CONTAINERS

(SEQ) QList<T>
QLinkedList<T>
QVector<T>

(ASS) QMap<key, T>
QHash<Key, T>

T → ASSIGNABLE DATA TYPE

BASIC TYPES

PUBLIC DEF CONST
COPY CONS
ASSIGN. OPERATOR

QObject → NO ASSIGNABLE DATA TYPE

QFile → QList<QFile*>

SEC 6.9

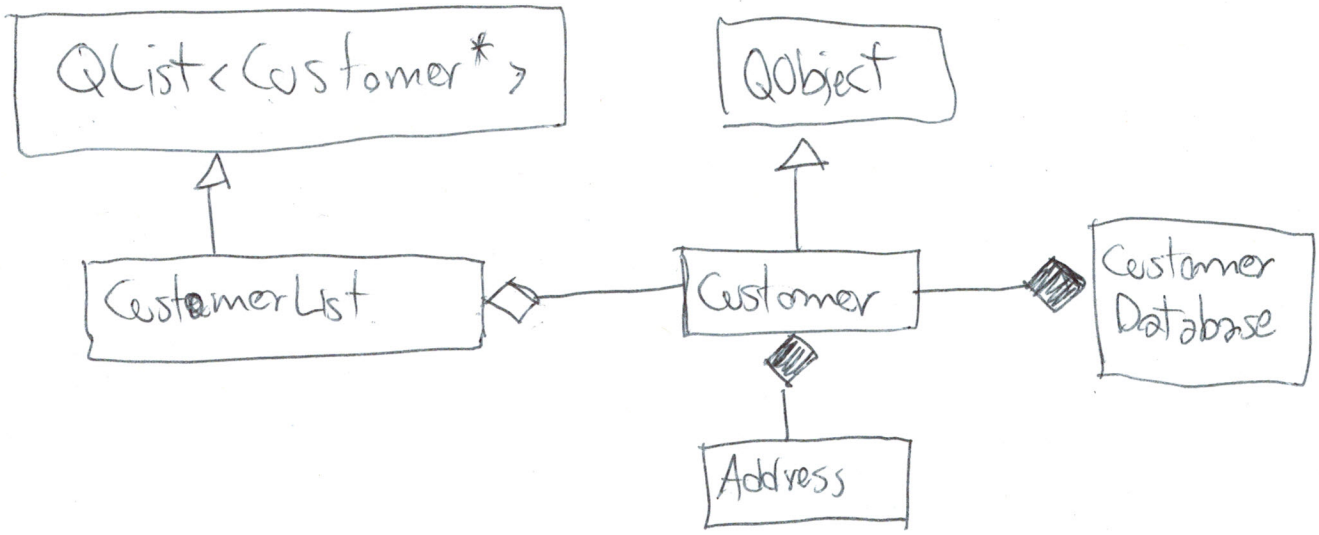
MANAGED CONTAINER

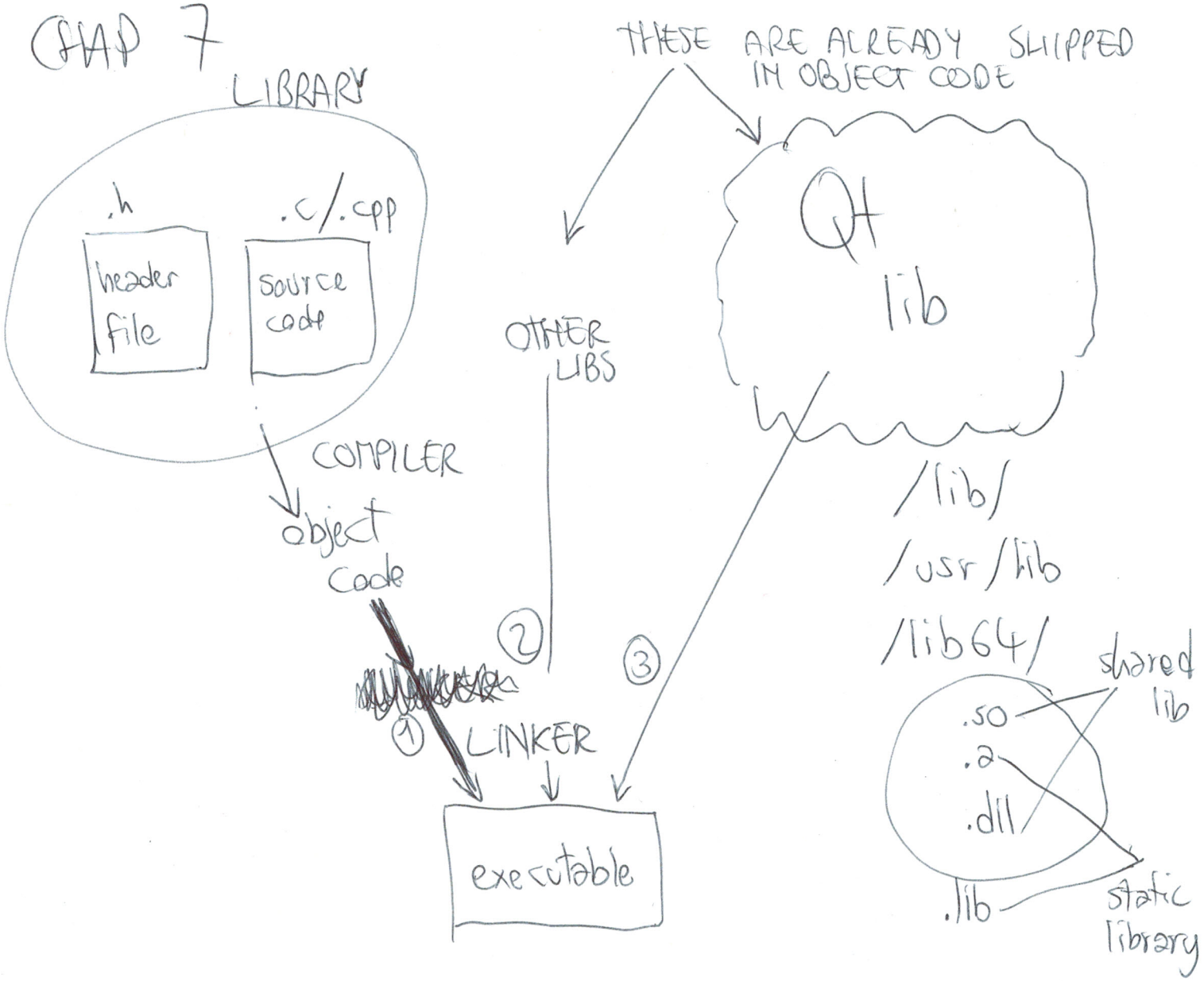
COMPOSITE RELATIONSHIP

DELETING qDeleteAll(..) ← container class

UNMANAGED CONTAINER

AGGREGATE RELA.




CHAP 7
LIBRARY
.h
header file
.c/.cpp
source code
COMPILER
object code
THESE ARE ALREADY SLIPPED IN OBJECT CODE
Qt
lib
OTHER LIBS
/lib/
/usr/lib
/lib64/
shared lib
.so
.a
.dll
.lib
static library
LINKER
executable
1
2
3

FORWARD DECLARATION (p240)

```
#include "classb.h"

class ClassC;

class ClassA : public ClassB {
public:
        ClassC* f1(something);
        ...
};
```

