

**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»**

Выполнила студентка группы ИВТ-б-о-20-1

Хацукова А. 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.
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"

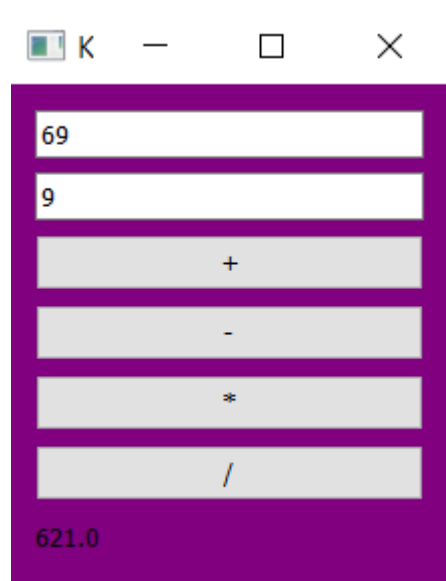


Рисунок 1 – Задание 1

Код:

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

from PySide2.QtWidgets import QApplication, QLabel, QWidget, QLineEdit, QPushButton, QVBoxLayout
import sys

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, 220, 250)
    self.setWindowTitle("Калькулятор")
    vbox = QVBoxLayout()
    vbox.addWidget(self.line_edit1)
    vbox.addWidget(self.line_edit2)
    vbox.addWidget(self.button1)
    vbox.addWidget(self.button2)
    vbox.addWidget(self.button3)
    vbox.addWidget(self.button4)
    vbox.addWidget(self.label)
    self.setLayout(vbox)

    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.label.setText("Введите числа")

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

if __name__ == '__main__':
    app = QApplication(sys.argv)
    window = MainWindow()
    window.setObjectName("MainWindow")

```

```
window.setStyleSheet("#MainWindow{background-color:purple}")
sys.exit(app.exec_())
```

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.

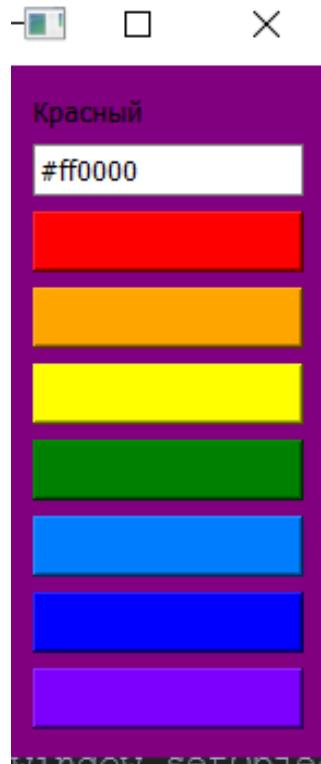


Рисунок 2 – Задание 2

Код:

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

import sys
from PySide2.QtWidgets import QWidget, QApplication, QPushButton,
QHBoxLayout, QVBoxLayout, QLabel, QLineEdit
from PySide2.QtCore import Qt

"""
Перепишите программу из задания 3 так,
чтобы интерфейс выглядел иначе.
"""

class MainWindow(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Rainbow 3")
        self.setGeometry(230, 110, 230, 110)
        self.line_edit = QLineEdit(self)
        self.label = QLabel(self)
        self.label.setAlignment(Qt.AlignHCenter)
        self.button1 = QPushButton()
```

```
self.button2 = QPushButton()
self.button3 = QPushButton()
self.button4 = QPushButton()
self.button5 = QPushButton()
self.button6 = QPushButton()
self.button7 = QPushButton()

self.btn_click()

def btn_click(self):
    self.button1.setStyleSheet("background-color: #ff0000;")
    self.button1.clicked.connect(self.red)
    self.button2.setStyleSheet("background-color: #ff7d00;")
    self.button2.clicked.connect(self.orange)
    self.button3.setStyleSheet("background-color: #ffff00;")
    self.button3.clicked.connect(self.yellow)
    self.button4.setStyleSheet("background-color: #00ff00;")
    self.button4.clicked.connect(self.green)
    self.button5.setStyleSheet("background-color: #007dff;")
    self.button5.clicked.connect(self.cuan)
    self.button6.setStyleSheet("background-color: #0000ff;")
    self.button6.clicked.connect(self.blue)
    self.button7.setStyleSheet("background-color: #7d00ff;")
    self.button7.clicked.connect(self.purple)

def widgets(self):
    vbox = QVBoxLayout()
    layout = QHBoxLayout()
    vbox.addWidget(self.label)
    vbox.addWidget(self.line_edit)
    layout.addWidget(self.button1)
    layout.addWidget(self.button2)
    layout.addWidget(self.button3)
    layout.addWidget(self.button4)
    layout.addWidget(self.button5)
    layout.addWidget(self.button6)
    layout.addWidget(self.button7)
    vbox.addLayout(layout)
    self.setLayout(vbox)

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 cuan(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()
    window.widgets()
    window.show()
    window.setObjectName("MainWindow")
    window.setStyleSheet("#MainWindow{background-color:purple}")
    sys.exit(app.exec_())

