**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П.О.СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

Дисциплина «высокоуровневые языки программирования»

ОТЧЕТ

Лабораторная работа № 3

Основные компоненты. Размещение компонентов в окнах

Выполнил студент

Группы ИТД-21

Чайдаков И.М.

Принял преподаватель

Ястребов А. А.

Гомель 2024

Основные компоненты. Размещение компонентов в окнах

**Цель работы:** изучить основные компоненты, определенные в модуле *QtWidgets*; изучить способы размещения компонентов в окнах.

**Задание:** Создать, которое при запуске выводит окно с надписью. Поменять значок в заголовке на свой. В заголовке окна указать свои имя, фамилию и группу.

Под надписью должно находиться 3 кнопки. При наведении мыши на кнопку должна выводиться подсказка.

Кнопки

1 — Открыть окно с анкетой. Использовать типы размещения и типы компонентов. После нажатия на кнопку отправить все ответы пользователя должны выводиться в сеть.

2 — Открыть окно, в котором по введенному значению аргумента, вычисляется значение функции, заданной в виде графика. В окне должен быть организован ввод пользователем аргумента *X*. И вывод значения аргумента *Y*. На форме вывести график в виде рисунка. Необходимо добавить кнопки: «решить», «очистить», «выход».

3 — Открыть окно, в котором по введенному значению аргумента, определяется попадает ли точка с заданными координатами в область, закрашенную серым цветом. В окне должен быть организован ввод пользователем аргументов *X*, *Y* и радиуса *R*. И вывод — попадает или не попадает точка в область. На форме исходные данные вывести в виде рисунка. Необходимо добавить кнопки: «определить», «очистить», «выход».

**Ход выполнения задания:**

**1.** Описываем класс для создания главного окна (рисунок 1).

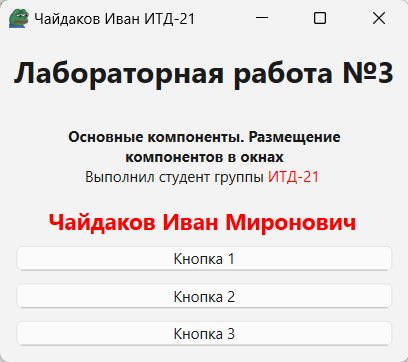


Рисунок 1 — Основное окно для вызова остальных окон.

**2.** Описываем класс для вывода анкеты и отображаем ее на экране нажатием второй кнопки (рисунок 2).

**3.** Описываем код основной программы для запуска методов ранее описанных классов (см. Приложение C).

**4.** Запускает файл основной программы и видим перед собой основное окно с опциями (Рисунок 1).

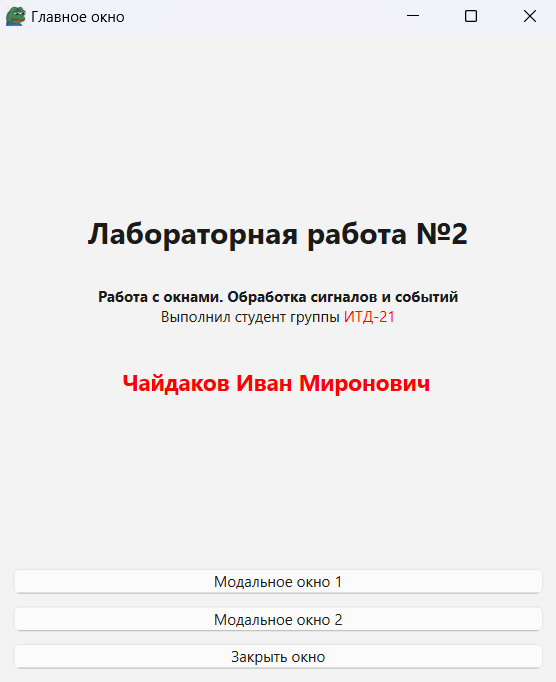


Рисунок 1 — Основное окна с кнопками вызова дополнительных.

**5.** Нажимаем на кнопку "Модальное окно 1" и открываем справа от основного окна модальное окно с прозрачным фоном (Рисунок 2).

