

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

Обработка событий и рисование в 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)

**Цель работы:** приобретение навыков улучшения графического интерфейса пользователя GUI с помощью обработки событий и рисования, реализованных в пакете 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 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

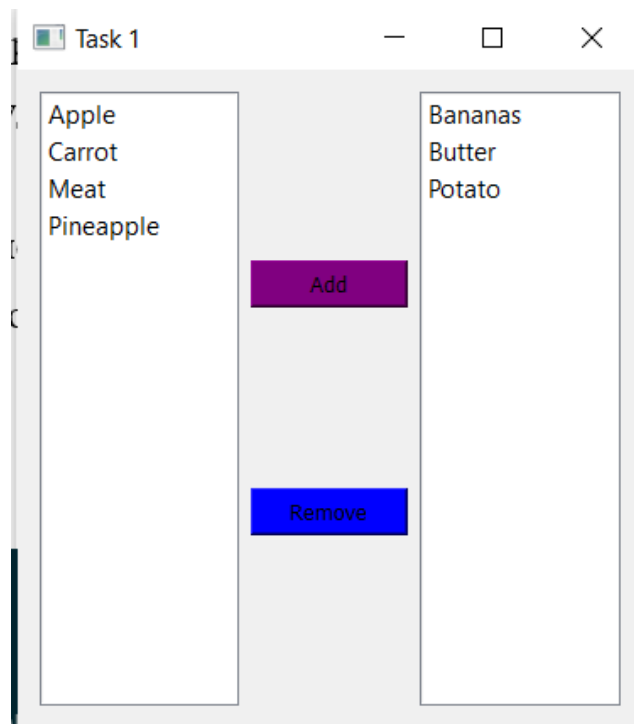


Рисунок 1 – Задание 1

Код:

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

import sys
from PySide2.QtWidgets import *
```

```

QApplication,
QWidget,
QPushButton,
QHBoxLayout,
QVBoxLayout,
QListWidget,
QAbstractItemView,
)

class MainWindow(QWidget):
    def __init__(self, trp):
        super().__init__()
        self.setWindowTitle("Task 1")
        self.setGeometry(370, 390, 370, 390)
        self.lst1 = QListWidget()
        self.lst1.setSelectionMode(QAbstractItemView.ExtendedSelection)
        self.lst2 = QListWidget()
        self.lst2.setSelectionMode(QAbstractItemView.ExtendedSelection)
        self.lst1.addItem(trp)
        self.inp1 = QPushButton("Add")
        self.inp2 = QPushButton("Remove")
        self.setting()

    def setting(self):
        hbox = QHBoxLayout()
        vbox = QVBoxLayout()
        hbox.addWidget(self.lst1)
        hbox.addLayout(vbox)
        vbox.addWidget(self.inp1)
        vbox.addWidget(self.inp2)
        hbox.addWidget(self.lst2)
        self.setLayout(hbox)
        self.inp1.clicked.connect(self.toright)
        self.inp2.clicked.connect(self.toleft)
        self.inp1.setStyleSheet("background: purple")
        self.inp2.setStyleSheet("background: blue")

    def toright(self):
        listItems = self.lst1.selectedItems()
        for item in listItems:
            self.lst1.removeItem(self.lst1.row(item))
            self.lst2.addItem(item)

    def toleft(self):
        listItems = self.lst2.selectedItems()
        for item in listItems:
            self.lst2.removeItem(self.lst2.row(item))
            self.lst1.addItem(item)

def main():
    app = QApplication(sys.argv)
    trp = (
        "Apple",
        "Bananas",
        "Carrot",
        "Butter",
        "Meat",
        "Potato",
        "Pineapple"
    )
    application = MainWindow(trp)
    application.show()

```

