

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

Кафедра инфокоммуникаций

Отчет по лабораторной работе №1

Основы работы с PySide2

По дисциплине «Объектно-ориентированное программирование»

Выполнил студент группы ИВТ-б-о-20-1

Галяс Д. 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.

Ссылка на репозиторий: <https://github.com/DIMITRY-GALYAS1/4.7-PySide2-.git>

Ход работы:

1. Создал новый репозиторий на github, после клонировал его и создал в папке репозитория новый проект PyCharm.
2. Выполнил первое задание.

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

"""
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".
"""

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

class MainWindow(QWidget):
    def __init__(self):
        super().__init__()
        self.label_1 = QLabel("Введите числа")
        self.label_1.setFont(QFont('Arial', 13))
        self.label_2 = QLabel("Результат операции:")
        self.line_1 = QLineEdit(self)
        self.line_2 = QLineEdit(self)
        self.btn1 = QPushButton("+", self)
        self.btn2 = QPushButton("-", self)
        self.btn3 = QPushButton("*", self)
        self.btn4 = QPushButton("/", self)
        self.initialization()

    def initialization(self):
        self.setGeometry(100, 100, 400, 230)
```

Рисунок 1. Код первого задания

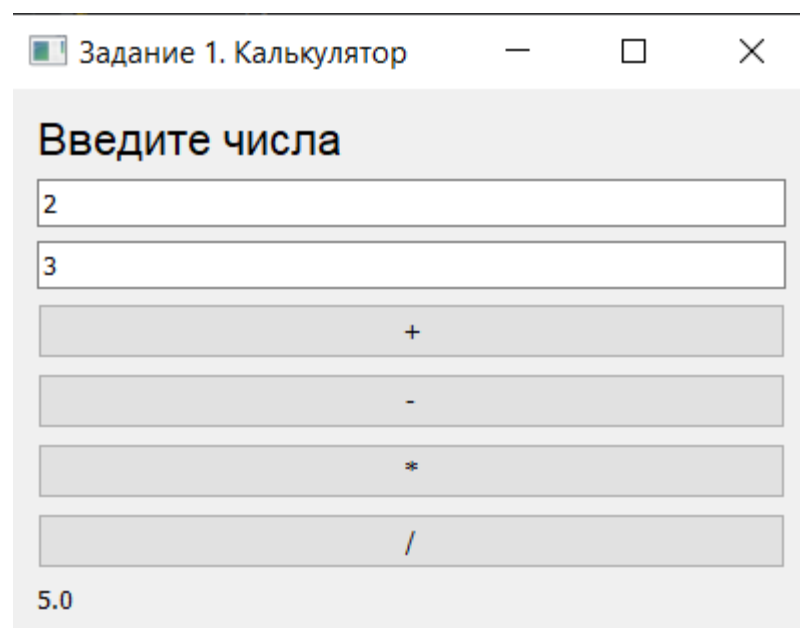


Рисунок 2. Результат работы кода

3. Выполнил второе задание.

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

"""
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.
Коды цветов в шестнадцатеричной кодировке: #ff0000 - красный, #ff7d00 - оранжевый, #ffff00 - желтый, #00ff00 - зеленый,
#007dff - голубой, #0000ff - синий, #7d00ff - фиолетовый.
"""

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

class MainWindow(QWidget):
    def __init__(self):
        super().__init__()
        self.label_1 = QLabel(self)
        self.line_1 = QLineEdit(self)
        self.bbtn1 = QPushButton(self)
        self.bbtn2 = QPushButton(self)
        self.bbtn3 = QPushButton(self)
        self.bbtn4 = QPushButton(self)
        self.bbtn5 = QPushButton(self)
        self.bbtn6 = QPushButton(self)
        self.bbtn7 = QPushButton(self)
        self.initialization()

    def initialization(self):
        self.setGeometry(100, 100, 400, 230)
        self.setWindowTitle("Задание 2. Выбор цвета")

MainWindow
```