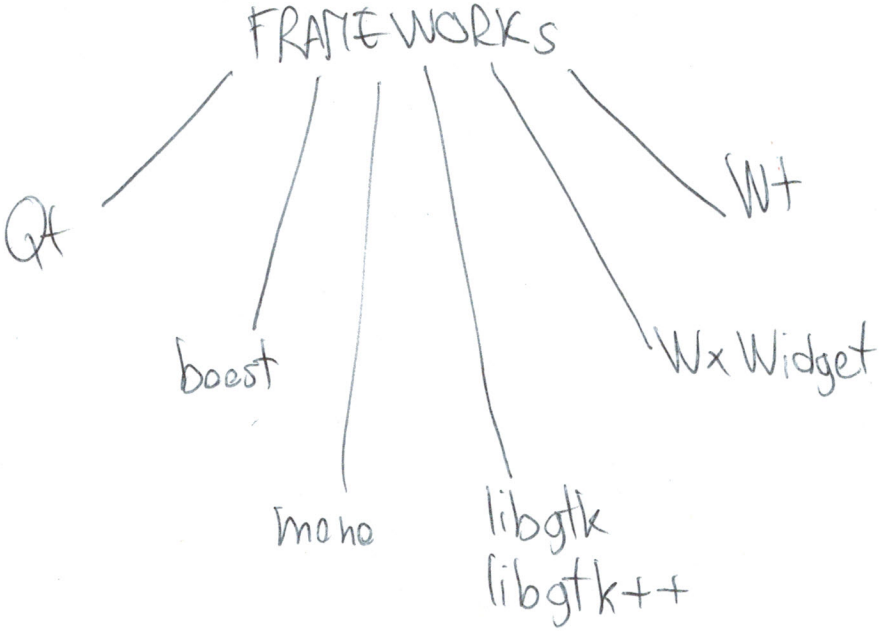
FORWARD DECL. of ClassC

---

static

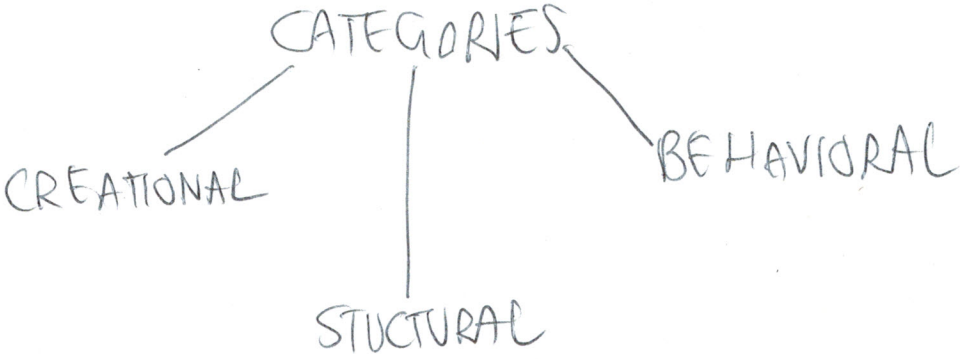
extern 

SEC 7.3



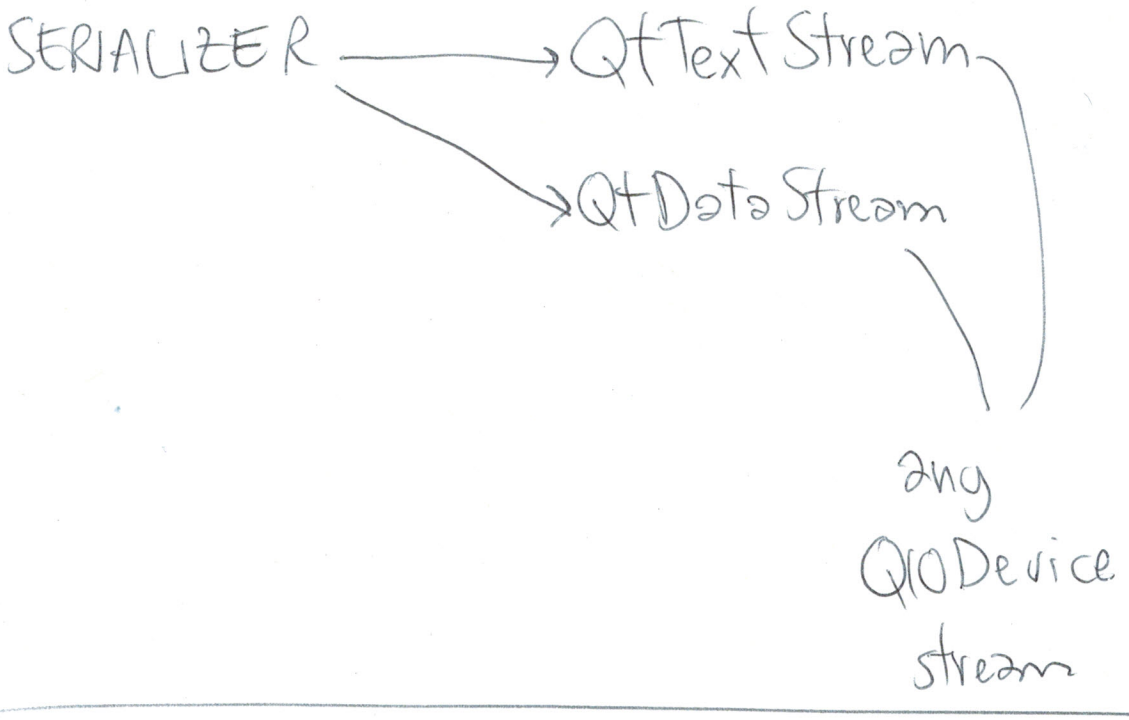
SEC 7.4

Gamma 1995 . 23 design pattern



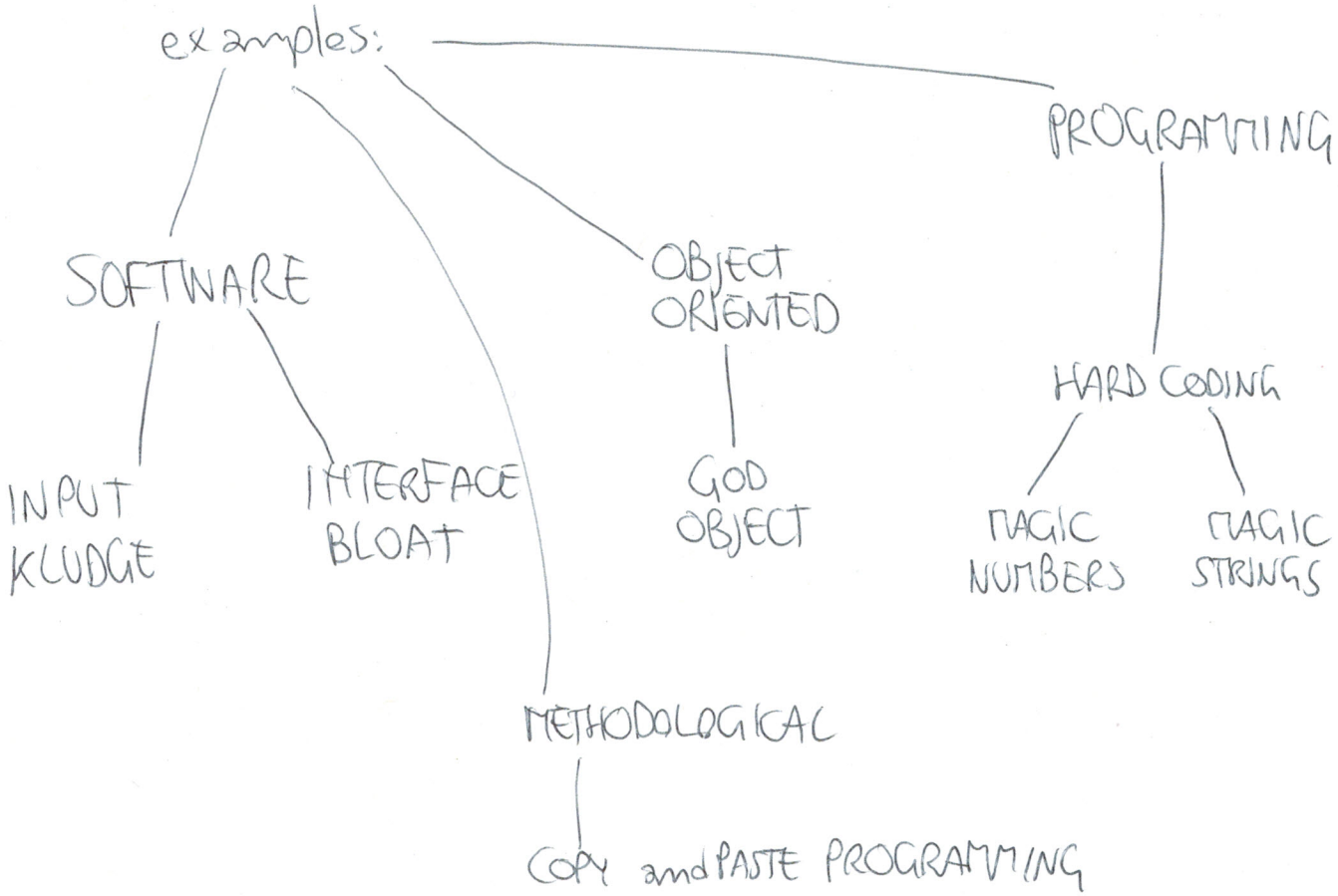
SEC 7.4.1

SERIALIZER

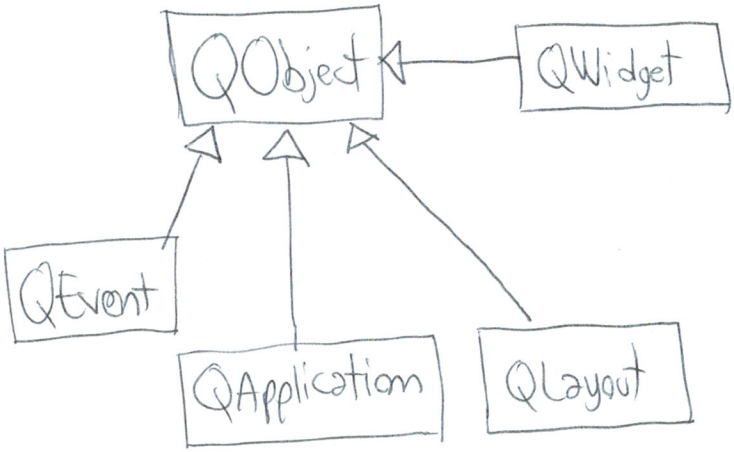


