ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ РЕСПУБЛИКИ ХАКАСИЯ  
«ХАКАССКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

Специальность 09.02.07

«Информационные системы и программирование»

ТЕМА: Компоновка в PyQt

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Студент гр. | ИС(ТП)-31 |  |  | | | |  |  | Соломатин А.Д. | | |
|  |  |  | *подпись* | | | |  |  | *Фамилия, И.О.* | | |
| Руководитель |  |  |  |  |  |  | | | |  | Брюханова И.Н. |
|  | *оценка* |  | *дата* |  |  | *подпись* | | | |  | *Фамилия, И.О.* |

Абакан, 2024

**Цель работы: https://github.com/Aleksey4ik6/Alekseyka**

**Задание №1**

Вашему новому заказчику нужен свой собственный калькулятор.

Верстальщики предоставили вам Wireframe (чертёж (макет) приложения),

представленный на рисунке 1, в котором показано как конкретно должны

располагаться элементы пользовательского интерфейса.

Используя возможности PyQt, реализуйте пользовательский интерфейс

по следующему макету. Размеры и отступы должны совпадать и иметь

минимальную погрешность. Приложение не должно ничего выполнять.

2.Листинг программы:

import sys

from PyQt5.QtWidgets import (

QApplication, QWidget, QGridLayout, QLineEdit, QPushButton, QVBoxLayout

)

from PyQt5.QtCore import Qt

class CalculatorUI(QWidget):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.initUI()

def initUI(self):

self.setWindowTitle("Calculator Wireframe")

self.setFixedSize(300, 400)

main\_layout = QVBoxLayout(self)

main\_layout.setSpacing(10)

self.display = QLineEdit()

self.display.setFixedHeight(50)

self.display.setAlignment(Qt.AlignRight)

self.display.setReadOnly(True)

main\_layout.addWidget(self.display)

button\_grid = QGridLayout()

button\_grid.setSpacing(5)

positions = [(i, j) for i in range(5) for j in range(4)]

for position in positions:

button = QPushButton("")

button.setFixedSize(50, 50)

button\_grid.addWidget(button, \*position)

main\_layout.addLayout(button\_grid)

self.setLayout(main\_layout)

if \_\_name\_\_ == '\_\_main\_\_':

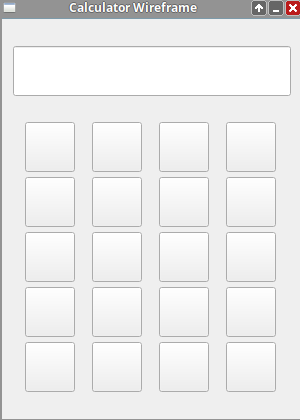
app = QApplication(sys.argv)

calculator = CalculatorUI()

calculator.show()

sys.exit(app.exec\_())

3.Результаты

****

**Задание №2**

В материалах лекции не разобран ещё один вид компоновки –

QFormLayout. Найдите и предоставьте информацию о нём в отчёте (не

используйте материалы по Qt и C++, вас интересуют материалы из PyQt на

Python).

С помощью QFormLayout реализуйте форму регистрации пользователя

по следующим требованиям:

− должно быть поле ввода логина;

− поле ввода пароля (для сокрытия ввода используйте метод

.setEchoMode(QLineEdit.Password);

− выпадающий список (QComboBox) для выбора страны;

− кнопка «Войти»;

− кнопка «Забыли пароль?»;

− каждый элемент интерфейса должен быть подписан (либо через

QLabel, либо используя стандартные средства (для полей ввода

.setPlaceholderText, для кнопок указан при создании);

− все элементы должны быть расположены ровно;

− окно должно иметь корректный размер с читабельным текстом.

QFormLayout в PyQt используется для организации элементов интерфейса в виде формы. Он автоматически размещает метки и соответствующие виджеты в два столбца: метки располагаются в левом столбце, а виджеты — в правом. Такой способ компоновки удобен для создания форм ввода данных, где каждый элемент интерфейса должен быть подписан и расположен ровно.

**Краткое описание QFormLayout:**

• **Назначение**: Упрощает создание форм ввода с двумя столбцами — метками и виджетами.

• **Методы**: • addRow(label, widget) — добавляет строку в форму, где label — название поля, а widget — элемент ввода (например, QLineEdit, QComboBox). • setSpacing(value) — устанавливает расстояние между строками. • setAlignment() — задаёт выравнивание элементов.

2.Листинг программы:

import sys

from PyQt5.QtWidgets import (

QApplication, QWidget, QFormLayout, QLineEdit, QPushButton, QComboBox, QLabel

)

from PyQt5.QtCore import Qt

class RegistrationForm(QWidget):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.setWindowTitle("Форма регистрации")

self.setup\_ui()

def setup\_ui(self):

# Создание компоновщика формы

layout = QFormLayout()

# Поле ввода логина

self.login\_input = QLineEdit()

self.login\_input.setPlaceholderText("Введите логин")

layout.addRow("Логин:", self.login\_input)

# Поле ввода пароля

self.password\_input = QLineEdit()

self.password\_input.setEchoMode(QLineEdit.Password) # Скрытие ввода пароля

self.password\_input.setPlaceholderText("Введите пароль")

layout.addRow("Пароль:", self.password\_input)

# Выпадающий список для выбора страны

self.country\_combo = QComboBox()

self.country\_combo.addItems(["Россия", "США", "Германия", "Франция", "Китай"])

layout.addRow("Страна:", self.country\_combo)

# Кнопка "Войти"

self.login\_button = QPushButton("Войти")

layout.addRow(self.login\_button)

# Кнопка "Забыли пароль?"

self.forgot\_password\_button = QPushButton("Забыли пароль?")

layout.addRow(self.forgot\_password\_button)

# Настройка окна

self.setLayout(layout)

self.setFixedSize(300, 200) # Фиксированный размер окна

self.show()

# Запуск приложения

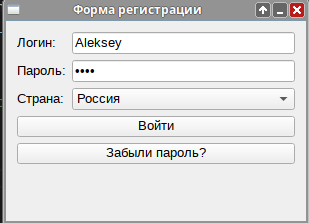
if \_\_name\_\_ == "\_\_main\_\_":

app = QApplication(sys.argv)

window = RegistrationForm()

sys.exit(app.exec\_())

3.Результаты

****