имя указано в поле класса Entry, а содержимое файла должно загружаться в поле типа Text. При клике на вторую кнопку текст, введенный пользователем в экземпляр Text должен сохраняться в файле под именем, которое пользователь указал в однострочном текстовом поле

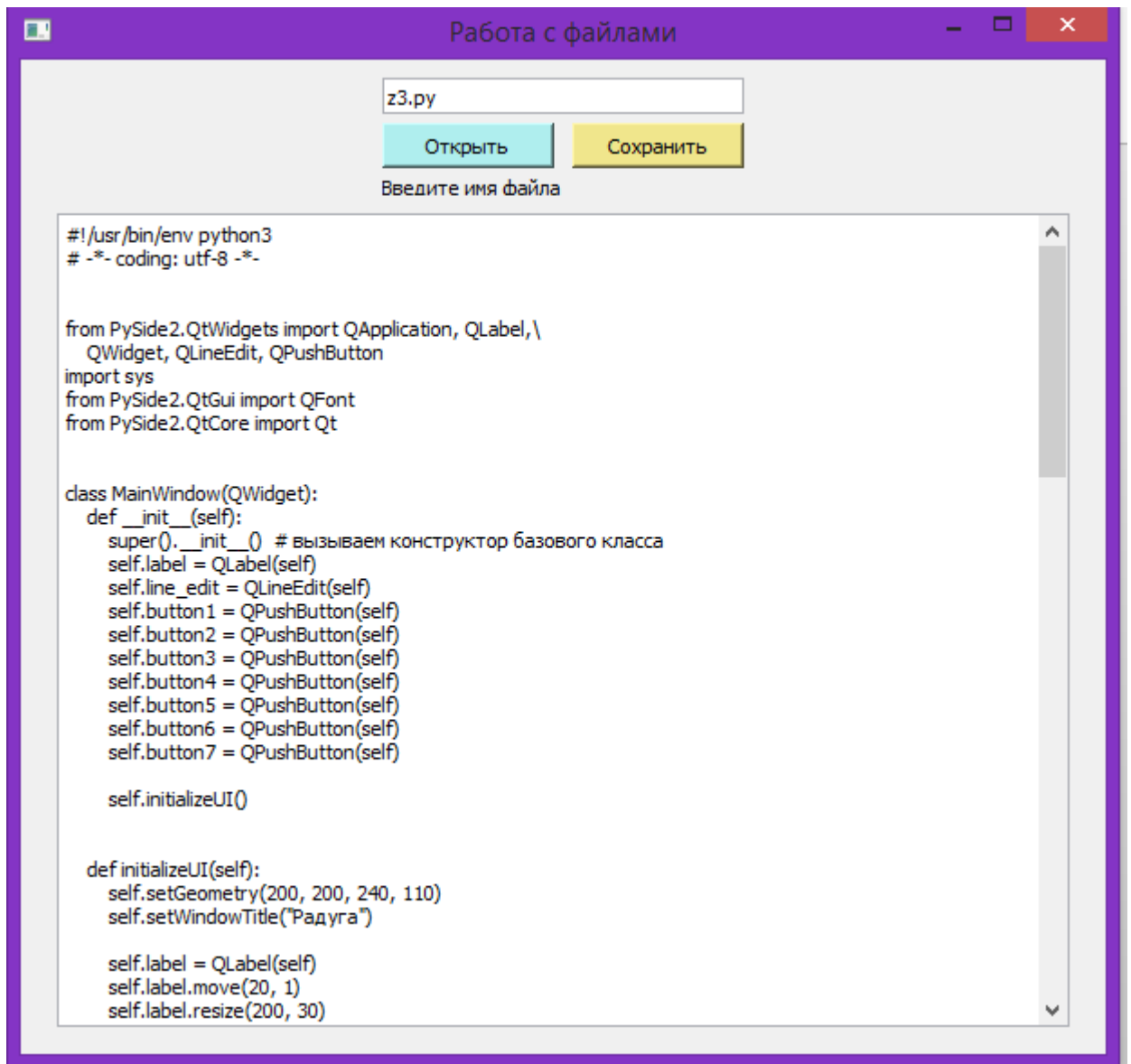


Рисунок 4 – Задание 4

Код:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

from PySide2.QtWidgets import QApplication, QLabel,\
    QWidget, QLineEdit, QPushButton, QFileDialog, QTextEdit
from PySide2.QtCore import QByteArray
import sys

class MainWindow(QWidget):
    def __init__(self):
        super().__init__() # вызываем конструктор базового класса
        self.line_edit = QLineEdit(self)
        self.button1 = QPushButton("Открыть", self)
        self.button2 = QPushButton("Сохранить", self)
        self.label = QLabel(self)
        self.text = QTextEdit(self)
        self.initializeUI()
```

