**class Start\_window**(PyQt5.QtWidgets.QMainWindow, converted\_forms\_to\_py.start\_window.Ui\_start\_window)

| Класс стартового окна

|

| Method resolution order:

| Start\_window

| PyQt5.QtWidgets.QMainWindow

| PyQt5.QtWidgets.QWidget

| PyQt5.QtCore.QObject

| sip.wrapper

| PyQt5.QtGui.QPaintDevice

| sip.simplewrapper

| converted\_forms\_to\_py.start\_window.Ui\_start\_window

| builtins.object

|

| Methods defined here:

|

| \_\_init\_\_(self)

| Initialize self. See help(type(self)) for accurate signature.

|

| start\_laba\_11(self)

| Открывает форму 11ой лабы

|

| start\_laba\_14(self)

| Открывает форму 14ой лабы

|

| start\_laba\_15(self)

| Открывает форму 15ой лабы

|

| start\_laba\_2(self)

| Открывает форму 2ой лабы

**class Laba2**(PyQt5.QtWidgets.QMainWindow, converted\_forms\_to\_py.laba2.Ui\_Laba2)

| Класс лабы 2, инициализирует форму и

| заполняет её элементами

|

| Method resolution order:

| Laba2

| PyQt5.QtWidgets.QMainWindow

| PyQt5.QtWidgets.QWidget

| PyQt5.QtCore.QObject

| sip.wrapper

| PyQt5.QtGui.QPaintDevice

| sip.simplewrapper

| converted\_forms\_to\_py.laba2.Ui\_Laba2

| builtins.object

|

| Methods defined here:

|

| \_\_init\_\_(self)

| Initialize self. See help(type(self)) for accurate signature.

|

| change\_picture\_key(self)

| Коннектится к key\_slider, при изменении

| его значения показывает в label\_key другую

| картинку,соответствующую текущему

| положению ключа

|

| change\_slot\_battery(self)

| Меняет положение батарейки на

| противоположное А так же отрисовывает

| положение батарейки(в слоте она или

| нет)

|

| change\_value\_galvanometer(self)

| Коннектится к Reostat'у, при изменении

| его значение, рассчитывает нужное

| показание гальванометра и обновляет

| изображение гальванометра

|

| eventFilter(self, obj, e)

| Для кликабельности лейблов

| :param obj: лайбел

| :param e: событие

|

| show\_info\_about\_laba(self)

| Показывает окно с документацией по

| выполнению л.р.

**class Laba11**(PyQt5.QtWidgets.QMainWindow, converted\_forms\_to\_py.laba11.Ui\_Laba11)

| Класс лабы 11, инициализирует форму и

| заполняет её элементами

|

| Method resolution order:

| Laba11

| PyQt5.QtWidgets.QMainWindow

| PyQt5.QtWidgets.QWidget

| PyQt5.QtCore.QObject

| sip.wrapper

| PyQt5.QtGui.QPaintDevice

| sip.simplewrapper

| converted\_forms\_to\_py.laba11.Ui\_Laba11

| builtins.object

|

| Methods defined here:

|

| \_\_init\_\_(self)

| Initialize self. See help(type(self)) for accurate signature.

|

| change\_map\_resistors(self)

| Меняет картинку label\_resistors в

| зависимости от выбранного типа

| подключения и выбранных резисторов тем

| самым отрисовывая вид текущего

| подключения

|

| change\_resistors\_store\_value(self)

| Пересчитывает суммарное

| сопротивление на магазине

| сопротивлений. Метод подключается

| коннектом к каждому dial'y

|

| check\_only\_one\_checkbox\_true(self)

| Проверяет подключено ли более 1

| резистора Если только 1 резистор

| включён,то возвращает true, в противном

| случае false

|

| eventFilter(self, obj, e)

| Нужен для кликабельности лейблов

| :param obj: лайбел

| :param e: событие

|

| reset(self)

| Сбрасывает все dial'ы на позицию 0. тем

| самым обнуляя сопротивление в мазазине

| сопротивлений

|

| resistance\_calculation(self)

| Считает и устанавливает текущее

| сопротивление в зависимости от

| выбранного типа подключения и

| выбранных резисторов

|

| show\_info\_about\_laba(self)

| Показывает окно с документацией по

| выполнению л.р.

**class Laba14**(PyQt5.QtWidgets.QMainWindow, converted\_forms\_to\_py.laba14.Ui\_Laba14)

| Класс лабы 14, инициализирует форму и

| заполняет её элементами

|

| Method resolution order:

| Laba14

| PyQt5.QtWidgets.QMainWindow

| PyQt5.QtWidgets.QWidget

| PyQt5.QtCore.QObject

| sip.wrapper

| PyQt5.QtGui.QPaintDevice

| sip.simplewrapper

| converted\_forms\_to\_py.laba14.Ui\_Laba14

| builtins.object

|

| Methods defined here:

|

| \_\_init\_\_(self)

| Initialize self. See help(type(self)) for accurate signature.

|

| set\_amperage(self)

| Коннектится к amperDial, когда изменяется

| значение пересчитывает ампераж и

| передаёт его амперметру,чтобы тот

| обновил показания

|

| set\_induction(self)

| Коннектится к amperDial и slider\_stock, когда

| изменяется значение одного из них,то

| Вычисляет индукцию магнитного поля

| соленоида(штока) в соответствии с

| формулой

|

| show\_info\_about\_laba(self)

| Показывает окно с документацией по

| выполнению л.р.