```

sys.exit(app.exec_())

if __name__ == "__main__":
    main()

```

Задание 2. Напишите программу по следующему описанию. Нажатие Enter в однострочном текстовом поле приводит к перемещению текста из него в список (экземпляр Listbox ). При двойном клике по элементу-строке списка, она должна копироваться в текстовое поле

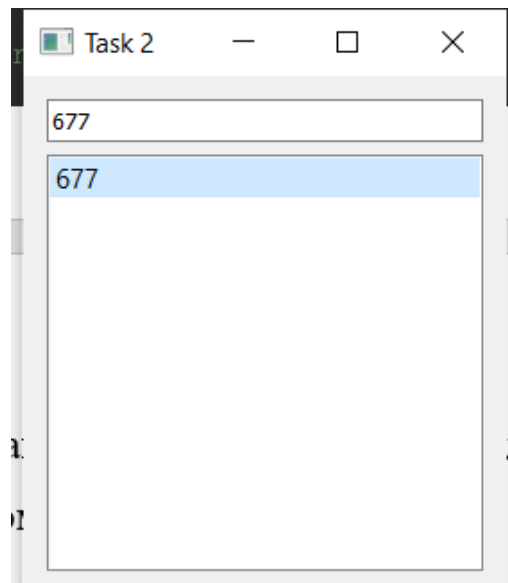


Рисунок 2 – Задание 2

Код:

```

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

import sys
from PySide2.QtWidgets import QApplication, QWidget, QLineEdit, QVBoxLayout, QListWidget

class MainWindow(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Task 2")
        self.setGeometry(270, 290, 270, 290)
        self.lst1 = QListWidget()
        self.lst1.itemDoubleClicked.connect(self.replaceitem)
        self.inp1 = QLineEdit()
        self.inp1.returnPressed.connect(self.replacetxt)
        self.create()

    def create(self):
        vbox = QVBoxLayout()
        vbox.addWidget(self.inp1)
        vbox.addWidget(self.lst1)

```

```

        self.setLayout(vbox)

    def replacetxt(self):
        self.lst1.addItem(self.inp1.text())
        self.inp1.clear()

    def replaceitem(self):
        listItems = self.lst1.selectedItems()
        if not listItems:
            return None
        for item in listItems:
            self.inp1.setText(item.text())

def main():
    app = QApplication(sys.argv)
    application = MainWindow()
    application.show()
    sys.exit(app.exec_())

if __name__ == "__main__":
    main()

```

Задание 3. Напишите программу по описанию. Размеры многострочного текстового поля определяются значениями, введенными в однострочные текстовые поля. Изменение размера происходит при нажатии мышью на кнопку, а также при нажатии клавиши Enter.

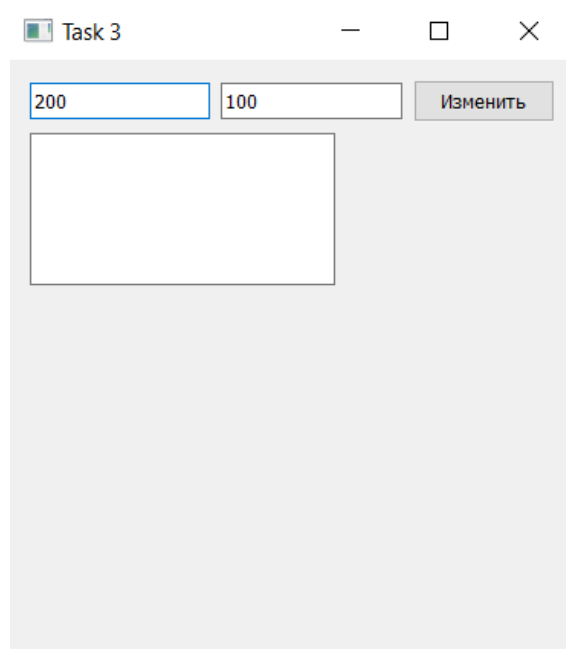


Рисунок 3 – Задание 3

Код:

```

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

import sys

```

```

from PySide2.QtWidgets import (
    QApplication,
    QWidget,
    QLineEdit,
    QHBoxLayout,
    QVBoxLayout,
    QPushButton,
    QTextEdit,
)

class MainWindow(QWidget):
    def __init__(self):
        super().__init__()
        QApplication.instance().focusChanged.connect(self.on_focusChanged)
        self.setWindowTitle("Task 3")
        self.setGeometry(370, 390, 370, 390)
        self.inp1 = QLineEdit()
        self.inp2 = QLineEdit()
        self.inp2.returnPressed.connect(self.edit_size)
        self.txtbox = QTextEdit()
        self.btn1 = QPushButton("ИЗМЕНИТЬ")
        self.btn1.clicked.connect(self.edit_size)
        self.widget()

