



Lista 5

Aluno: Nathann Zini dos Reis

Matrícula: 19.2.4007

Questão 1 -

A chamada de retorno é uma função passada para outras como argumento que será executado no programa posteriormente. Resumidamente, uma função callback é um código passado como parâmetro para outro código. Assim, a camada de software de nível mais baixo pode invocar funções de nível mais alto.

```
float someFunc(int a, int b, float (*func)(int, int));
```

Questão 2-

Um signal é uma mensagem que está presente em uma classe como declaração de uma função membro void. Signals não são invocados, mas emitidos por um objeto de classe. Ou seja, são métodos de classes que sinalizam a mudança do estado interno de um objeto.

Slots são métodos de classes que são disparados quando um signal é emitido.

O conceito é que widgets de interfaces gráficas podem enviar sinais contendo informações de eventos que podem ser recebidos por outras widgets / controles usando funções especiais conhecidas como slots. Isso é semelhante às callbacks em C/C++ (ponteiros de função), mas o sistema de sinais/slots garante que os argumentos da callback tenham os tipos corretos.

Signals e Slots implementam um mecanismo de comunicação entre objetos ou widgets.

Questão 3-

- a) Todas as classes que herdam de QObject ou uma de suas subclasses (por exemplo, QWidget) podem conter sinais e slots.
- b) Um signal pode estar associado a n slots. Um mesmo slot pode receber signals de objetos diferentes; e um mesmo signal pode enviar informações para vários slots.
- c) Nesse caso, o código após a emit palavra-chave continuará imediatamente e os slots serão executados mais tarde. Se vários slots estiverem conectados a um sinal, os slots serão executados um após o outro, na ordem em que foram conectados, quando o sinal for emitido.

Questão 5-

```
class MyClass : public QObject {
    Q_OBJECT
public:
    MyClass(QObject *parent = 0);
    ~MyClass();
signals:
    void mySignal();
public slots:
    void mySlot();
};

//-----MAIN-----//

myClass x, y;
connect(&x, SIGNAL(mySignal()), &y, SLOT(mySlot()));
x.mySlot();
```

Práticas

Questão 1-

O botão acompanhou o tamanho da janela.

Questão 3-

Nesse caso, o botão não seguiu o tamanho da tela, mas continuou no tamanho fixo.

Questão 4-

Não foi possível alterar o tamanho da janela.

Questão 5-

Duas janelas iguais foram abertas durante a execução.

```
int main(int argc, char *argv[]){
    QApplication app(argc, argv);
    MyWidget widget;
    MyWidget widget2;
    widget.show();
    widget2.show();
    return app.exec();
}
```

Questão 6-

A interface não funciona mais quando o número de dígitos do Visor LCD é menor que a quantidade de dígitos do número da faixa da Barra de Rolagem. Por exemplo:

```
QLCDNumber *lcd = new QLCDNumber(3);
lcd->setSegmentStyle(QLCDNumber::Filled);

QSlider *slider = new QSlider(Qt::Horizontal);
slider->setRange(0, 9999);
slider->setValue(0);
```

Questão 7-

Com o QSpinBox, por mais que o QLCDNumber tenha o valor de dígitos menor que a quantidade de dígitos do QSpinBox, a interface ainda funciona, mesmo que o Visor LCD esteja limitado a mostrar o número de dígitos declarado.

Questão 9-

```
//Linhas de código adicionadas:  
srand(time(0));  
QRandomGenerator valueRandom = QRandomGenerator::global()->bounded(0,99);  
//Linha de código alterada:  
slider->setValue(valueRandom.generate());
```