

---

## Programmeermethoden

MATLAB, Java, Qt, ...

week 14: 12–16 december 2011

`http://www.liacs.nl/home/kosters/pm/`

`www.mathworks.nl/products/matlab,`  
`java.sun.com/ en qt.nokia.com/`

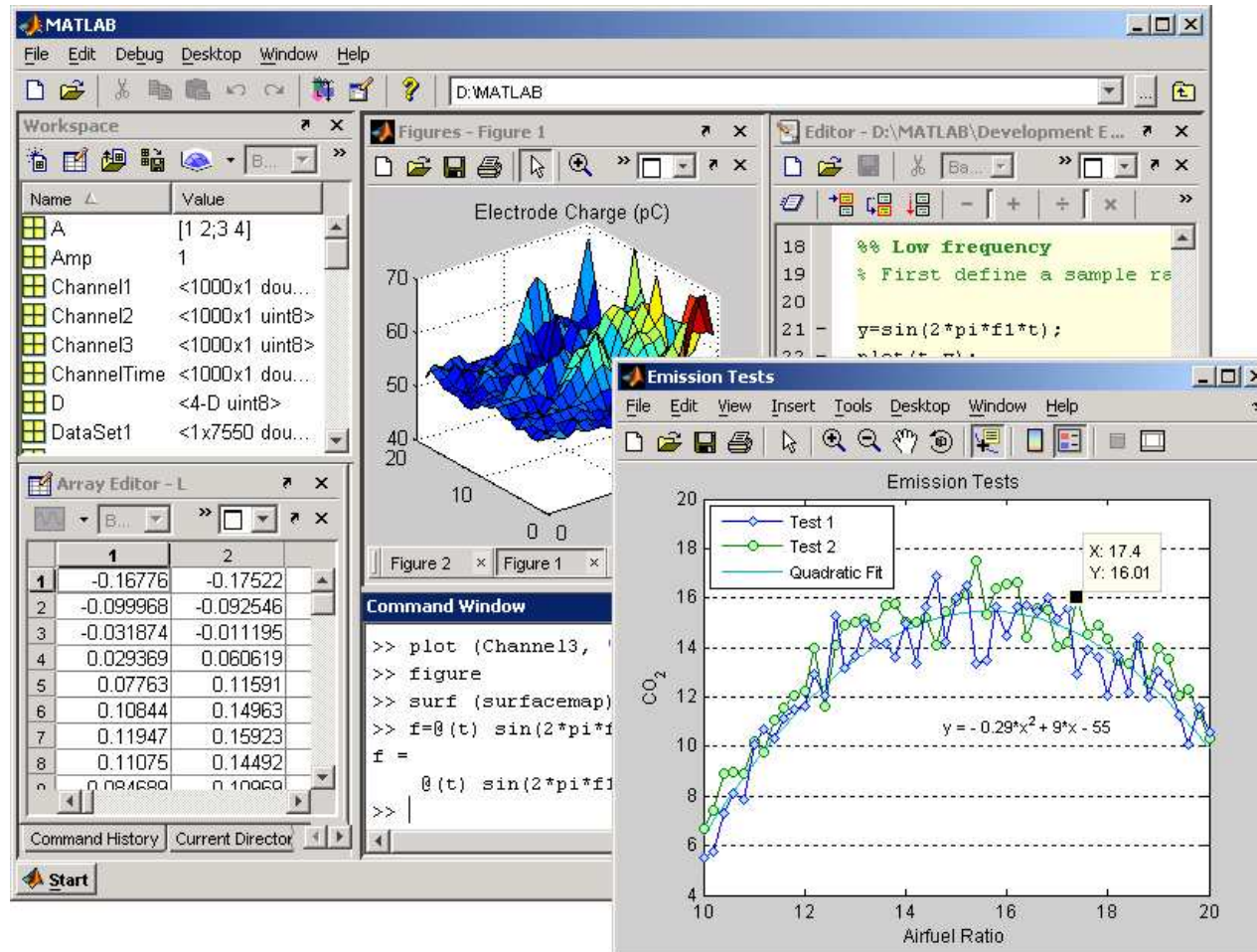
De laatste jaren wordt —naast C++— ook veel gebruik gemaakt van **Java**, **Python** en **MATLAB**. En **Qt** voor interfaces. We kijken straks kort naar Java en Qt.

