

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

Кафедра инфокоммуникаций

**Отчет по лабораторной работе №4.8 PySide2
Создание UI
по дисциплине «Объектно-ориентированное программирование»**

Выполнил студент группы ИВТ-б-о-20-1

Ищенко Т.С. « » _____ 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

Ход работы:

1. Выполнил первое задание

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
"""
напишите программу, состоящую из двух списков Listbox.
В первом будет, например, перечень товаров, заданный программно.
Второй изначально пуст, пусть это будет перечень покупок.
При клике на одну кнопку товар должен переходить из одного списка в другой.
При клике на вторую кнопку – возвращаться (человек передумал покупать).
Предусмотрите возможность множественного выбора элементов списка и их перемещения.
"""

from PySide2.QtWidgets import (
    QApplication,
    QWidget,
    QVBoxLayout,
    QHBoxLayout,
    QPushButton,
    QListWidget,
    QAbstractItemView
)
import sys
```

Рисунок 1 – Фрагмент кода для первого задания

2. Результат работы кода для первого задания

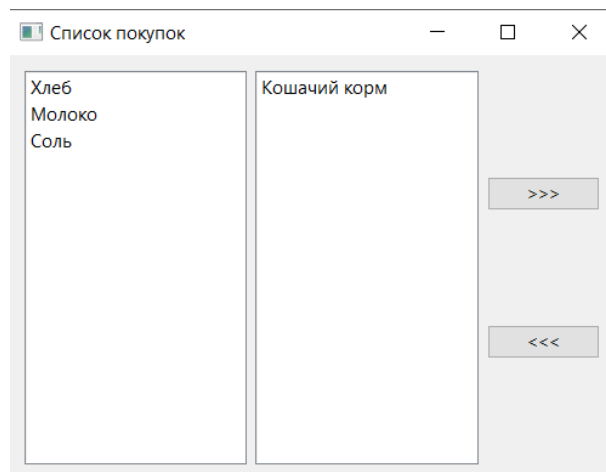


Рисунок 2 – Результат работы кода для первого задания

3. Выполнил второе задание

```
class ListBox(QWidget):
    def __init__(self):
        super().__init__()
        self.list = QListWidget()
        self.line_edit = QLineEdit()

        self.initial_ui()

    def initial_ui(self):
        self.setGeometry(350, 350, 500, 350)
        self.setWindowTitle("Список")

        self.list.itemDoubleClicked.connect(self.copy_text)
        self.line_edit.returnPressed.connect(self.move_text)

        layout = QVBoxLayout()
        layout.addWidget(self.list)
        layout.addWidget(self.line_edit)
        self.setLayout(layout)

    def copy_text(self):
        list = self.list.selectedItems()
        if not list: return
        for item in list:
            self.line_edit.setText(item.text())

    def move_text(self):
        self.list.addItem(self.line_edit.text())
        self.line_edit.clear()
```

Рисунок 3 – Фрагмент кода для второго задания

4. Результат работы кода для второго задания

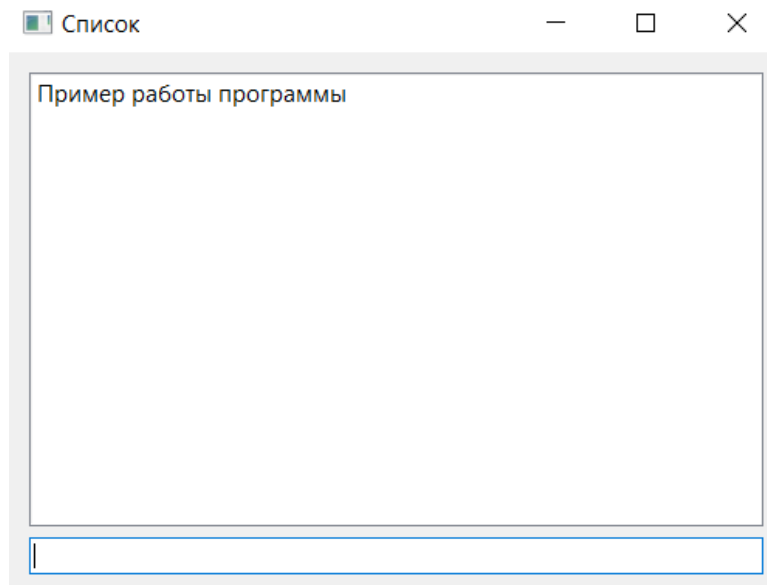


Рисунок 4 – Результат работы кода для второго задания

5. Выполнил третье задание

```
def initial_ui(self):
    self.setGeometry(200, 200, 300, 200)
    self.setWindowTitle("multiline text")