```

Задание 3. Перепишите программу



Рисунок 3 – Задание 3

Код:

```

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

import sys
from PySide2.QtWidgets import QWidget, QApplication, QPushButton,
QHBoxLayout, QVBoxLayout, QLabel, QLineEdit
from PySide2.QtCore import Qt

"""
Перепишите программу из задания 3 так,
чтобы интерфейс выглядел иначе.
"""

class MainWindow(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Rainbow 3")
        self.setGeometry(230, 110, 230, 110)
        self.line_edit = QLineEdit(self)
        self.label = QLabel(self)
        self.label.setAlignment(Qt.AlignHCenter)
        self.button1 = QPushButton()
        self.button2 = QPushButton()
        self.button3 = QPushButton()
        self.button4 = QPushButton()
        self.button5 = QPushButton()
        self.button6 = QPushButton()
        self.button7 = QPushButton()

        self.btn_click()

```

```

def btn_click(self):
    self.button1.setStyleSheet("background-color: #ff0000;")
    self.button1.clicked.connect(self.red)
    self.button2.setStyleSheet("background-color: #ff7d00;")
    self.button2.clicked.connect(self.orange)
    self.button3.setStyleSheet("background-color: #ffff00;")
    self.button3.clicked.connect(self.yellow)
    self.button4.setStyleSheet("background-color: #00ff00;")
    self.button4.clicked.connect(self.green)
    self.button5.setStyleSheet("background-color: #007dff;")
    self.button5.clicked.connect(self.cuan)
    self.button6.setStyleSheet("background-color: #0000ff;")
    self.button6.clicked.connect(self.blue)
    self.button7.setStyleSheet("background-color: #7d00ff;")
    self.button7.clicked.connect(self.purple)

def widgets(self):
    vbox = QVBoxLayout()
    layout = QHBoxLayout()
    vbox.addWidget(self.label)
    vbox.addWidget(self.line_edit)
    layout.addWidget(self.button1)
    layout.addWidget(self.button2)
    layout.addWidget(self.button3)
    layout.addWidget(self.button4)
    layout.addWidget(self.button5)
    layout.addWidget(self.button6)
    layout.addWidget(self.button7)
    vbox.addLayout(layout)
    self.setLayout(vbox)

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 cuan(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()
    window.widgets()

```

```

window.show()
window.setObjectName("MainWindow")
window.setStyleSheet("#MainWindow{background-color:purple}")
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 -*-

import sys
from PySide2 import QtWidgets
from PySide2.QtWidgets import QFileDialog, QWidget, QVBoxLayout, QTextEdit,
QHBoxLayout, QPushButton, QApplication

class MainWindow(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Task 4")
        self.setGeometry(100, 100, 100, 100)
        self.text = QTextEdit()
        self.initializeUI()

    def initializeUI(self):
        self.label.resize(80, 10)
        self.text_edit.resize(60, 20)

        self.button1.setStyleSheet("background-color: red;")
        self.button1.clicked.connect(self.red)

        self.button2.setStyleSheet("background-color: green;")

