**Задание 1**

**Цель задания**

В результате выполнения этого задания вы научитесь создавать собственные слоты.

**Инструкция к заданию**

1. Создайте новый проект — приложение Qt Widgets.
2. В конструкторе класса внедрите в интерфейс компоненты.
3. Объявите в классе и реализуйте собственной слот.

**Задание**

Реализуйте приложение “сумматор” позволяющее складывать два числа. Обратите внимание, что слот, реагирующий на нажатие кнопки должен подключаться к сигналу при помощи connect().

**Критерии зачёта**

1. Программа выполняет вычисления безошибочно.
2. Интерфейс приложения должен быть создан программно в конструкторе главного класса.
3. Слот подключен с помощью connect().

**Задание 2**

**Цель задания**

В результате выполнения этого задания вы научитесь взаимодействовать с классом QTimer.

**Инструкция к заданию**

1. Создайте новый проект — приложение Qt Widgets.
2. Добавьте виджеты на форму главного окна.
3. Реализуйте логику изменения интерфейса.

**Задание**

Вам предстоит создать простое приложение “Часы”.

Создайте форму, которая включает:

* 1 лейбл;

В конструкторе главного класса приложения подключите сигнал таймера со слотом изменения времени в лейбле (формат чч:мм:сс). Реализуйте логику слота.

Один из вариантов получения времени – использование класса [QTime](https://doc.qt.io/qt-6/qtime.html). и его методов hour(), minute(), second().

Один из вариантов получения времени – использование класса QTime. Чтобы получить текущее время – воспользуйтесь методом (Ваша переменная) = QTime::currentTime(); . Чтобы получить время из этого объекта – используйте методы hour(), minute(), second().

**Критерии зачёта**

1. Формат времени совпадает с указанным в задании.
2. Работа приложения соответствует описанию задания.

**Задание 3**

**Цель задания**

В результате выполнения этого задания вы укрепите свои навыки работы с connect() и disconnect().

**Инструкция к заданию**

1. Создайте копию проекта “Часы”.
2. Добавьте дополнительные виджеты на форму главного окна.
3. Реализуйте логику работы программы.

**Задание**

Добавьте в приложение возможность запуска таймера по заданному времени. Выбор реализации ввода времени остается за выполняющим задание. Главный аспект работы программы в демонстрации оставшегося времени после запуска таймера и выполнения цветового сигнала по истечению времени (например, мигание заднего фона в течение 5 секунд).

**Критерии зачёта**

1. При не запущенном таймере приложение должно демонстрировать время.
2. Присутствует возможность ввести время для таймера через интерфейс программы.
3. По истечению времени таймера производится цветовой сигнал.

**Задание 4**

**Задание**

Реализуйте приложение “целочисленный калькулятор”. За основу возьмите интерфейс любого простого калькулятора, знакомого вам. Приложение должно поддерживать неограниченное выражение в формате “1+2-5\*12/2”.

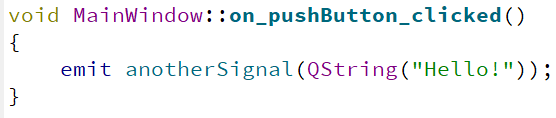
Для отображения выражения используйте текстовое поле (НЕ поле ввода).

Создайте собственный сигнал для добавления символа в строку и подключите к нему соответствующий слот. При нажатии на кнопку для добавления символа, вызывайте сигнал с переданным аргументом (символом). Таким образом, учитывая слот для кнопки подсчёта выражения, у вас должно получиться всего 2 слота на всю программу. Для работы программы достаточно одного собственного сигнала и нескольких сигналов clicked().

Пример вызова сигнала с аргументом при срабатывании сигнала без аргумента выглядит следующим образом:



Альтернативный пример, который работает без переопределения методов pushButton:



Также, существует следующий пример передачи сигнала в слот :



В данном случае, в качестве слота используется лямбда-выражение. Так как и сигнал и слот имею равное количество аргументов, то для компилятора это не является ошибкой.

**Критерии зачёта**

1. Программа выполняет вычисления безошибочно.
2. В программе существует всего 2 слота.