

## 1

Окно приложения содержит 2 LCD-индикатора (QLCDNumber), список, комбинированный список, 2 бегунка, 2 текстовых редактора, 2 кнопки нажатия. При вводе текста в первый текстовый редактор и нажатии клавиши <Enter> - в список добавляется новый элемент с текстом из первого редактора, при вводе текста во второй текстовый редактор и нажатии клавиши <Enter> - в комбинированный список добавляется новый элемент с текстом из второго редактора. При нажатии на первую кнопку нажатия – из списка удаляется элемент, при нажатии на вторую кнопку нажатия – из комбинированного списка удаляется элемент. Номера удаляемых элементов задаются с помощью бегунков. Индикаторы отображают позицию бегунков.

## 2

Окно приложения содержит LCD-индикатор (QLCDNumber), список, бегунок, надпись, кнопку нажатия. С помощью бегунка можно добавлять и удалять элементы из списка (число элементов равно позиции бегунка). Элементы списка являются **случайными** целыми числами. При нажатии на кнопку надпись должна отображать все элементы списка (через “\n”). LCD-индикатор должен отображать текущее количество элементов в списке.

## 3

Окно приложения содержит кнопку нажатия, бегунок, два списка, надпись, LCD-индикатор (QLCDNumber). Количество элементов списков и их содержимое – произвольное (при этом списки должны для наглядности быть как минимум наполовину заполнены). При нажатии на кнопку запускается/останавливается следующий процесс: по событию таймера элементы (по одному) из одного списка перемещаются в другой (и наоборот – по очереди). Элементы выбираются случайно. Бегунок отвечает за интервал таймера (в секундах). Надпись должна быть расположена рядом со списками (над) и иметь текст “Осторожно – идет расчет!”. LCD-индикатор должен отображать значение последнего элемента перемещенного из одного списка в другой.

P.S.

Каждый класс, унаследованный от QObject содержит свои собственные таймеры. Метод int startTimer(int interval) класса QObject производит запуск таймера и возвращает идентификатор необходимый для распознавания таймеров используемых в объекте (таймеров в общем случае может быть много, и у каждого свой интервал времени генерации события). Если не удастся создать таймер, возвращаемое значение будет равно нулю. Параметр interval задает интервал времени в миллисекундах, по истечении которого будет генерироваться событие QTimerEvent. Метод void killTimer(int id) класса QObject уничтожает таймер с идентификатором id, метод void killTimers() класса QObject уничтожает все таймеры относящиеся к данному объекту.

Для обработки события от таймера QTimerEvent необходимо перезаписать **защищенный виртуальный** метод timerEvent(QTimerEvent\* e). В этом методе необходимо разместить обработку данного события (работа со списками). Метод класса QTimerEvent int timerId() возвращает идентификационный номер таймера. В данном случае будет создаваться 1 таймер, поэтому параметр метода timerEvent можно вообще не использовать.

## 4

Окно приложения содержит LCD-индикатор (QLCDNumber), список, комбинированный список, бегунок, кнопку нажатия. Для комбинированного списка необходимо обеспечить возможность редактирования (добавления новых элементов). При нажатии на кнопку нажатия все элементы из комбинированного списка перемещаются в “обычный” список. Индикатор должен отображать число элементов в списке. Бегунок задает максимальное число элементов комбинированного списка, а индикатор должен отображать это число.