```



```

def initializeUI(self):
    vbox = QVBoxLayout()
    grid = QHBoxLayout()
    vbox.addLayout(grid)
    btn1 = QPushButton("Save")
    btn1.clicked.connect(self.save)
    btn2 = QPushButton("Open")
    btn2.clicked.connect(self.open)
    grid.addWidget(btn1)
    grid.addWidget(btn2)
    vbox.addWidget(self.text)
    self.setLayout(vbox)

def save(self):
    filename, _ = QFileDialog.getSaveFileName(
        self,
        'Save File As',
        '',
        "Text Files (*.txt)"
    )
    if filename:
        text = self.text.toPlainText()
        with open(filename, 'w', encoding="utf-8") as f:
            f.write(text)

def open(self):
    self.text.clear()
    name = QFileDialog.getOpenFileName()
    with open(name[0], 'r', encoding="utf-8") as f:
        data = f.read()
    self.text.setText(str(data))

if __name__ == "__main__":
    app = QApplication(sys.argv)
    main_window = MainWindow()
    main_window.show()
    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.

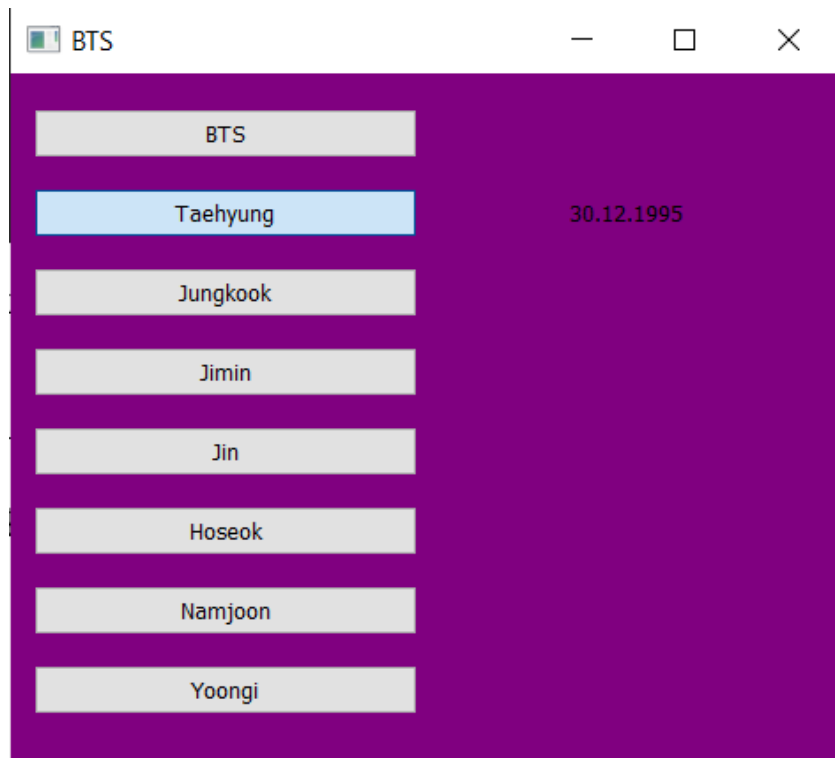


Рисунок 5 – Задание 5

Код:

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

import sys
from PySide2.QtWidgets import QWidget, QApplication, QPushButton,
QButtonGroup, QGridLayout, QLabel
from PySide2.QtCore import Qt

"""
Напишите программу, в которой имеется несколько объединенных в группу
радиокнопок. Если какая-нибудь кнопка включается,
то в метке должна отображаться соответствующая ей информация. Обычных кнопок
в окне быть не должно.
"""

members = {
    'BTS': 'also known as the Bangtan Boys,\n '
           'is a South Korean boy band formed in\n '
           '2010 and debuting in 2013 under Big Hit Entertainment.\n '
           'Consisting of members: Kim Namjoon, Kim Seokjin, Min Yoongi,\n '
           'Jeong Hoseok,\n '
           'Park Jimin, Kim Taehyung, Jeon Jungkook',
    'Taehyung': '30.12.1995',
    'Jungkook': '1.09.1997',
    'Jimin': '12.10.1995',
    'Jin': '4.12.1992',
    'Hoseok': '12.02.1994',
    'Namjoon': '12.09.1994',
    'Yoongi': '9.03.1993'
}
```

```

class MainWindow(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("BTS")
        self.setGeometry(470, 390, 470, 390)
        self.label = QLabel()
        self.label.setAlignment(Qt.AlignCenter)
        self.radio_button_1 = QPushButton('BTS')
        self.radio_button_1.setCheckable(True)
        self.radio_button_2 = QPushButton('Taehyung')
        self.radio_button_2.setCheckable(True)
        self.radio_button_3 = QPushButton('Jungkook')
        self.radio_button_3.setCheckable(True)
        self.radio_button_4 = QPushButton('Jimin')
        self.radio_button_4.setCheckable(True)
        self.radio_button_5 = QPushButton('Jin')
        self.radio_button_5.setCheckable(True)
        self.radio_button_6 = QPushButton('Hoseok')
        self.radio_button_6.setCheckable(True)
        self.radio_button_7 = QPushButton('Namjoon')
        self.radio_button_7.setCheckable(True)
        self.radio_button_8 = QPushButton('Yoongi')
        self.radio_button_8.setCheckable(True)

        self.button_group = QButtonGroup()
        self.button_group.addButton(self.radio_button_1)
        self.button_group.addButton(self.radio_button_2)
        self.button_group.addButton(self.radio_button_3)
        self.button_group.addButton(self.radio_button_4)
        self.button_group.addButton(self.radio_button_5)
        self.button_group.addButton(self.radio_button_6)
        self.button_group.addButton(self.radio_button_7)
        self.button_group.addButton(self.radio_button_8)
        self.button_group.buttonClicked.connect(self._radiobutton_clk)
        self.grid_layout()
        self.show()

    def grid_layout(self):
        grid = QGridLayout()
        grid.setSpacing(10)
        grid.addWidget(self.radio_button_1, 1, 0)
        grid.addWidget(self.radio_button_2, 2, 0)
        grid.addWidget(self.radio_button_3, 3, 0)
        grid.addWidget(self.radio_button_4, 4, 0)
        grid.addWidget(self.radio_button_5, 5, 0)
        grid.addWidget(self.radio_button_6, 6, 0)
        grid.addWidget(self.radio_button_7, 7, 0)
        grid.addWidget(self.radio_button_8, 8, 0)
        grid.addWidget(self.label, 2, 2)
        self.setLayout(grid)

    def _radiobutton_clk(self, button):
        self.label.setText(members[button.text()])

if __name__ == "__main__":
    app = QApplication(sys.argv)
    window = MainWindow()
    window.setObjectName("MainWindow")
    window.setStyleSheet("#MainWindow{background-color:purple}")
    window.show()
    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.