MATLAB, zelf in C geschreven, is met name goed in allerlei wiskundige operaties: matrices, ...; de taal heeft ook veel grafische mogelijkheden: plotten, ...

Nadeel: MATLAB is duur.

Naast de workspace heb je de “command history” en de “command window”.

Programma's (scripts) hebben extensie `.m`.



MATLAB is een “weakly dynamically typed” taal. Na

```
>> A = 'abc';
```

is A een string, en na

```
>> A = [10 20 30; 7 8 9];
```

wordt A dan een  $2 \times 3$  array.

Als je de punt-komma aan het eind weglaat, krijg je geen uitvoer in het commando-venster voor deze regel. Dus:

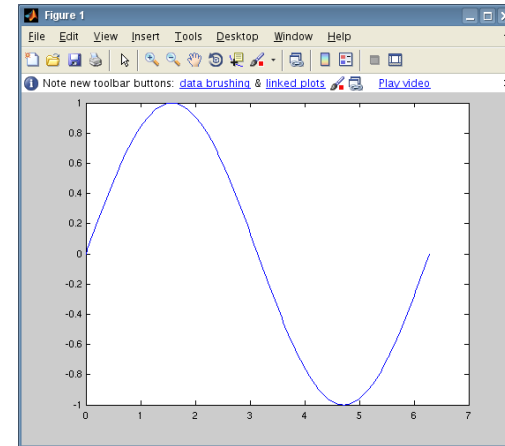
```
>> A(2,1)
```

```
ans =
```

```
7
```

Stel je wilt de sinus plotten:

```
>> x = 0:pi/100:2*pi;  
>> y = sin(x);  
>> plot(x,y)
```



Je kunt namelijk een vector (een 1-dimensionaal array) maken met

```
>> array = 1:2:9  
array =  
    1  3  5  7  9
```

waarbij “initialisatie:stapgrootte:terminator” gebruikt wordt.

De programmeertaal **Java** lijkt veel op C++.

Enkele belangrijke verschillen:

- Java is (nog) meer object-georiënteerd
- Java heeft automatische **garbage collection**
- Java heeft (bijna) geen pointers
- Java is platform-onafhankelijk
- Java kent **applets**: WWW-applicaties, met GUI's

Hoe compileer je een Java-programma, dat in `Iets.java` staat? Met `javac *.java` compileer je al je Java-files naar **bytecode**, in ons geval naar `Iets.class`. Deze kan met `java Iets` op elke computer gedraaid (om precies te zijn: *geïnterpreteerd*) worden met behulp van de **Java Virtual Machine (JVM)**. Dit bovenstaande geldt voor “tekst-applicaties”.

Een **applet** wordt bekeken met een webbrowser, en je moet dan een aparte HTML-pagina `naam.html` maken die de bytecode aanroept. Rechtstreeks (zonder browser) kan vaak ook eenvoudig met `appletviewer naam.html`, zie later. Overigens, Java is iets heel anders dan JavaScript.

Maak een file Hello.java met

```
public class Hello {  
    public static void main (String[ ] argv) {  
        System.out.println ("Hello world!");  
    }//main  
}//class Hello
```

En dan `javac Hello.java` en `java Hello`.

Let op: filenaam = naam public klasse (met hoofdletter).

Er kunnen meer klassen in een file staan, maar er is er maar één public, en bij diens `public static void main` begint de executie van het programma.

Met

```
int x = 10, y = 20;  
System.out.print (x + y);  
System.out.println (" of toch: " + x + y);
```

krijg je 30 of toch: 1020, gevolgd door een regelovergang, op het beeldscherm te zien,

Pas dus op met type-conversies.

En nu een programma `Fac.java`, met een wat ingewikkelder klasse met twee **methodes** (member-functies):

```
public class Fac {
    private static int facul (int n) {
        if ( n == 0 ) return 1;
        else return n * facul (n-1);
    }//facul
    public static void main (String[ ] argv) {
        int m = 5;
        System.out.println (m + "! is " + facul (m));
    }//main
}//class Fac
```

Let op het woordje `static` bij de functie-definitie: de functie mag dan aangeroepen worden zonder dat er een object van die klasse is.

