QMultiMap+QSharedPointer

Введение

Библиотека Qt предоставляет множество удобных средств для разработки любых приложений, например api для взаимодействия с мобильными платформами или набор классов для GUI приложений. Сложно представить себе крупное приложение написанное без использования стандартных контейнеров (векторов, деревьев). В данной статье речь пойдет об одной не совсем очевидной особенности совместного использования таких классов как QMultiMap и QSharedPointer. В качестве среды используется Ubuntu 10.04 с установленным Qt 4.6 .

Проблема

Рассмотрим небольшой пример, который наглядно демонстрирует специфику совместного использования QMultiMap и QSharedPointer .

```
#include <QSharedPointer>
#include <QMultiMap>
#include <QList>
#include <cstdio>

int main(int argc,char** argv){
        QMultiMap<QSharedPointer<int>, QString> map;
        int* a,*b;
        a=new int;
        b=new int;
        QSharedPointer<int> aa(a);
        QSharedPointer<int> bb(b);
        map.insert(aa,"q");
        map.insert(bb,"qq");
        map.insert(bb,"qqq");
        QList<QSharedPointer<int> > unq=map.uniqueKeys();
        return 0;
}
```

Если мы попытаемся скомпилировать данный пример, то получим следующие ошибки:

```
In file included from /usr/include/qt4/QtCore/QMultiMap:1,
                 from main.cpp:2:
/usr/include/qt4/QtCore/qmap.h: In member function ‘QList<T> QMap<Key,
T>::uniqueKeys() const [with Key = QSharedPointer<int>, T = QString]’:
main.cpp:23:   instantiated from here
/usr/include/qt4/QtCore/qmap.h:782: error: no match for ‘operator<’ in ‘aKey
< i.QMap<Key, T>::const_iterator::key [with Key = QSharedPointer<int>, T =
QString]()’
/usr/include/qt4/QtCore/qchar.h:385: note: candidates are: bool
operator<(QChar, QChar)
```

```
/usr/include/qt4/QtCore/qbytearray.h:520: note:                 bool
operator<(const QByteArray&, const QByteArray&)
/usr/include/qt4/QtCore/qbytearray.h:522: note:                 bool
operator<(const QByteArray&, const char*)
/usr/include/qt4/QtCore/qbytearray.h:524: note:                 bool
operator<(const char*, const QByteArray&)
/usr/include/qt4/QtCore/qstring.h:927: note:                 bool
operator<(const char*, const QString&)
/usr/include/qt4/QtCore/qstring.h:940: note:                 bool
operator<(const char*, const QLatin1String&)
/usr/include/qt4/QtCore/qstring.h:953: note:                 bool
operator<(const QLatin1String&, const QLatin1String&)
/usr/include/qt4/QtCore/qstring.h:1181: note:                  bool
operator<(const QStringRef&, const QStringRef&)
/usr/include/qt4/QtCore/qmap.h: In function ‘bool qMapLessThanKey(const
Key&, const Key&) [with Key = QSharedPointer<int>]’:
/usr/include/qt4/QtCore/qmap.h:760:   instantiated from ‘QMapData::Node*
QMap<Key, T>::mutableFindNode(QMapData::Node**, const Key&) const [with Key
= QSharedPointer<int>, T = QString]’
/usr/include/qt4/QtCore/qmap.h:576:   instantiated from ‘QMap<Key,
T>::iterator QMap<Key, T>::insertMulti(const Key&, const T&) [with Key =
QSharedPointer<int>, T = QString]’
/usr/include/qt4/QtCore/qmap.h:953:   instantiated from ‘typename QMap<Key,
T>::iterator QMultiMap<Key, T>::insert(const Key&, const T&) [with Key =
QSharedPointer<int>, T = QString]’
main.cpp:20:   instantiated from here
/usr/include/qt4/QtCore/qmap.h:107: error: no match for ‘operator<’ in ‘key1
< key2’
/usr/include/qt4/QtCore/qchar.h:385: note: candidates are: bool
operator<(QChar, QChar)
/usr/include/qt4/QtCore/qbytearray.h:520: note:                 bool
operator<(const QByteArray&, const QByteArray&)
/usr/include/qt4/QtCore/qbytearray.h:522: note:                 bool
operator<(const QByteArray&, const char*)
/usr/include/qt4/QtCore/qbytearray.h:524: note:                 bool
operator<(const char*, const QByteArray&)
/usr/include/qt4/QtCore/qstring.h:927: note:                 bool
operator<(const char*, const QString&)
/usr/include/qt4/QtCore/qstring.h:940: note:                 bool
operator<(const char*, const QLatin1String&)
/usr/include/qt4/QtCore/qstring.h:953: note:                 bool
operator<(const QLatin1String&, const QLatin1String&)
/usr/include/qt4/QtCore/qstring.h:1181: note:                  bool
operator<(const QStringRef&, const QStringRef&)
make: *** [main.o] Ошибка 1
```

Обратим внимание на упоминаемую одной из первых функцию

```
‘QList<T> QMap<Key, T>::uniqueKeys() const [with Key = QSharedPointer<int>,
```

```
T = int]’
```

из-за которой и происходит сбой компиляции. Вот так выглядит ее прототип в документации Qt uniqueKeys :

```
QList<Key> QMap::uniqueKeys () const
```

Очевидно, что возвращаемые значения в обоих случаях не совпадают. Обратимся к описанию данной функции. Исходя из документации она возвращает список всех ключей в QMultiMap в порядке возрастания. Логично предположить, что основная проблема заключается в сортировке полученного списка ключей, ввиду отсутствия оператора сравнения определенного для типа QSharedPointer<T>. Определим свой оператор сравнения следующим образом:

```
template <class T>
bool operator <(const  QSharedPointer<T>& a, const  QSharedPointer<T>& b){
        return a.data()<b.data();
}
```

Если включить код данного оператора в код примера, то компиляция пройдет нормально и мы получим работающее приложение.

Проблема, похожая на описанную выше, возникает при замене QMultiMap на QMultiHash в примере. Для нормальной компиляции в таком случае необходимо определить собственную функцию qHash со следующим прототипом:

```
template <class T>
uint qHash(const QSharedPointer<T>& a);
```

Заключение

Описанная проблема устранена в версии Qt 4.7 которая входит в состав Qt SDK 1.1. Однако предложенное здесь решение остается актуальным, так как на момент написания статьи в OVI находится Qt 4.6.