

Lista de Exercícios no. 5 - Desenvolvimento de GUI :
Prof. Tiago Garcia de Senna Carneiro

Aluno: Lívia Stéffanny de Sousa

Matricula: 20.1.4029

Questões Teóricas

- 1- a) Uma call-back function é um código executável que é passado como argumento para outro executável, que deverá executar esse argumento em algum momento ou sob determinada condição. É útil quando se precisa escrever uma função que deve executar várias funções em algum momento ou quando se precisa executar uma função que é especificada apenas pelo caller.
b) Chamadas assíncronas de callback functions não são muito legíveis e mexer com ponteiros de funções frequentemente leva a erros irritantes.
- 2 - a) Um signal é um sinal emitido por widgets do Qt para indicar que uma ação de usuário ou mudança de estado aconteceu.
b) Slots são quase idênticos às funções comuns de C++. Podem ser sobrecarregadas, podem ser virtuais, podem ser públicas, protegidas e privadas, podem ser diretamente invocadas como qualquer função de C++ e seus parâmetros podem ser de qualquer tipo. A diferença é que um slot também pode ser conectado a um signal, onde será automaticamente invocado quando o signal for emitido.
c) Esse mecanismo possibilita ao programador ligar objetos sem que os objetos saibam qualquer coisa um sobre o outro. De forma automática ou manual, signals serão emitidos quando alguma ação for executada ou uma mudança de estado de objeto acontecer, tais signals, conectados a um slot, executarão automaticamente uma função previamente definida.
d) Deixa o código muito mais legível e organizado, não é necessário manipular ponteiros de função, facilita tratamento de eventos do mouse e teclado (facilita a programação assíncrona), dentre outros.
- 3 - a) Qualquer método pode emitir um signal de sua classe, seja ele próprio ou herdado.
b) Um signal pode ser conectado a vários slots, não há limites.
c) Os slots são executados um após o outro, em uma ordem não específica.

- 4) De certa forma, sim. Ao invés de termos objetos trocando mensagens com outros, temos que um evento dispara um método de um objeto, quando o estado de outro objeto é modificado, sem que eles tenham qualquer conhecimento um do outro.
- 5) Primeiro é necessário usar o Meta-Object Compiler, um programa que lida com extensões de C++ do Qt. O moc lê um arquivo de cabeçalho C++, se ele encontra uma ou mais declarações de classes que contenham a macro `Q_OBJECT`, ele produz um arquivo C++ contendo o código meta-object dessas classes. Dentre outras coisas, o código de meta-objeto é necessário para o mecanismo de signals e slots, a informação de tipo em tempo de execução e o sistema de propriedades dinâmicas. O código C++ gerado pelo moc deve ser compilado e linkado com a implementação da classe. A utilização do qmake automaticamente executa o moc.

Questões Práticas

- 1) O botão aumenta de tamanho junto com a tela, sem aumentar o tamanho do texto.
- 2) Com o signal `clicked()` o slot é executado quando o botão do mouse é pressionado e liberado sobre o objeto, com o signal `pressed()` o slot é executado quando o botão do mouse é pressionado, antes mesmo de acontecer a liberação do botão.
- 3) Tanto o tamanho do botão quando sua posição (absoluta) na tela se mantém inalteradas quando o tamanho da janela é modificado.
- 4) Nada foi alterado. O construtor da classe atribui um tamanho fixo para o widget que ele implementa, sendo assim, qualquer tentativa de redimensioná-lo será em vão.
- 5) Duas janelas idênticas foram criadas, e pressionar o botão de qualquer delas encerra a aplicação.
- 6) Sim. Apesar da aplicação permanecer funcional quando se tenta atribuir valor inicial fora do intervalo, ela deixa de funcionar se o número de dígitos no visor não é capaz de representar o valor da barra de deslizamento.
- 7) O valor corrente é exibido no próprio widget, sem necessidade de ter outro widget para exibí-lo. Também é possível digitar o valor exato desejado ao invés de encontrar seu ponto na barra de deslizamento. Foi preciso mudar o include, o tipo do objeto e a chamada do construtor. Não houve necessidade de alterar os métodos que definem os atributos do widget.
- 8) Foi preciso apenas acrescentar uma linha de código que conecta o signal `overflow()` da classe `QLCDNumber` ao slot `quit()` da aplicação.
- 9) (ver código)
- 10) (ver arquivos)