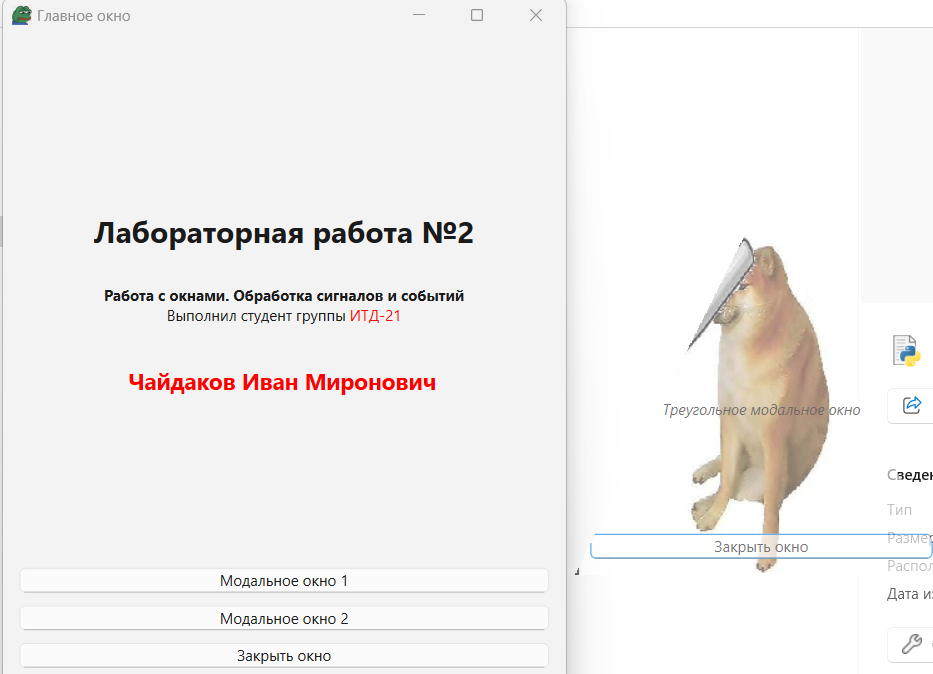


Рисунок 2 — Первое модальное окно.

**6.** Нажимаем на кнопку "Модальное окно 2" и в центре экрана модальное окно считывающие нажатия клавиши (Рисунок 3).

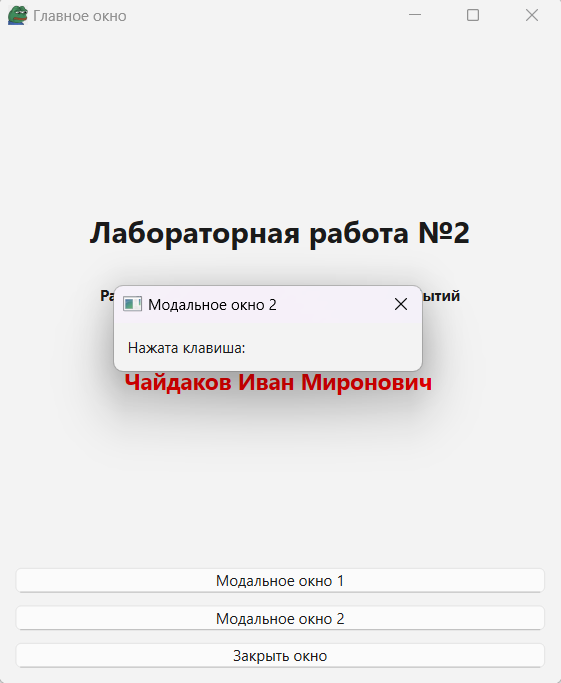


Рисунок 3 — Второе модальное окно.

**Вывод:** Использование компонентов в окнах открывает доступ к более сложным структурам окон и возможности работы с большим количеством данных пользователя. Компоненты являются очень полезным решением для создания программ для математических расчетов на языке *Python*.