Рисунок 3. Код второго задания

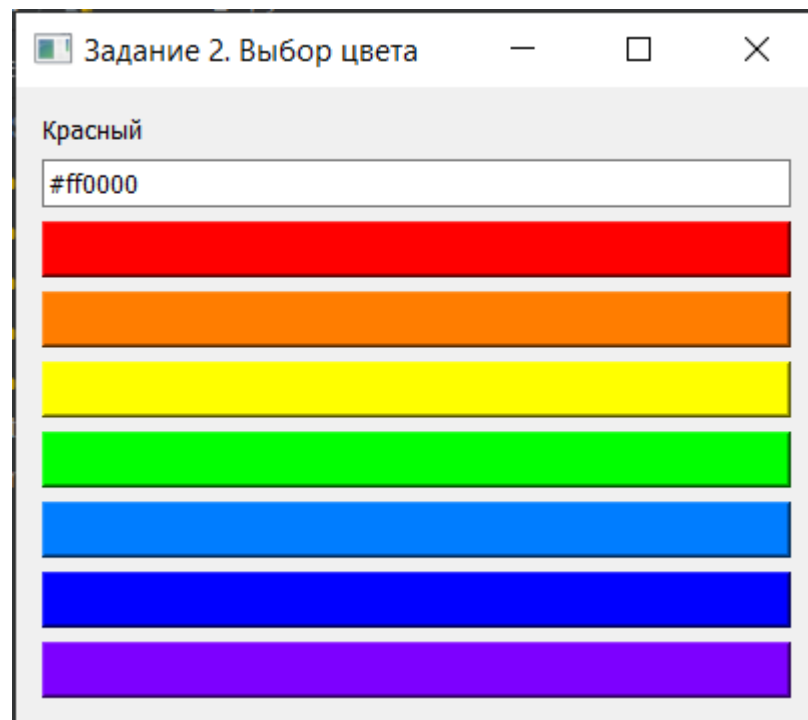


Рисунок 4. Работа кода

4. Сделал третье задание.

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

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

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

class MainWindow(QWidget):
    def __init__(self):
        super().__init__()
        self.label_1 = QLabel(self)
        self.label_1.setAlignment(Qt.AlignHCenter)
        self.line_1 = QLineEdit(self)
        self.butn1 = QPushButton(self)
        self.butn2 = QPushButton(self)
        self.butn3 = QPushButton(self)
        self.butn4 = QPushButton(self)
        self.butn5 = QPushButton(self)
        self.butn6 = QPushButton(self)
        self.butn7 = QPushButton(self)
        self.initialization()

    def initialization(self):
        self.setGeometry(100, 100, 100, 100)
        self.setWindowTitle("Задание 3")
        self.display_widgets()
```

Рисунок 5. Код третьего задания

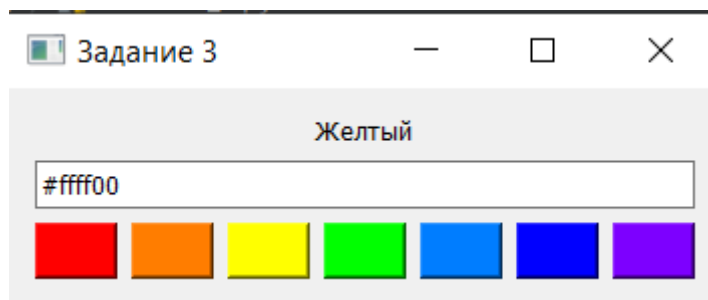


Рисунок 6. Работа кода

5. Выполнил четвертое задание.

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

"""
Напишите программу, состоящую из однострочного и многострочного текстовых полей и двух кнопок "Открыть" и "Сохранить".
При клике на первую кнопку 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.
Файлы будут читаться и записываться в том же каталоге, что и файл скрипта, если указывать имена файлов без адреса.
"""

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

class MainWindow(QWidget):
    def __init__(self):
        super().__init__()
        self.line_1 = QLineEdit(self)
        self.label_1 = QTextEdit(self)
        self.butn1 = QPushButton("Открыть файл", self)
        self.butn2 = QPushButton("Сохранить файл", self)
        self.initialization()

    def initialization(self):
        self.setGeometry(500, 500, 500, 500)
        self.setWindowTitle("Задание 4")
        self.display_widgets()
```