    def widget(self):
        hbox = QHBoxLayout()
        vbox = QVBoxLayout()
        hbox.addWidget(self.inp1)
        hbox.addWidget(self.inp2)
        hbox.addWidget(self.btn1)
        vbox.addLayout(hbox)
        vbox.addWidget(self.txtbox)
        self.setLayout(vbox)
        self.w = 0
        self.h = 0

    def edit_size(self):
        self.w = int(self.inp1.text())
        self.h = int(self.inp2.text())
        self.txtbox.resize(self.w, self.h)

    def on_focusChanged(self, old, now):
        if self.txtbox == now:
            self.txtbox.setStyleSheet("background-color: white;")
        elif self.txtbox == old:
            self.txtbox.setStyleSheet("background-color: lightgrey;")

def main():
    app = QApplication(sys.argv)
    application = MainWindow()
    application.show()
    sys.exit(app.exec_())

if __name__ == "__main__":
    main()

```

Задание 4. Создайте на холсте изображение

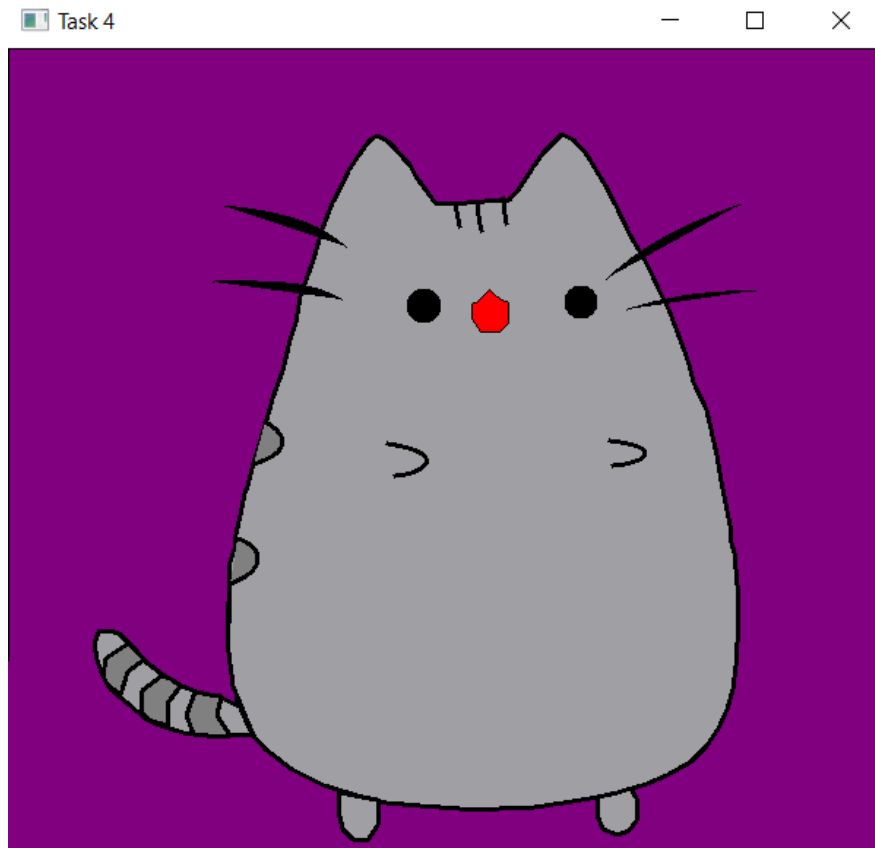


Рисунок 4 – Задание 4

Код:

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

import sys
from PySide2.QtWidgets import QWidget, QApplication
from PySide2.QtCore import QPoint, QPropertyAnimation

class MainWindow(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Task 5")
        self.setGeometry(200, 200, 500, 500)
        self.circle = QWidget(self)
        self.circle.setStyleSheet("background-color: green; border-radius:
50%;")
        self.circle.resize(100, 100)
        self.anim = QPropertyAnimation(self.circle, b"pos")
        self.anim.setDuration(1500)

    def mousePressEvent(self, event):
        self.anim.setEndValue(QPoint(event.x() - 25, event.y() - 25))
        self.anim.start()

def main():
    app = QApplication(sys.argv)
    window = MainWindow()
    window.show()
    window.setObjectName("MainWindow")
    window.setStyleSheet("#MainWindow{background-color:purple}")
    sys.exit(app.exec_())
```

```
if __name__ == "__main__":  
    main()
```

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