**Диалоги ввода информации**

Кроме диалога, позволяющего выбрать файл, существуют и другие.

Рассмотрим виджет **QInputDialog**. Особенность этого виджета в том, что его можно настроить на получение различных типов данных: строки (одной или нескольких), числа или одного значения из списка, вызывая различные методы.

Напишем программу, которая получает имя пользователя с помощью диалогового окна.

import sys

from PyQt5.QtWidgets import QWidget, QApplication, QPushButton

from PyQt5.QtWidgets import QInputDialog

class Example(QWidget):

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

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

self.initUI()

def initUI(self):

self.setGeometry(300, 300, 150, 150)

self.setWindowTitle('Диалоговые окна')

self.button\_1 = QPushButton(self)

self.button\_1.move(20, 40)

self.button\_1.setText("Кнопка")

self.button\_1.clicked.connect(self.run)

def run(self):

name, ok\_pressed = QInputDialog.getText(self, "Введите имя",

"Как тебя зовут?")

if ok\_pressed:

self.button\_1.setText(name)

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

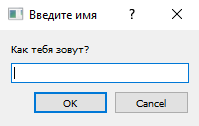
app = QApplication(sys.argv)

ex = Example()

ex.show()

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

В методе, вызываемой нажатием на кнопку, создается диалоговое окно (QInputDialog) и указывается, какой тип оно будет возвращать (getText). В качестве параметров передаются родительское окно (self), заголовок нового окна и сообщение для пользователя. Возвращает эта функция кортеж, где на первом месте записаны введенные пользователем данные (или пустая строка, если нажата кнопка «Отмена»), а на втором — состояние кнопки «Ok».



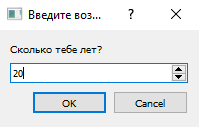
В случае если мы хотим получить целое число, лучше воспользоваться методом getInt(), в котором мы можем указать значение по умолчанию, минимальное значение, максимальное значение и шаг.

age, ok\_pressed = QInputDialog.getInt(

self, "Введите возраст", "Сколько тебе лет?",

20, 18, 27, 1)

В этом примере значение по умолчанию — 20, минимальное значение — 18, максимальное — 27, а шаг — 1.



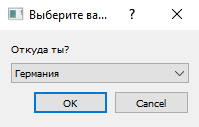
Иногда нам нужно, чтобы пользователь выбрал какое-то значение из предоставленных. Для этого есть метод getItem.

country, ok\_pressed = QInputDialog.getItem(

self, "Выберите вашу страну", "Откуда ты?",

("Россия", "Германия", "США"), 1, False)

В качестве параметров необходимо указать: родителя, заголовок окна, вопрос, итерируемый объект с вариантами ответа и индекс значения по умолчанию. А чтобы пользователь не смог что-то самостоятельно ввести, укажем следующим параметром False.



Для выбора цвета существует другой виджет — **QColorDialog**. Вот пример программы с его использованием:

import sys

from PyQt5.QtWidgets import QWidget, QApplication, QPushButton

from PyQt5.QtWidgets import QColorDialog

class Example(QWidget):

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

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

self.initUI()

def initUI(self):

self.setGeometry(300, 300, 150, 150)

self.setWindowTitle('Диалоговые окна')

self.button\_1 = QPushButton(self)

self.button\_1.move(20, 40)

self.button\_1.setText("Кнопка")

self.button\_1.clicked.connect(self.run)

def run(self):

color = QColorDialog.getColor()

if color.isValid():

self.button\_1.setStyleSheet(

"background-color: {}".format(color.name()))

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

app = QApplication(sys.argv)

ex = Example()

ex.show()

sys.exit(app.exec())

Если цвет валиден, фон кнопки окрасится в него. Функция .name() возвращает цвет в шестнадцатитиричном формате. Управлять стилями элементов достаточно просто, синтаксис похож на CSS, с которым вы должны были познакомиться во время изучения HTML.

Примеры можно посмотреть в [документации](https://doc.qt.io/qtforpython/overviews/stylesheet-examples.html).

