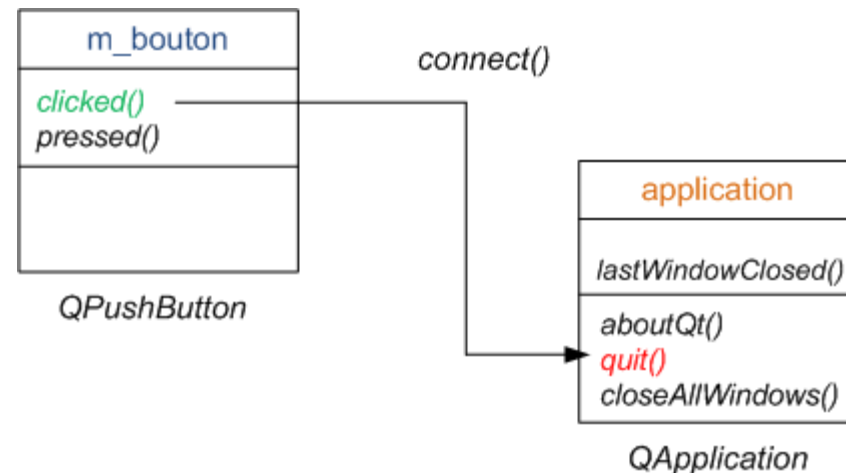




Connexion  
par le code

# Connexion entre un signal et un slot

- Nous voulons connecter le signal « bouton cliqué » au slot « quitter l'application ». Ainsi, un clic sur le bouton provoquerait l'arrêt de l'application.
- Pour cela, on utilise une méthode statique de QObject :: connect().



# Méthode connect()

- connect() est une méthode statique.

*Rappel : une méthode statique est une méthode d'une classe que l'on peut appeler sans créer d'objet. C'est en fait exactement comme une fonction classique.*

- Pour appeler une méthode statique, il faut faire précéder son intitulé du nom de la classe dans laquelle elle est déclarée. Comme connect() appartient à la classe QObject, il faut donc écrire :

`QObject::connect();`

# Méthode connect()

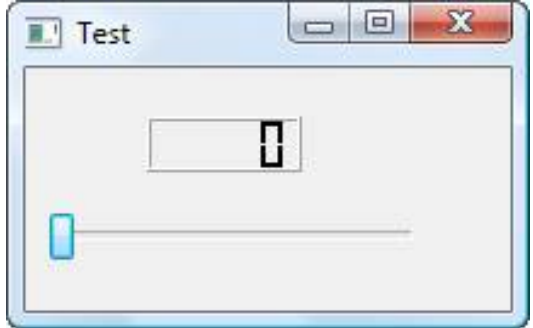
- La méthode connect() prend 4 arguments :
  - un pointeur vers l'objet qui émet le signal ;
  - le nom du signal que l'on souhaite « intercepter » ;
  - un pointeur vers l'objet qui contient le slot récepteur ;
  - le nom du slot qui doit s'exécuter lorsque le signal se produit.
- Il existe aussi une méthode disconnect() permettant de casser la connexion entre deux objets

# Exemple

```
21     m_bouton = new QPushButton("Quitter", this);  
22     m_bouton->move(110, 50);  
23  
24     // Connexion du clic du bouton à la fermeture de l'application  
25     QObject::connect(m_bouton, SIGNAL(clicked()), qApp, SLOT(quit()));
```

- **m\_bouton** : c'est un pointeur vers le bouton qui va émettre le signal.
- **SIGNAL(clicked())** : nom du signal. SIGNAL() est une macro du préprocesseur.
- **qApp** : c'est un pointeur vers l'objet de type QApplication . Qt crée automatiquement un pointeur appelé qApp vers l'objet de type QApplication que nous avons créé. Ce pointeur est défini dans le header <QApplication> .
- **SLOT(quit())** : c'est le slot qui doit être appelé lorsqu'on a cliqué sur le bouton. Il faut utiliser la macro SLOT() .

# Passage de paramètre entre signal et slot

- Les signaux et les slots peuvent s'échanger des paramètres.
  - Exemple entre deux widgets `QSlider`(curseur) et `QLCDNumber`(afficheur)
- 
- On dispose du signal et du slot suivants :
    - Le signal `valueChanged(int)` du `QSlider` : il est émis dès que l'on change la valeur du curseur du slider en le déplaçant. La particularité de ce signal est qu'il envoie un paramètre de type `int` (la nouvelle valeur du slider).
    - Le slot `display(int)` du `QLCDNumber` : il affiche la valeur qui lui est passée en paramètre.

- La connexion se fait avec le code suivant :

```
QObject::connect(m_slider, SIGNAL(valueChanged(int)), m_lcd, SLOT(display(int)));
```

- Il suffit d'indiquer le type du paramètre envoyé, ici un int, sans donner de nom à ce paramètre. Qt fait automatiquement la connexion entre le signal et le slot et « transmet » le paramètre au slot.
- Le type des paramètres doit absolument correspondre. On ne peut pas connecter un signal qui envoie (int, double) à un slot qui reçoit (int, int). C'est un des avantages du mécanisme des signaux et des slots : il respecte le type des paramètres.
- Un signal peut envoyer plus de paramètres à un slot que celui-ci ne peut en recevoir. Dans ce cas, les paramètres supplémentaires seront ignorés.