**ПРИЛОЖЕНИЕ А**

**Классы для создания модальных окон**

from PyQt6.QtWidgets import QApplication, QDialog, QVBoxLayout, QLabel, QPushButton

from PyQt6.QtGui import QRegion, QPolygon, QPalette, QBrush, QPixmap

from PyQt6.QtCore import QPoint, Qt

class ModalTypeA(QDialog):

    def modalTypeA(self, parent=None):

        self.setWindowTitle("Модальное окно 1")

        self.setModal(True)

        self.setGeometry(100, 100, 300, 300)

        self.setWindowOpacity(0.6)

        palette = self.palette()

        palette.setBrush(QPalette.ColorGroup.Normal, QPalette.ColorRole.Window,

                        QBrush(QPixmap("Image/Bonk.jpg").scaled(self.size(), Qt.AspectRatioMode.IgnoreAspectRatio)))

        self.setPalette(palette)

        polygon = QPolygon([QPoint(0, 300), QPoint(150, 0), QPoint(300, 300)])

        region = QRegion(polygon)

        self.setMask(region)

        self.label = QLabel("<center><i><br><br><br><br>Треугольное модальное окно</i></center>")

        self.btnQuit = QPushButton("&Закрыть окно")

        self.vbox = QVBoxLayout()

        self.vbox.addWidget(self.label)

        self.vbox.addWidget(self.btnQuit)

        self.setLayout(self.vbox)

        self.btnQuit.clicked.connect(self.close)

        self.move(1000, 250)

        self.exec()

class ModalTypeB(QDialog):

    def modalTypeB(self, parent=None):

        self.setWindowTitle("Модальное окно 2")

        self.setModal(True)

        self.label = QLabel("<center>Модальное окно считывающие нажатия</center>")

        self.vbox = QVBoxLayout()

        self.vbox.addWidget(self.label)

        self.setLayout(self.vbox)

        self.exec()

    def keyPressEvent(self, event):

            if event.key() == Qt.Key.Key\_Escape:

                self.label.setText("Нажата клавиша Escape")

                self.close()

            elif event.key() == Qt.Key.Key\_Enter or event.key() == Qt.Key.Key\_Return:

                self.label.setText("Нажата клавиша Enter")

            else:

                self.label.setText(f"Нажата клавиша: {event.text()}")

if \_\_name\_\_ == "\_\_main\_\_":

    app = QApplication([])

    modal = ModalTypeA()

    window = QPushButton("открыть модальное окно")

    window.clicked.connect(modal.modalTypeA)

    window.show()

    app.exec()

**ПРИЛОЖЕНИЕ B**

**Класс для создания основного окна**

from PyQt6.QtWidgets import QWidget, QApplication, QVBoxLayout, QLabel, QPushButton

from PyQt6.QtGui import QIcon

from ModalWindow import ModalTypeA, ModalTypeB # type: ignore

class MainWindow(QWidget):

    def \_\_init\_\_(self, parent=None):

        QWidget.\_\_init\_\_(self, parent)

        self.modalA = ModalTypeA()

        self.modalB = ModalTypeB()

        self.setWindowIcon(QIcon("Image/Pepe.jpg"))

        self.label = QLabel('''

            <center>

                <h1>Лабораторная работа №2</h1><br>

                <b>Работа с окнами. Обработка сигналов и событий</b><br>

                Выполнил студент группы <span style="color: red;">ИТД-21<br>

                <h2>Чайдаков Иван Миронович</h2></span>

            </center>

        ''')

        modalButton\_1 = QPushButton("Модальное окно 1")

        modalButton\_1.setToolTip("Модальное окно треугольной формы")

        modalButton\_2 = QPushButton("Модальное окно 2")

        modalButton\_2.setToolTip("Модальное окно считывающие клавиши")

        self.btnQuit = QPushButton("&Закрыть окно")

        self.vbox = QVBoxLayout()

        self.vbox.addWidget(self.label)

        self.vbox.addWidget(modalButton\_1)

        self.vbox.addWidget(modalButton\_2)

        self.vbox.addWidget(self.btnQuit)

        self.setLayout(self.vbox)

        modalButton\_1.clicked.connect(self.modalA.modalTypeA)

        modalButton\_2.clicked.connect(self.modalB.modalTypeB)

        self.btnQuit.clicked.connect(QApplication.instance().quit)

if \_\_name\_\_ == "\_\_main\_\_":

    import sys

    app = QApplication(sys.argv)

    window = MainWindow()

    window.setWindowTitle("Главное окно")

    window.resize(300, 70)

    window.show()

    sys.exit(app.exec())

**ПРИЛОЖЕНИЕ C**

**Код основной программы**

from PyQt6.QtWidgets import QApplication

from MainWindow import MainWindow # type: ignore

app = QApplication([])

window = MainWindow()

window.setWindowTitle("Главное окно")

window.resize(450, 520)

window.show()

app.exec()