**class Laba15**(PyQt5.QtWidgets.QMainWindow, converted\_forms\_to\_py.laba15.Ui\_Laba15)

| Класс лабы 15, инициализирует форму и заполняет её элементами

|

| Method resolution order:

| Laba15

| PyQt5.QtWidgets.QMainWindow

| PyQt5.QtWidgets.QWidget

| PyQt5.QtCore.QObject

| sip.wrapper

| PyQt5.QtGui.QPaintDevice

| sip.simplewrapper

| converted\_forms\_to\_py.laba15.Ui\_Laba15

| builtins.object

|

| Methods defined here:

|

| \_\_init\_\_(self)

| Initialize self. See help(type(self)) for accurate signature.

|

| eventFilter(self, obj, e)

| Нужен для кликабельности лейблов

| :param obj: лайбел

| :param e: событие

|

| measure\_capacitor(self)

| Изменяет напряжение конденсатора, если включён милливольтметр

|

| measure\_coil(self)

| Изменяет напряжение на катушке, если включён милливольтметр

|

| measure\_resistor(self)

| Изменяет напряжение на резисторе, если включён милливольтметр

|

| show\_info\_about\_laba(self)

| Показывает окно с документацией по выполнению л.р.

|

| update\_ammeter(self)

| Коннектится к slider\_voltage и при изменении значения

| обновняет value в поле класса voltage\_regulator и обновляет картинку амперметра

**class svg\_widget\_ammeter**(builtins.object)

| Свг-виджет, прибор амперметр.

| Для использования в форме необходимо:

| 1) создать layout с размером и желаемым расположением прибора

| 2) добавить в этот layout поле self.svg\_widget экземпляра данного класса

| 3) всё готово!

|

| Methods defined here:

|

| \_\_init\_\_(self)

| Initialize self. See help(type(self)) for accurate signature.

|

| create\_widget\_ammeter(self)

| Создаёт свг\_виджет, устанавливает стрелку прибора на -60 градусов,

| тобишь на 0 ампер

| :return: готовый к интеграции виджет

|

| update\_svg\_ammeter(self, value\_ammeter)

| Обновляет показания прибора,путём изменения угла наклона стрелки прибора

| 1 градус равен 0,0083 ампер, соответственно 120 градусов это 1 ампер, 0 это 0

| :param value\_ammeter: это то количество амперов,которое нужно отобразить на приборе

|

| value(self)

| :return: Возвращает текущее показание амперметра

**class svg\_widget\_milli\_voltmeter**(builtins.object)

| Свг-виджет, прибор милливольтметер.

| Для использования в форме необходимо:

| 1) создать layout с размером и желаемым расположением прибора

| 2) добавить в этот layout поле self.svg\_widget экземпляра данного класса

| 3) всё готово!

|

| Methods defined here:

|

| \_\_init\_\_(self)

| Initialize self. See help(type(self)) for accurate signature.

|

| change\_color\_lamp\_svg(self)

| Меняет цвет лампы индикации подключения прибора к сети на протиположный

|

| change\_power\_svg(self)

| Меняет положение рычажка включения/отключения прибора на противоположное от имеющегося сейчас

|

| create\_widget\_milli\_voltmeter(self)

| Создаёт свг\_виджет, устанавливает стрелку прибора на 0 градусов,

| тобишь на 0 вольт

| :return: готовый к интеграции виджет

|

| update\_angle\_svg\_milli\_voltmeter(self, value\_milli\_voltmeter)

| Обновляет показания прибора,путём изменения угла наклона стрелки прибора

| 1 градус равен 0,25 деления милливольтметра

| :param value\_milli\_voltmeter: это то количество вольт,которое нужно отобразить на приборе

| :return:

|

| value(self)

| :return: Возвращает текущее показание милливольтметра

**class svg\_widget\_galvanometer**(builtins.object)

| Свг-виджет, прибор гальванометр.

| Для использования в форме необходимо:

| 1) создать layout с размером и желаемым расположением прибора

| 2) добавить в этот layout поле self.svg\_widget экземпляра данного класса

| 3) всё готово!

|

| Methods defined here:

|

| \_\_init\_\_(self)

| Initialize self. See help(type(self)) for accurate signature.

|

| create\_widget\_galvanometer(self)

| Создаёт свг\_виджет, устанавливает стрелку прибора на 0 градусов,

| тобишь на 0 вольт

| :return: готовый к интеграции виджет

|

| update\_svg\_galvanometer(self, value\_galvanometer)

| Обновляет показания прибора,путём изменения угла наклона стрелки прибора

| 1 градус равен 0,862 деления гальванометра

| :param value\_galvanometer: это то, какое значение нужно отобразить на приборе

| :return:

|

| value(self)

| :return: Возвращает текущее показание гальванометра