

2029-事件过滤处理分析--零声教育

一、Qt 事件过滤器

专门用于拦截传递到目标对象的事件，则实现监控目标对象事件作用。

通过事件过滤器可以监控其他组件接收到的相关事件，QObject 对象都可以作为事件过滤来应用。事件过滤器在组件之前接收到事件，才能够决定是否将事件转发至某个组件对象。

常用 API 函数如下：

1、安装事件过滤器

void QObject::installEventFilter(QObject *filterObj)

Installs an event filter *filterObj* on this object. For example:

```
monitoredObj->installEventFilter(filterObj);
```

An event filter is an object that receives all events that are sent to this object. The filter can either stop the event or forward it to this object. The event filter *filterObj* receives events via its `eventFilter()` function. The `eventFilter()` function must return true if the event should be filtered, (i.e. stopped); otherwise it must return false.

If multiple event filters are installed on a single object, the filter that was installed last is activated first.

Here's a `KeyPressEater` class that eats the key presses of its monitored objects:

2、事件监听

[virtual] bool QObject::eventFilter(QObject *watched, QEvent *event)

Filters events if this object has been installed as an event filter for the *watched* object.

In your reimplementation of this function, if you want to filter the *event* out, i.e. stop it being handled further, return true; otherwise return false.

Example:

```
class MainWindow : public QMainWindow
{
public:
    MainWindow();

protected:
    bool eventFilter(QObject *obj, QEvent *ev) override;

private:
    QTextEdit *textEdit;
};
```

3、删除事件过滤器

void QObject::removeEventFilter(QObject *obj)

Removes an event filter object *obj* from this object. The request is ignored if such an event filter has not been installed.

All event filters for this object are automatically removed when this object is destroyed.

It is always safe to remove an event filter, even during event filter activation (i.e. from the `eventFilter()` function).

See also `installEventFilter()`, `eventFilter()`, and `event()`.

二、【项目模块实战：运行结果】