    QApplication.instance().focusChanged.connect(self.focus)
    self.button.clicked.connect(self.text_edit_resize)
    self.line_edit_second.returnPressed.connect(self.text_edit_resize)

    horizontal_layout = QHBoxLayout()
    vertical_layout = QVBoxLayout()
    horizontal_layout.addWidget(self.line_edit_first)
    horizontal_layout.addWidget(self.line_edit_second)
    horizontal_layout.addWidget(self.button)
    vertical_layout.addWidget(self.text_edit)
    vertical_layout.addLayout(horizontal_layout)
    self.setLayout(vertical_layout)

def focus(self, old, new):
    if self.text_edit == new:
        self.text_edit.setStyleSheet('background-color: #fff;')
    elif self.text_edit == old:
        self.text_edit.setStyleSheet("background-color: #d3d3d3;")
```

Рисунок 5 – Фрагмент кода для третьего задания

6. Результат работы кода для третьего задания

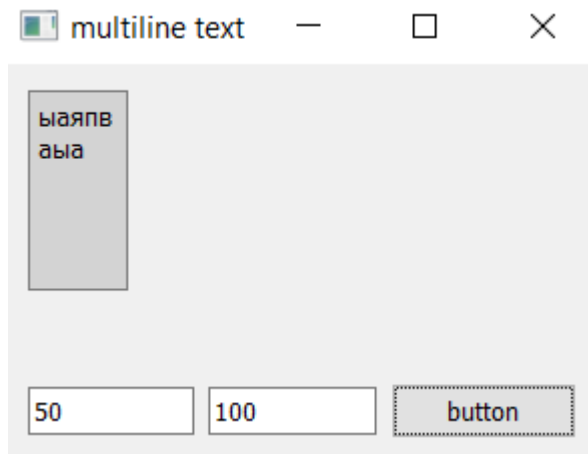


Рисунок 6 – Результат работы кода для третьего задания

7. Выполнил четвёртое задание

```
class HomePainting(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Рисунок")
        self.setGeometry(500, 500, 500, 400)
        self.setStyleSheet('background-color: #87ceeb;')

    def paintEvent(self, event):
        painter = QPainter(self)
        self.house(painter)
        self.sun(painter)
        self.grass(painter)
        # self.kit(painter)

    def house(self, painter):
        painter.setBrush(QColor(255, 255, 51))
        painter.drawRect(170, 138, 180, 250) # Корпус дома

        painter.setBrush(QColor(255, 178, 102))
        roof_points = QPolygon([
            QPoint(170, 138),
            QPoint(260, 57),
            QPoint(350, 138)
        ])
        painter.drawPolygon(roof_points) # Крыша
```

Рисунок 7 – Фрагмент кода для четвёртого задания

8. Результат работы кода для четвёртого задания

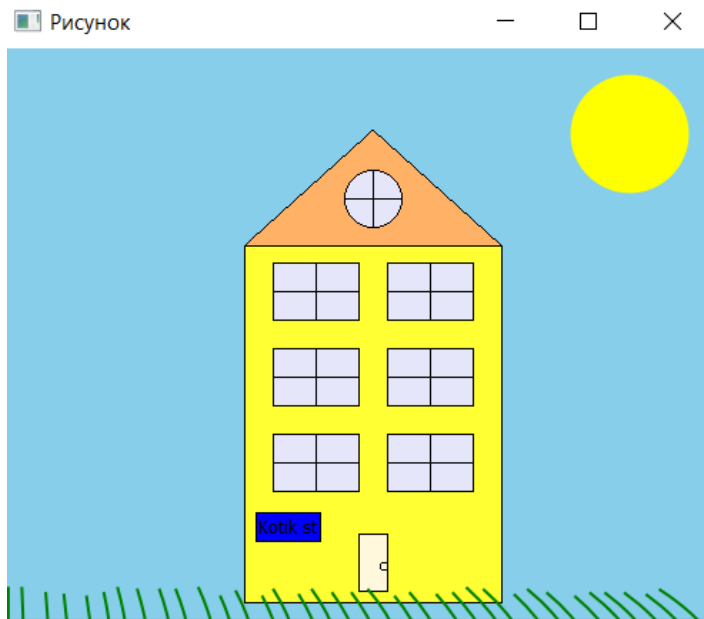


Рисунок 8 – Результат работы кода для четвёртого задания

9. Выполнил пятое задание

```
class Ball(QWidget):
    def __init__(self):
        super().__init__()
        self.ball = QWidget(self)
        self.animation = QPropertyAnimation(self.ball, b

        self.initial_ui()

    def initial_ui(self):
        self.setGeometry(350, 350, 350, 350)
        self.setWindowTitle("Шарик")

        self.ball.setStyleSheet("background-color: red ;
        self.ball.resize(50, 50)
        self.animation.setDuration(1000)

    def mousePressEvent(self, event):
        self.animation.setEndValue(QPoint(event.x() - 25
        self.animation.start()
```

Рисунок 9 – Фрагмент кода для пятого задания

10.Результат работы кода для пятого задания

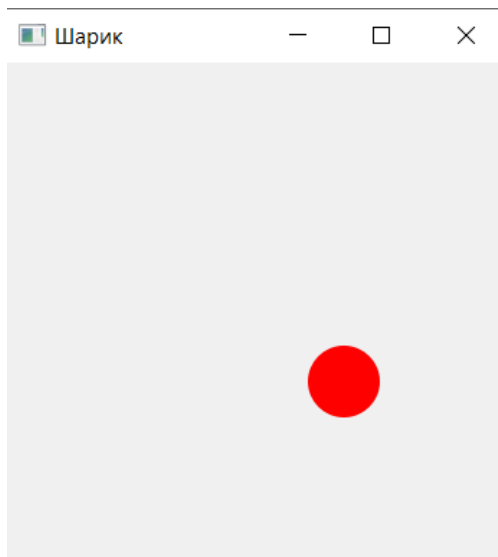


Рисунок 10 – Результат работы кода для пятого задания

Вывод: в ходе выполнения лабораторной работы приобрёл навыки построения графического интерфейса для пользователя с помощью пакета PySide2.