En iets aangepast in Fac2.java:

```
public class Fac2 {
    private int facul (int n) {
        if ( n == 0 ) return 1;
        else return n * facul (n-1);
    }//facul
    public static void main (String[ ] argv) {
        int m = 5;
        Fac2 prog = new Fac2 ( );
        System.out.println (m + "! is " + prog.facul (m));
    }//main
}//class Fac2
```

Nu maken we met `new` een object `prog` van onze klasse `Fac2`, waarna we van dat object de methode `facul` (nu niet meer `static`) aanroepen. Er is geen `delete` nodig!

Variabelen worden in Java altijd “call-by-value” doorgegeven. In het bijzonder geldt dit voor `int`'s, `double`'s, `char`'s en `boolean`'s (in C++ `bool`'s geheten).

In geval van klassen wordt er eigenlijk een pointer doorgegeven — maar dat “weet niemand”. Het object waar naar gewezen wordt kan dus wel veranderen! De speciale waarde `null` wijst naar niks.

Er zijn automatische initialisaties.

Java kent uiteraard ook **arrays**. Deze moeten voor gebruik altijd ge-new-ed worden:

```
double A[ ] = new double[25]; // A[0],A[1],...,A[24]
int B[ ][ ] = new int[4][3]; // 2-dimensionaal array B
```

Let er op dat de notatie soms net iets van die van C++ verschilt. Zo mag `double [ ] A = new double[25];` ook.

In bovenstaand voorbeeld is `A.length` gelijk aan 25, de lengte van het array. Dus er is geen extra functie-parameter nodig om de array-lengte door te geven.

Een functie die een string omkeert:

```
public static String keerStringOm (String origineel) {
    char [ ] temp = origineel.toCharArray ( );
    // converteer originele string naar char-array
    // er wordt "vanzelf" ge-new-ed!
    for ( int i = 0; i < temp.length/2; i++ ) {
        char waarde = temp[i];
        temp[i] = temp[temp.length-i-1];
        temp[temp.length-i-1] = waarde; }//for
    return new String (temp);
} //keerStringOm
```

Deze functie geeft dus eigenlijk een pointer naar een nieuwe string terug. Geen zorgen om delete's!

Hoe gaan pointerlijsten dan in Java — je hebt toch geen pointers!?

In C++:

```
class Vakje { public: int info; Vakje* volgende; };  
...  
Vakje* p; p = new Vakje; p->info = 37;
```

In Java:

```
class Vakje { public int info; public Vakje volgende; }  
...  
Vakje p; p = new Vakje; p.info = 37;
```

Als er tijdens het draaien een fout optreedt, wordt er soms een **exception** gegenereerd. Die *kun* je “opvangen”:

```
try {  
    ...  
    i = doeiets (i);  
    A[i] = 42;  
    ...  
} catch (IndexOutOfBoundsException ex) {  
    System.out.println ("Foutje ... " + ex);  
} //catch
```

Als zo'n exception nergens (ook niet in aanroepende functies) wordt opgevangen, stopt het programma.

Input in Java is iets ingewikkelder georganiseerd:

```
import java.io.*;
// net als #include <iostream> in C++
try {
    String inputString; int waarde;
    BufferedReader BR; // net als cin in C++
    BR = new BufferedReader (
        new InputStreamReader (System.in));
    inputString = BR.readLine ( );
    waarde = Integer.parseInt (inputString);
    System.out.println (waarde * waarde);
    BR.close ( );
} catch (Exception ex) {
    System.out.println (ex); }//catch
```

Input uit een file gaat ongeveer net zo:

```
BR = new BufferedReader (  
    new FileReader ("mijnbestand.txt"));
```

Dit is bedoeld om char's mee in te lezen. Er zijn diverse varianten.

En een outputfile (een van de mogelijkheden):

```
PW = new PrintWriter (  
    new OutputStreamWriter(  
        new FileOutputStream ("eenfile.txt")));
```

In een internet-pagina kun je eenvoudig een Java-programma opnemen, met een fraaie grafische user interface (GUI). We maken dan een **applet**.

