Qt

İlker Korkmaz & Kaya Oğuz
CE 350

Qt?

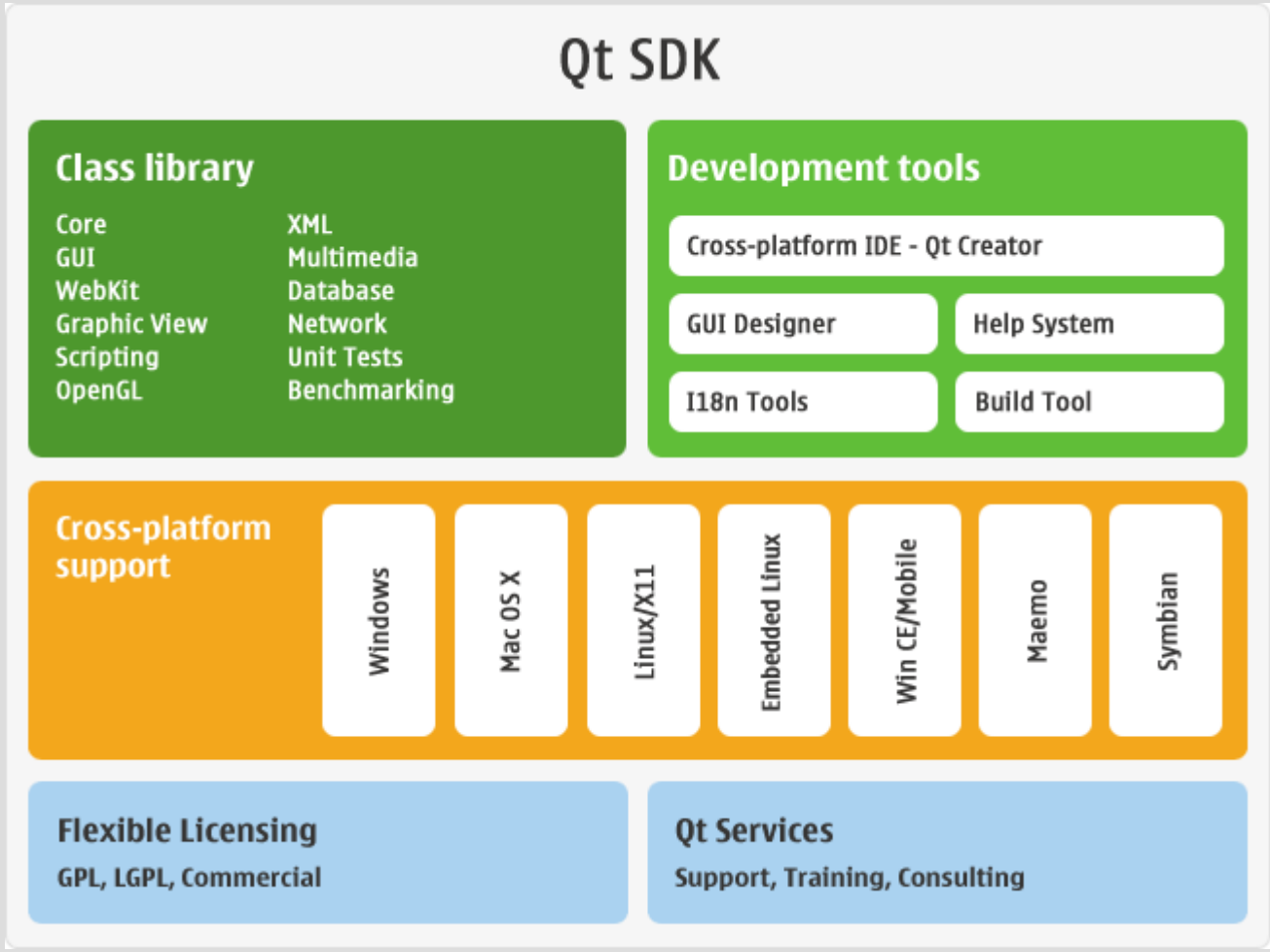
- Qt: Cute
- I'm a Qt programmer → cute programmer

- Write code once to target multiple platforms (Embedded Linux, Mac OSX, Windows, Linux/X11, Windows Mobile, Windows CE, Symbian, Maemo and MeeGo.)
- Do more with less and faster
- Use integrated developer tools
- Has great documentation

Who uses Qt?

- Nokia
- Opera web browser
- Skype Instant Messenger
- ASUS EeePC
- Samsung Digital Photo Frame

Qt SDK


Qt SDK
Class library
Core
GUI
WebKit
Graphic View
Scripting
OpenGL
XML
Multimedia
Database
Network
Unit Tests
Benchmarking
Development tools
Cross-platform IDE - Qt Creator
GUI Designer
Help System
I18n Tools
Build Tool
Cross-platform support
Windows
Mac OS X
Linux/X11
Embedded Linux
Win CE/Mobile
Maemo
Symbian
Flexible Licensing
GPL, LGPL, Commercial
Qt Services
Support, Training, Consulting

What do I need?

- You need to know C++ & Object Oriented Programming (there are bindings to other languages: Python, Java etc.)
- Download libraries (on Linux, they might be already there)
- Qt Designer and Qt Assistant may be used to create a simple application.

Before going further

- A GUI application runs in an infinite loop
- It waits for the user to interact with the application; click a button, or use a shortcut etc.
- These interactions are called events. GUI applications are event-driven.
- To handle events, you bind the event to a function.
- When there is click on a specific button, a certain function (of a class) has to be called.

