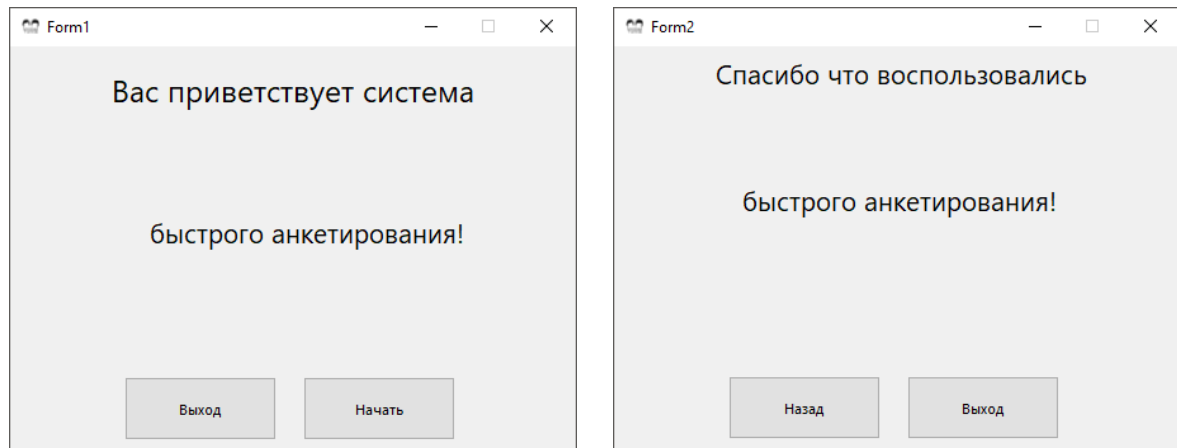


**ЛЕКЦИЯ 8 – МНОГООКОННЫЕ ПРОГРАММЫ В PYQT5**

```
import sys
from PyQt5 import QtCore, QtWidgets, uic, QtGui

data_forms = "" # Глобальная переменная для общих данных между формами

# Класс создания первой формы
class Form1(QtWidgets.QMainWindow):
    signal_emit = QtCore.pyqtSignal(str) # Переменная для генерирования текстового сигнала

    # Метод вызывается автоматически при создании класса
    def __init__(self):
        super(Form1, self).__init__()

        # Загрузка формы, установка заголовка и иконки
        uic.loadUi('uis/form1.ui', self)
        self.setWindowTitle('Form1')
        self.setWindowIcon(QtGui.QIcon('images/logo.png'))

        # Привязка к кнопке "btn_exit" формы стандартного обработчика закрытия формы
        self.btn_exit.clicked.connect(self.close)
        # Привязка к кнопке "btn next" формы обработчика нажатия - метод "next"
        self.btn_next.clicked.connect(self.next)

    # Метод-обработчик для перехода на следующую форму
    def next(self):
        global data_forms # Указываем, что переменная "data_forms" глобальная

        # Запоминаем в глобальной переменной значение из метки "label_welcome_2"
        data_forms = QtGui.QTextDocument(self.label_welcome_2.text()).toPlainText()

        self.signal_emit.emit('form2show') # Посылаем сигнал

# Класс создания второй формы
class Form2(QtWidgets.QMainWindow):
    signal_emit = QtCore.pyqtSignal(str) # Переменная для генерирования текстового сигнала

    # Метод вызывается автоматически при создании класса
    def __init__(self):
        super(Form2, self).__init__()
```

```
# Загрузка формы, установка заголовка и иконки
uic.loadUi('uis/form2.ui', self)
self.setWindowTitle('Form2')
self.setWindowIcon(QtGui.QIcon('images/logo.png'))

# Привязка к кнопке "btn_exit" формы стандартного обработчика закрытия формы
self.btn_exit.clicked.connect(self.close)
# Привязка к кнопке "btn_back" формы обработчика нажатия - метод "back"
self.btn_back.clicked.connect(self.back)

# Установка значения метки из общей глобальной переменной
global data_forms # Указываем, что переменная "data_forms" глобальная
self.label_2.setText(data_forms) # Устанавливаем значение метки из
глобальной переменной

# Метод-обработчик для перехода на предыдущую форму
def back(self):
    self.signal_emit.emit("form1show") # Посылаем сигнал

# Класс для показа форм (окон)
class Controller:
    def __init__(self):
        pass # Автоматически ничего не делаем при создании класса контроллера

# Показать первую форму
def start(self):
    self.form1 = Form1() # Создание первой формы
    self.form1.signal_emit.connect(
        self.show_hide_form) # Подключение сигналов первой формы к методу
"show_hide_form"
    self.form1.show() # Показ первой формы

# Универсальный метод показать/спрятать форму
def show_hide_form(self, text): # В переменной "text" текст сигнала
    # Показать вторую форму, спрятав первую
    if text == 'form2show': # Для сигнала "form2show"
        self.form2 = Form2() # Создание второй формы
        self.form2.signal_emit.connect(
            self.show_hide_form) # Подключение сигналов второй формы к методу
"show_hide_form"
        self.form1.close() # Закрытие первой формы
        self.form2.show() # Показ второй формы
    # Показать первую форму, спрятав вторую
    if text == 'form1show': # Для сигнала "form1show"
        self.form1 = Form1() # Создание первой формы
        self.form1.signal_emit.connect(
            self.show_hide_form) # Подключение сигналов первой формы к методу
"show_hide_form"
        self.form2.close() # Закрытие второй формы
        self.form1.show() # Показ первой формы

# Основной метод программы
def main():
    app = QtWidgets.QApplication(sys.argv)
    controller = Controller() # Создаем контроллер окон
    controller.start() # Запускаем контроллер окон
    sys.exit(app.exec_())

if __name__ == '__main__': # Запуск программы
    main()
```