Eerst wat HTML, in (zeg) naam.html:

```
<HTML><TITLE>Test123</TITLE><BODY>
<APPLET CODE="MyApplet.class" WIDTH=400 HEIGHT=200>
  <PARAM NAME="buttonName" VALUE="Klik hier">
</APPLET>
</BODY></HTML>
```

Bekijken met `appletviewer naam.html` of via een browser.

Met **Swing** heb je talloze mogelijkheden . . .

```
import java.awt.*;          // "event-driven" programmeren
import java.awt.event.*;
import javax.swing.*;
import java.applet.*;

public class MyApplet extends JApplet implements ActionListener {
    private JButton myButton; // applet met een knop
    public void init ( ) {
        Container contentPane = getContentPane ( );
        contentPane.setLayout (null);
        myButton = new JButton (getParameter ("buttonName"));
        myButton.setBounds (10,10,100,50);
        myButton.addActionListener (this);
        contentPane.add (myButton);
    } //init
    public void actionPerformed (ActionEvent evt){
        Object source = evt.getSource ( );
        if ( source == myButton ) // waar komt event vandaan?
            myButton.setText ("Geklikt ...");
    } //actionPerformed
} //class MyApplet
```

```
http://www.liacs.nl/home/kosters/java/muis.html
http://www.liacs.nl/home/kosters/java/sleep.html
import java.awt.*; import java.applet.*; import java.awt.event.*;
public class Muis5 extends Applet
    implements MouseListener, MouseMotionListener {
    public int teller = 0;
    public void init ( ) {
        addMouseListener (this); addMouseMotionListener (this); }//init
    public void paint (Graphics g) {
        g.setColor(Color.white); g.fillRect(10,10,500,500); }//paint
    public void mouseClicked (MouseEvent event) { }//mouseClicked
    public void mouseReleased (MouseEvent event) { }//mouseReleased
    public void mousePressed (MouseEvent event) { }//mousePressed
    public void mouseEntered (MouseEvent event) { }//mouseEntered
    public void mouseExited (MouseEvent event) { }//mouseExited
    public void mouseDragged (MouseEvent event) {
        Graphics g = getGraphics ( ); teller++;
        switch ( teller % 4 ) {
            case 0: g.setColor (Color.red); break;
            case 1: g.setColor (Color.green); break;
            case 2: g.setColor (Color.black); break;
            case 3: g.setColor (Color.blue); break; }//switch
        g.drawLine (event.getX ( ),event.getY ( ),
            event.getX ( )+10,event.getY ( )+10); }//mouseDragged
    public void mouseMoved (MouseEvent event) { }//mouseMoved
}//Muis
```

Wat we hier —en straks ook in Qt— zien is dat de “control-flow” **event-driven** (event-gestuurd) is. En dat is anders dan wat we gewend zijn!

Een onzichtbare hoofdloop (de “event-lus”) loopt “einde-loos” door. Intussen worden, door **events** zoals muis-acties af te handelen, de gebruikerswensen vervuld. Dat zijn precies methoden (= memberfuncties) behorend bij bepaalde objecten.

Qt (“cute” of “ku-tee”; de ‘t’ is van toolkit) is een door Nokia (voorheen Trolltech) gemaakte “cross-platform ontwikkelomgeving”, met een ingebouwde C++-bibliotheek. Samenwerken met Java kan ook. Voor niet commercieel gebruik is het gratis.

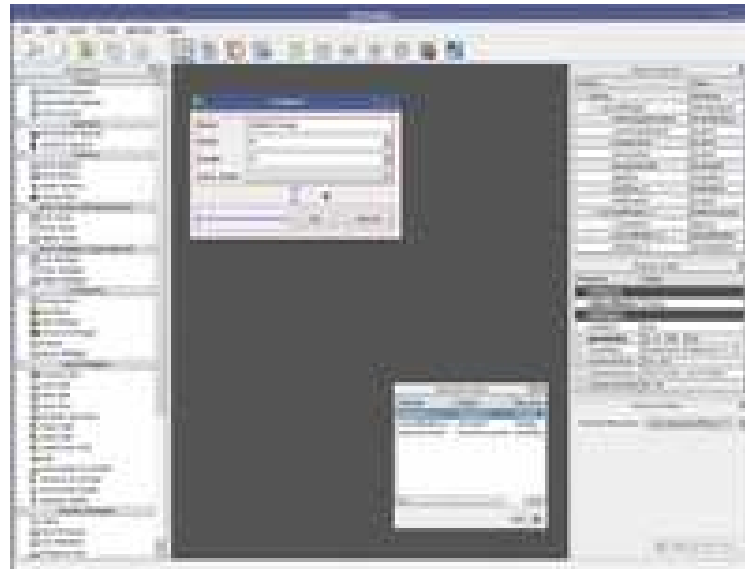


KDE is erin geschreven — en Google Earth, Skype, ...

