Informatik I für D-ITET Musterlösung 10 WS 04/05

URL: http://www.ti.inf.ethz.ch/ew/courses/inf1-ITET/

Aufgabe 1

Programm: graphic_sort.C

```
// Programm: graphic_sort.C
// Stellt Quicksort und Bubblesort graphisch dar.

#include <iostream>
#include <utility>
#include <vector>
#include <algorithm>
#include <cassert>
#include <IFET/window>
#include <IFET/math.h>

typedef std::pair<unsigned int,unsigned int>  Pair;
typedef std::vector<Pair>                     Vec;
typedef Vec::iterator                         It;

namespace ifet {

  void draw_segment(int x, int y)
  // PRE: wio.xmin() <= x <= wio.xmax(), wio.ymin() <= y <= wio.ymax().
  // POST: In wio wurde an der x-Koordinate x ein blauer Balken der
  //   Hoehe y gezeichnet.
  {
    assert(ifet::wio.xmin() <= x && x <= ifet::wio.xmax() &&
           ifet::wio.ymin() <= y && y <= ifet::wio.ymax());
    ifet::wio << ifet::blue << ifet::Line(x, ifet::wio.ymin()-1, x, y)
              << ifet::white << ifet::Line(x, y, x, ifet::wio.ymax()+1)
              << ifet::flush;
  } // draw_segment(x, y)

  void graphic_swap(It a, It b)
  // PRE: Es gibt einen gueltigen range, der a und b enthaelt.
  // POST: a->first und b->first wurden vertauscht und die
  //   entsprechenden Balken in wio gezeichnet.
  {
    std::swap(a->first, b->first);
    ifet::draw_segment(a->second, a->first);
    ifet::draw_segment(b->second, b->first);
  } // graphic_swap(a, b)

  It split(It b, It e)
  // PRE: [b,e) ist ein nichtleerer gueltiger range.
  // POST: Die Elemente in [b,e) wurden permutiert.
  //   Rueckgabewert ist ein s aus [b,e), so dass
  //   fuer alle i in [b,s): v[i] <= pivot,
  //   v[s] == pivot sowie
  //   fuer alle i in (s,e): v[i] > pivot,
  //   wobei pivot == *b beim Aufruf der Funktion.
  {
    assert(b != e);
```

```
    It i = b;
    for (It j = ++i; j != e; ++j)
      // Invariante: *[b,i) <= *b && *[i,j) > *b.
      if (*j <= *b) ifet::graphic_swap(i++, j);

    // Tausche das Pivot-Element in die richtige Position
    // (am Ende des ersten Blocks)
    ifet::graphic_swap(b, --i);
    return i;
  } // split(b, e)

  void quicksort(It b, It e)
  // PRE: [b,e) ist ein gueltiger range.
  // POST: Die Elemente in [b,e) sind aufsteigend sortiert.
  {
    if (b == e) return;
    It mid = ifet::split(b, e);
    ifet::quicksort(b, mid);
    ifet::quicksort(++mid, e);
  } // quicksort(b, e)

  void bubble_sort(It b, It e)
  // PRE: [b,e) ist ein gueltiger range.
  // POST: Die Elemente in [b,e) sind aufsteigend sortiert.
  {
    // Wurden im aktuellen Durchlauf zwei Elemente getauscht?
    bool swapped = true;
    while (b != e && swapped) {
      swapped = false;
      It prev = b;
      It cur = prev;
      while (++cur != e) {
        if (*cur < *prev) {
          ifet::graphic_swap(cur, prev);
          swapped = true;
        }
        prev = cur;
      }
      // Das letzte Element ist jetzt auf jeden Fall korrekt:
      --e;
    }
  } // bubble_sort(b, e)

} // namespace ifet

int main()
{
  // Eingabe
  double rand;
  do {
    std::cout << "Startwert fuer Zufallszahlengenerator " <<
      "(ein double aus [0,1): ";
    std::cin >> rand;
  } while (rand < 0 || rand >= 1);
  int c;
  do {
    std::cout << "Soll mit Quicksort[0] oder Bubblesort[1] sortiert werden? ";
    std::cin >> c;
```

```
  } while (c != 0 && c != 1);

  // Initialisierung
  const unsigned int window_size = 512;
  Vec v;
  for (unsigned int i = 0; i < window_size; ++i) {
    v.push_back(Pair(i, i));
    ifet::draw_segment(i, i);
  }

  // permutiere zufaellig
  for (It b = v.begin(); b != v.end(); ++b) {
    ifet::graphic_swap(b, b + int((v.end()-b) * rand));
    rand = ifet::random(rand);
  }

  // Sortieren
  if (c == 0)
    ifet::quicksort(v.begin(), v.end());
  else
    ifet::bubble_sort(v.begin(), v.end());

  // Warte auf mouse-click
  std::cout << "-------------------------------------------------\n"
            << "Ein Mouseclick beendet das Programm.\n"
            << "-------------------------------------------------"
            << std::endl;
  ifet::wio.wait_for_mouse_click();

  return 0;
}
```