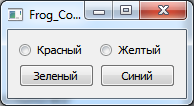
## Задание 1. Выбор цвета

Создайте приложение для выбора цвета фона формы. **При решении 1 используйте компоновки QHBoxLayout и QVBoxLayout.**

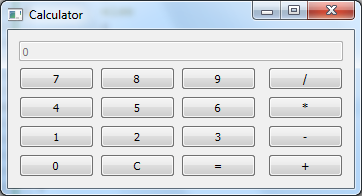


Подсказка:

* Для создания radiobutton :”Красный” и “Желтый” используйте класс QRadioButton;
* Конструктор QRadioButton::QRadioButton (const QString & text, QWidget\* parent = 0);
* Не забудьте #include <QRadioButton>;
* При нажатии на QRadioButton генерируется сигнал clicked() (как и на QPushButton);

## Задание 2. Приложение «Калькулятор»

Реализуем приложение калькулятор следующего вида:



* Используем компоновку QGridLayout;
* Работаем с целыми положительными числами. Число может быть не только однозначным☺;
* Помимо основной функциональности, должна работать следующая форма тестовой проверки: x+y-z\*r/… (символ ‘=’ не используется)
* При нажатии кнопки вызывается сигнал clicked();
* Необходимо связать сигнал кнопки с требуемым слотом. Достаточно трех слотов. Первый – принимает сигнал нажатия цифры, второй – сигнал нажатия кнопки с символом, третий – сигнал клавиши очистки памяти.
* Для того, чтобы узнать, какая кнопка нажата, можно узнать текст на ней следующим способом:

{

QPushButton\* pB = qobject\_cast<QPushButton\*> ( sender() );

QString str = pB->text();

…

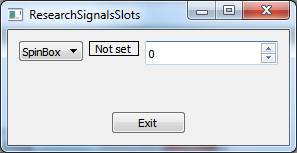
}

* Удобным представляется создать массив из трех элементов QString. В первом элементе хранить первое число, во втором – действие, в третьем – второе число.

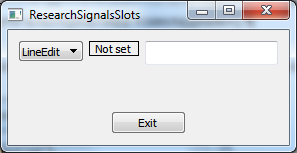
## Задание 3. Стековая компоновка - QStackedLayout

Стековая компоновка используется для таких задач, в которых требуется, чтобы в каждый момент времени видимым был только один из виджетов, включенных в такую компоновку (это альтернатива «закладкам»).

Создайте стековую компоновку следующим образом:







### Схема расположения компоновок и виджетов:

edit

spin

slider

Стековая компоновка

Горизонтальная компоновка

button

ComboBox

Label

Вертикальная компоновка

Подсказки:

1. Для «наполнения» combobox строками пользуйтесь методом  
   addItem(QString & text);
2. При выборе пользователем пункта из combobox генерируется сигнал activated(int);
3. В классе QStackedLayout для «переключения» компонент (в нашем случае - три виджета) предусмотрен слот - setCurrentIndex(int);
4. Чтобы при выборе пользователем пункта в combobox в стековой компоновке происходило «переключение», следует соединить эти сигнал и слот;
5. При создании горизонтальной компоновки, в которую входит combobox, высота компоновки определяется размером combobox с учетом выпадающего списка => это стоит учесть при формировании горизонтальной компоновки и «прижать» combobox и label кверху посредством третьего параметра метода QHBoxLayout::addWidget(…,…,Qt::Alignment);
6. Компоновки добавляются в другие компоновки с помощью метода addLayout()

## «Пружинка»

Если требуется «подпереть» элемент, чтобы при увеличении размера окна виджет не увеличивался пропорционально растяжению, можно в компоновку добавить «пружинку» (объект QSpacerItem).

* Создаем объект QSpacerItem
* Добавляем его в компоновку с помощью метода addItem()