Qt werkt met **signals** en **slots**. Als je iets doet, geeft je een signal, dat ergens door een slot wordt opgevangen.

Hoe leer je Qt? Of Qt Designer?

Ga naar een Linux-systeem, bijvoorbeeld in zaal 302/304 of 306/308, en start in KDE op. Geef in een terminal-window het commando `designer &` om Qt Designer te starten.



Maak vervolgens een mooie GUI (Grafische User Interface) voor Boter, kaas en eieren. Gebruik daarbij de C++-code in `boter.cc` die hier te vinden is:

<http://www.liacs.nl/home/kosters/pm/qt.php>



<http://www.liacs.nl/home/kosters/pm/qt.php>

|   |  |   |
|---|--|---|
| <p>Programmeermethoden <a href="http://www.liacs.nl/~kosters/pm/qt.php">http://www.liacs.nl/~kosters/pm/qt.php</a></p>  <h3>Qt</h3> <p>We zullen hier kort proberen uit te leggen hoe je met Qt ("cute" of "Q-T", <a href="http://qt.nokia.com">http://qt.nokia.com</a>) een mooie gebruikers-interface voor het spelletje Boter, kaas en eieren kunt maken. Het werkt onder KDE, ook wel onder GNOME, op de Mac, en onder Windows. KDE zelf is met behulp van Qt gemaakt. Er zijn vele versies van Qt in omloop; in het onderstaande is op een tweetal plekken onderscheid tussen Linux en Windows. Qt is een gigantisch pakket, het eventueel installeren kost dan ook aardig wat tijd.</p> <p>We doen drie stappen: eerst maken we de interface, daarna bouwen we het C++-programma dat Boter, kaas en eieren speelt, en tot slot combineren we dit tot een fraai programma.</p> <h4>Stap 1</h4> <p>We zullen met behulp van Qt Designer/Creator een mooie gebruikers-interface (GUI) maken. Begin in een lege directory <code>bke</code>.</p> <p><b>Linux-versie</b><br/>Op de computers in zaal 302304 is Qt al geïnstalleerd. Er zijn via internet ook gratis versies voor Linux op te halen.<br/>Start Qt Designer met <code>designer &amp;</code>. Creëer een nieuwe "Form" van type "Widget", die BKEForm moet gaan heten (bij objectName).</p> <p><b>Windows-versie</b><br/>Installeer onder de LGPL open source licentie van <a href="http://qt.nokia.com/downloads">http://qt.nokia.com/downloads</a>, op de eigen PC, "Qt SDK for Windows" (222 MB). Het uipakken van de file <code>qt-sdk-win-opensource-2010_05.exe</code> (versie november 2010) levert 1,6 GB aan software op! Er wordt onder meer een complete C++-compiler meegeleverd: MinGW.<br/>Start Qt Creator door op de Qtkoon te klikken. Kies File-&gt;New File or Project, en selecteer bij Projects nu</p> <p>1 of 4 <span style="float: right;">12/08/2010 09:22 AM</span></p>   |  | <p>Programmeermethoden <a href="http://www.liacs.nl/~kosters/pm/qt.php">http://www.liacs.nl/~kosters/pm/qt.php</a></p> <p>Qt (links), Qt Designer Form (rechts), Kies daarna Widget, en als filenaam <code>bke.ui</code> in directory <code>bke</code>. Noem de "Form" BKEForm (bij objectName).</p> <p>We gaan voor beide systemen als volgt verder. In deze stap maken we alleen de interface. De interface moet bestaan uit een 9-tal knoppen ("Push Buttons") waarmee straks Boter, kaas en eieren kan worden gespeeld, een Stop-knop (om te stoppen), een Undo-knop (die we voor van alles en nog wat kunnen gaan gebruiken; optioneel), en een tekst-veld ("Label") waarin de status staat, zoals "X is aan de beurt". Zie hier (of de hoofdstukken hiervoor en hierna) voor meer informatie. We gebruiken voorlopig alleen maar knoppen en een "Label", gelaten <code>TextLabel</code>.</p> <p>Noem de 9 spalknoppen bijvoorbeeld <code>Vak11</code>, <code>Vak12</code>, ..., <code>Vak33</code>, en maak de tekst in de knoppen leeg via de "properties". Gebruik "Horizontal Spacers" en "Vertical Spacers", en "Horizontal/Vertical Layout", om de ruimte goed te verdelen.</p> <p>Koppel het <code>clicked()</code>-signaal van de Stop-knop aan het <code>close()</code>-slot van de Form. Zie hier voor meer informatie. Gebruik Edit-Edit Signal/Slot. Je kunt dan met de linker muisknop op de Stop-knop klikken, naar de achtergrond van de "Form" slepen, en daar het juiste slot kiezen; misschien moet je nog een optie in het windowje aanvinken om de mogelijkheden zichtbaar te krijgen.</p> <p>Test de interface met Form-Preview, of bij Tools-Form Editor-Preview. Save hem geregeld als (bij voorbeeld) <code>bke.ui</code>. Als je wat aan de interface wilt wijzigen, moet dat dus hier.