```

def initializeUI(self):
    self.setGeometry(50, 100, 600, 550)
    self.setWindowTitle("Работа с файлами")

    self.line_edit.move(200, 10)
    self.line_edit.resize(200, 20)
    self.button1.move(200, 35)
    self.button1.resize(95, 25)
    self.button1.setStyleSheet("background: #AFEEEE")
    self.button2.move(305, 35)
    self.button2.resize(95, 25)
    self.button2.setStyleSheet("background: #F0E68C")
    self.label.move(200, 65)
    self.label.resize(200, 10)
    self.text.move(20, 85)
    self.text.resize(560, 450)

    self.button1.clicked.connect(self.OpenF)
    self.button2.clicked.connect(self.SaveF)
    self.show()

def OpenF(self):
    if self.line_edit.text() != '':
        try:
            file_name = self.line_edit.text()
            with open(file_name, 'r', encoding="utf-8") as f:
                file = f.read()
            self.text.setText(file)
        except Exception as excep:
            self.label.setText(f"Ошибка. {excep}")
    else:
        self.label.setText(f"Введите имя файла")

def SaveF(self):
    if self.line_edit.text() != '':
        file = self.text.toPlainText()
        file = bytes(file, encoding='utf-8')
        file = QByteArray(file)
        file_name = self.line_edit.text()
        QFileDialog.saveFileContent(file, file_name)
    else:
        self.label.setText("Введите имя файла")

if __name__ == '__main__':
    app = QApplication(sys.argv)
    window = MainWindow()
    sys.exit(app.exec_())

```

Задание 5. Напишите программу, в которой имеется несколько объединенных в группу радиокнопок, индикатор которых выключен (indicatoron=0). Если какая-нибудь кнопка включается, то в метке должна отображаться соответствующая ей информация.

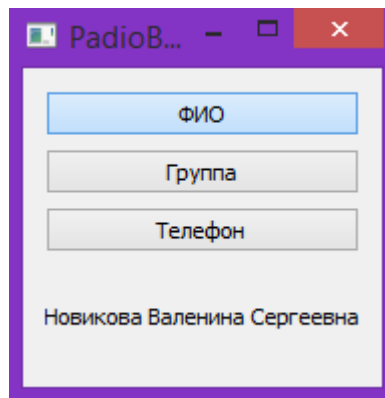


Рисунок 5 – Задание 5

Код:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys
from PySide2.QtWidgets import QWidget, QApplication, QMainWindow, \
    QPushButton, QButtonGroup, QLabel, QHBoxLayout, QVBoxLayout

class MainWindow(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("PadioButtons")
        self.setGeometry(100, 100, 280, 100)

        self.label = QLabel(self)
        self.label.move(50, 50)
        self.label.resize(100, 20)
        self.RadioButton1 = QPushButton('ФИО', self)
        self.RadioButton1.setCheckable(True)
        self.RadioButton1.move(10, 10)
        self.RadioButton1.resize(80, 20)
        self.RadioButton2 = QPushButton('Группа', self)
        self.RadioButton2.setCheckable(True)
        self.RadioButton2.move(100, 10)
        self.RadioButton2.resize(80, 20)
        self.RadioButton3 = QPushButton('Телефон', self)
        self.RadioButton3.setCheckable(True)
        self.RadioButton3.move(190, 10)
        self.RadioButton3.resize(80, 20)
        self.buttonGr = QButtonGroup()
        self.buttonGr.addButton(self.RadioButton1)
        self.buttonGr.addButton(self.RadioButton2)
        self.buttonGr.addButton(self.RadioButton3)
        self.buttonGr.buttonClicked.connect(self.RadioButtonClicked)
        Layout = QVBoxLayout(self)
        Layout.addWidget(self.RadioButton1)
        Layout.addWidget(self.RadioButton2)
        Layout.addWidget(self.RadioButton3)
        Layout.addWidget(self.label)
        self.show()

    def RadioButtonClicked(self, Rb):
        info = {
            'ФИО': 'Новикова Валенина Сергеевна',
            'Группа': 'ИВТ-6-о-20-1',
```

```
        'Телефон': '8-(988)-755-11-15'  
    }  
    self.label.setText(info[Rb.text()])  
  
if __name__ == "__main__":  
    app = QApplication(sys.argv)  
    window = MainWindow()  
    sys.exit(app.exec_())
```

Вывод: в ходе выполнения лабораторной работы были приобретены навыки построения графического интерфейса пользователя GUI с помощью пакета PySide2 языка программирования Python версии 3.x.