Before going further (2)

- These events and bindings are handled by Qt's signal and slot system.
- It is very simple; any object of a class can emit a signal and this signal can be connected to a slot (function) of any object of any class.
- A signal can be connected to more than one slot.
- A slot can have many signals connected to it.

User Interface

- Use Qt Designer to add widgets.
- The designer saves the file with the extension ui, in XML format.
- When compiling, Qt uses uic (user interface compiler) program to convert this file to C++ code.
- To use this C++ code, you should inherit a new class from it and develop using that class; therefore, when there is a change to UI, your work will be saved.

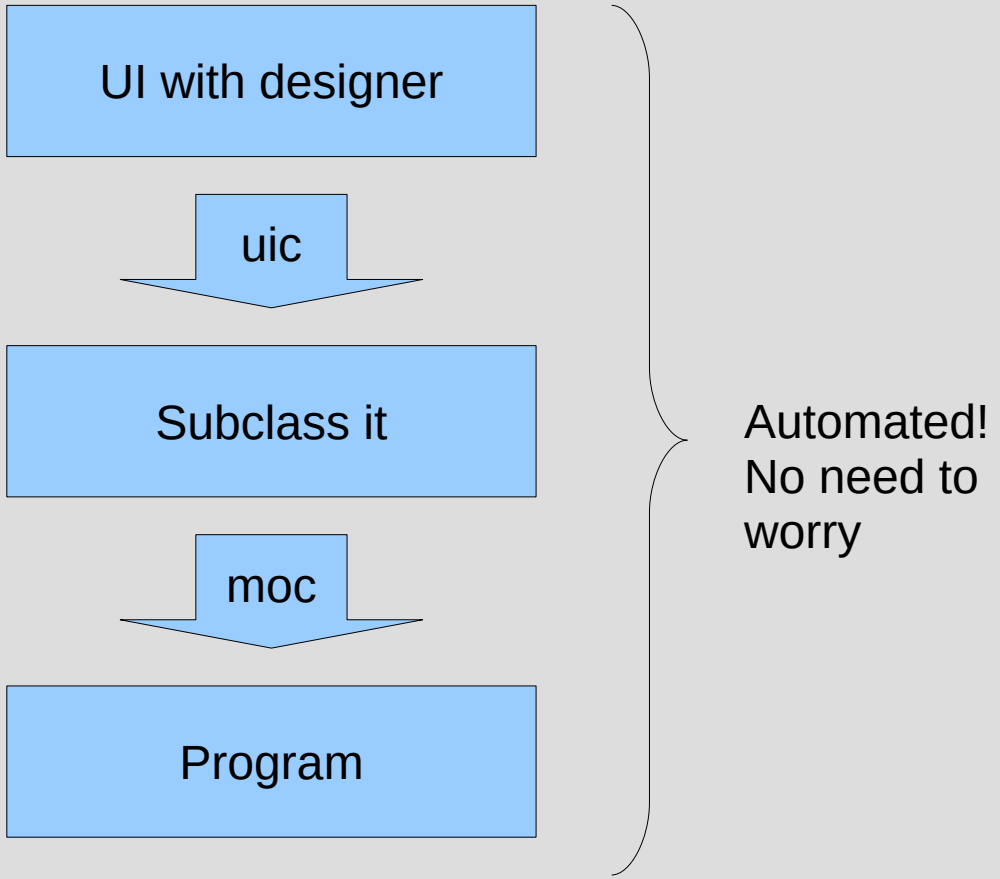
MOC

- To define signals and slots, you write:

```
Q_OBJECT
public slots:
```

- And define the slot as a regular C++ member function.
- However, C++ compiler would give an error to “public slots”.
- Qt uses moc (meta object compiler) to convert signals and slots to complex C++ code.

Here's a path:


UI with designer
uic
Subclass it
moc
Program
Automated!
No need to
worry

How is it automated?

- Makefiles!
- Once you have your ui and class files in place, run
  `qmake -project`
- This creates a .pro file, the project file.
- Then run
  `qmake`
- This parses the project file and creates a Makefile
- Now, run `make`. And you are done!

Makefiles?

- In the good old days, installing an application was almost always from the source code.
- You download the source code (source.tar.gz), unpack it, (tar -xzf source.tar.gz), cd into it, (cd source).
- The source might have had a “configure” script that would create a Makefile according to your system. (./configure)
- Without the configure, you may need to edit the Makefile yourself (in very very few cases).

Makefiles? (2)

- Makefiles are used with the make utility
- make is a utility that automatically builds executable programs and libraries from source code by reading files called makefiles which specify how to derive the target program (Thank you Wikipedia)
- With makefiles you define targets, source and header files, and automate the build process. It is very useful if you have many files.
- IDEs (like CodeBlocks) use make internally

Makefiles? (3)

- The next step installing is to compilation:
  `make`
- If make does not give any errors during compilation, then you have a working program. To install it system wide, run
  `make install`
  as super user. This will copy necessary files to system wide locations.

Makefiles? (4)

- With Qt, you have nothing to worry about.
- Qt creates makefiles and you don't have to manually edit them.
- Just type make, and the project will be compiled!

Sample App Code

```
#include <QMainWindow>
#include "ui_anapencere.h"

class anapencere:public QMainWindow, Ui::AnaPencere {
        Q_OBJECT
        public:

        anapencere():QMainWindow() {
                setupUi(this);
                connect(addButton, SIGNAL(clicked()), this,
SLOT(add()));
        }

        public slots:
        void add() {
            listWidget->addItem(lineEdit->text());
            lineEdit->clear();
        }
};
```