</p> <h4>Stap 2</h4> <p>Deze stap is geheel system-onafhankelijk. Haal de C++-file <code>boter.cc</code> op, en zet deze in directory <code>bke</code>. Dit is een compleet C++-programma dat een beetje Boter, kaas en eieren speelt. Compileer (met een compiler naar keuze, <code>g++</code> of <code>Dev-C++</code>) het maar eens, en speel er wat mee. Probeer de code enigszins te begrijpen.</p> <p>Splits de file in <code>boter.h</code> (met de definitie van de klasse <code>bke</code>) en <code>boter.cc</code> (met de functies). Voeg boven en onder in <code>boter.h</code> enkele regels toe:</p> <pre>#ifndef BOTER_H #define BOTER_H  ... Klasse bke ...  #endif //BOTER_H</pre> <p>Zo zorg je er voor dat zaken maar één keer meegecompileerd worden. Voeg boven in <code>boter.cc</code> toe (onder <code>#include &lt;iostream&gt;</code>): <code>#include "boter.h"</code>. Compileer het programma opnieuw en speel er weer even mee. Hoppelijk werkt het nog precies hetzelfde.</p> <p>2 of 4 <span style="float: right;">12/08/2010 09:22 AM</span></p> |
| <p>Programmeermethoden <a href="http://www.liacs.nl/~kosters/pm/qt.php">http://www.liacs.nl/~kosters/pm/qt.php</a></p> <p>Haal de functie <code>main</code> uit <code>boter.cc</code> weg — nu werkt het programma tijdelijk niet meer. Waarschijnlijk hoeven <code>boter.h</code> en <code>boter.cc</code> later niet meer gewijzigd te worden, maar het mag wel.</p> <h4>Stap 3</h4> <p>Nu moeten we de interface en de boter-code nog koppelen: het echte werk. Dit gaat op beide systemen nagenoeg hetzelfde, met of zonder Qt.</p> <p>Haal allereerst <code>main.cc</code> op. Deze file hoeft ook niet meer gewijzigd te worden. Nu komt het. Je hoeft je niet druk te maken om de "control-flow" zoals tot nu toe gebruikelijk. Het wordt namelijk event-driven: je hoeft slechts aan te geven wat er moet gebeuren als je op de knoppen klikt. Haal daartoe <code>bkespel.h</code> en <code>bkespel.cc</code> op, en pas deze twee herhaald aan door functies hiertoe te voegen en steeds te wijzigen. Let erop dat verschillende namen hierin overeen moeten komen met de namen zoals die in Stap 1 zijn gekozen. De executable gaat bij voorbeeld heten naar de directory. En de "Form" moet als objectName (zie properties) BKEForm hebben.</p> <p>Als je al deze files hebt (<code>bke.ui</code>, <code>boter.h</code>, <code>boter.cc</code>, <code>main.cc</code>, <code>bkespel.h</code> en <code>bkespel.cc</code>), alle in dezelfde directory <code>bke</code>, geef dan onder Linux achtereenvolgens:</p> <pre>qmake -project qmake make</pre> <p>De eerste regel maakt een projectfile <code>bke.pro</code>, daarna maakt <code>qmake</code> een "makefile", die het hele compilatie-proces netjes regelt, en die je steeds met <code>make</code> uitvoert. Hoppelijk is er nu een executable <code>bke</code> ontstaan. Later hoeft je trouwens alleen steeds <code>make</code> te doen, tenzij je files hebt toegevoegd. Onder Windows kan dit ook, maar kun je het ook met de knoppen regelen. Eenvoudiger is de file <code>bke.pro</code> hier op te halen, dit project in Qt te laden, en dan met de knoppen te compileren. De executable gaat hier <code>bke.exe</code> heten.</p> <p>Het middelde van de 9 vakjes, <code>Vak22</code>, doet nu iets. Probeer het te snappen, verbeter het, en breidt dit uit naar de andere vakjes. Het spel moet nu goed gespeeld kunnen worden. Het is wellicht verstandig — als je dit allemaal werkend wilt krijgen — om een functie te maken die een coördinatenpaar binnen krijgt, en dan de juiste acties doet. Of eigenlijk was dit vanaf het begin beter geweest. En een functie <code>printoord ( )</code> is ook niet gek. Tot slot: laat de "Undo"-knop het veld weer leeg maken. Of laat de computer een zet doen via de functie <code>randobzet ( )</code>.</p> <p>Als je meer hulp wilt: geef onder Linux <code>assistant &amp;</code>, of kijk onder Windows in de help-files. Wil je</p> <p>3 of 4 <span style="float: right;">12/08/2010 09:22 AM</span></p> | <p>Programmeermethoden <a href="http://www.liacs.nl/~kosters/pm/qt.php">http://www.liacs.nl/~kosters/pm/qt.php</a></p> <p>bij voorbeeld plaatjes op de knoppen zetten? Doe <code>Vak22-&gt;setIcon (QIcon ("lets.jpg"))</code>. Zoek maar eens naar de functies van <code>QPushButton's</code>. Natuurlijk valt er veel meer over Qt te vertellen, maar dat doen we hier niet. Succes!</p> <hr/> <p style="text-align: right;">Tim Cox &amp; Walter Kosters</p> <p>4 of 4 <span style="float: right;">12/08/2010 09:22 AM</span></p> |   |