Рисунок 7. Код четвертого задания

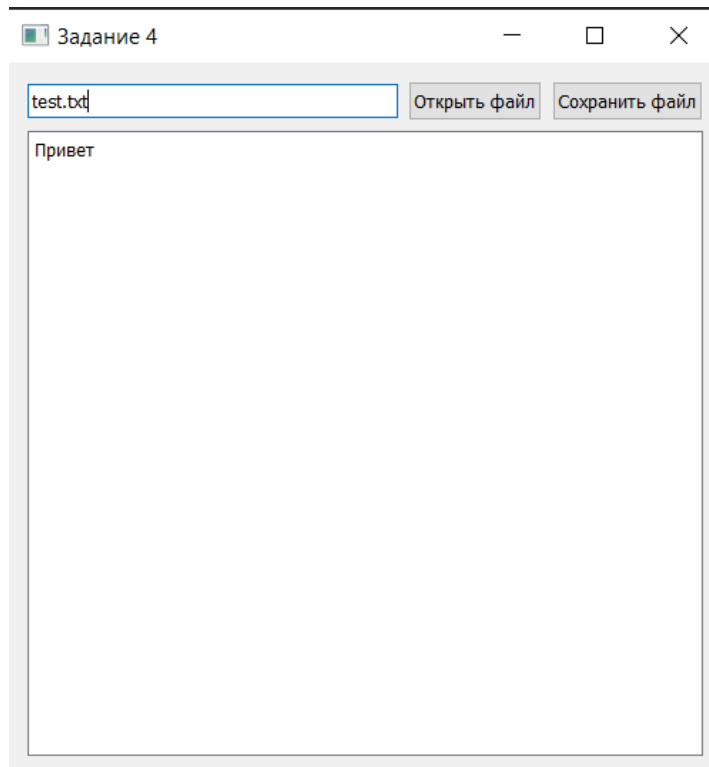


Рисунок 8. Работа четвертого задания

6. Приступил к выполнению пятого задания.

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

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

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

class MainWindow(QWidget):
    def __init__(self):
        super().__init__()
        self.label_1 = QLabel(self)
        self.label_1.setAlignment(Qt.AlignCenter)
        self.group = QButtonGroup()
        self.radio_1 = QPushButton('СКОУ')
        self.radio_2 = QPushButton('МИРЭА')
        self.radio_3 = QPushButton('СТГАУ')
        self.initialization()

    def initialization(self):
        self.setGeometry(200, 200, 350, 200)
        self.setWindowTitle("Задание 5")
        self.display_widgets()
```

Рисунок 9. Код пятого задания

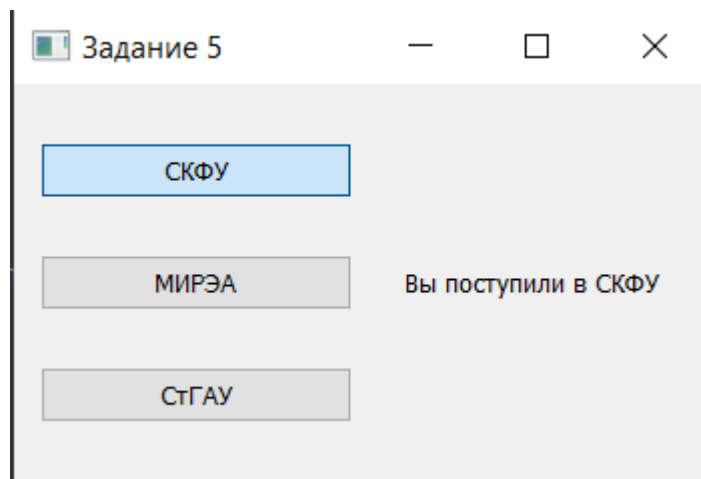


Рисунок 10. Работа пятого задания

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