SEC 7.4.2 ANTI PATTERN

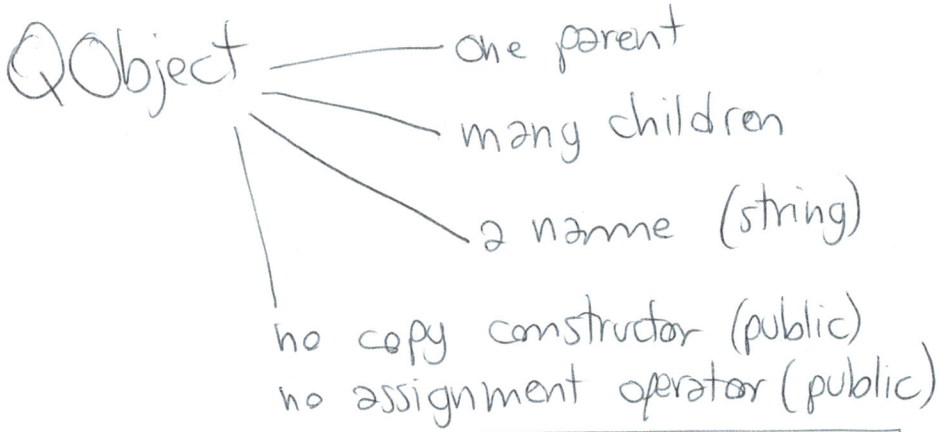
EXTENSIVE LIST on WIKIPEDIA



CHAP 8 QObject, QApplication



qtbase/src/corelib/kernel/qobject.h ← Qt SRC CODE



EACH QObject INSTANCE IS UNIQUE!

```
class QObject {
public:
    explicit QObject(QObject *parent=0);
    QObject* parent() const;
    QString objectName() const;
    void setParent(QObject *parent);
    const ObjectList& children() const;
```

THERE IS A RELATIONSHIP TREE BETWEEN OBJECTS!

OBJECT INSTANCES

WHICH IS THE DIFFERENCE BETWEEN PARENT/CHILDREN and BASE CLASS/SUBCLASS RELATION?

NOTE on page 261

RUNTIME RELA

COMPILE/DESIGN TIME RELA