---

Het schrijven van zo'n programma gaat in drie stappen:

1. bouw een mooie interface;
2. maak een niet-grafisch goed werkend C++-programma;
3. plak deze twee in elkaar; gebruik `make`.

Of misschien  $1 \longleftrightarrow 2$ ?

```
class bke {                                     // file boter.h
public:
    char bord[4][4];
    // we gebruiken bord[0][.] en bord[.][0] niet
    char beurt;
    // wie is aan de beurt: '0' of 'X'
    bke ( );
    bool gewonnen (char & wie);
    bool vol ( );
    void drukaf ( );
    void randomzet ( );
    bool magzet (int i, int j);
    void doezet (int i, int j);
    char wiemoet ( );
    void menszet ( );
}; //bke
```

Als op knop  $(i, j)$  van het speelbord geklikt wordt, moet je een aantal dingen doen:

- controleren of dit mag (vakje vrij?, spel nog niet afgelopen?);
- de inhoud (kleur?) van deze knop wijzigen;
- de zet “administreren”;
- tekst updaten;
- wellicht een computerzet doen.

Probleem is vaak dat alle knoppen bijna dezelfde code hebben, maar dat je lastig informatie kunt doorgeven. Oplossing: maak een extra functie `probeerzet(i, j)` die als het ware een array “is”.

- werk aan de vierde programmeeropgave — de deadline is ongeveer nu
- dinsdag 13 en donderdag 15 december 2011: werkcollege met oude tentamens, zalen B1 en B2
- **vragenuur over tentamen**: dinsdag 3 januari 2012, 11.00 uur, Snellius zaal 174
- **tentamen**: woensdag 4 januari 2012, 10.00–13.00 uur, Snellius (hertentamens: 28 maart, 30 juli 2012)