Import JJIO
Class SortTimer
   -- Name Anon Nymous
   -- Does time some Sorts of random arrays

Routine mainRoutine (none) is public
   Boxes size, i ofType int
   Boxes startTime, finishTime ofType int
   Box a ofType int

Start
   Output "Enter the size of array "
   Input size
   Outputln size -- echo
   NewArray a ofType int[size]
   Call getRandArray with (a, size)
   -- Outputln "Some first values, random "
   -- Call outArray with (a, 20) -- test
   -- Time the first sort
   Set startTime = jjMillisSinceStart()
   Call swapSort with (a, size) -- TIME!!
   Set finishTime = jjMillisSinceStart()
   Output "Time duration of swap1 sort is "
   Outputln (finishTime - startTime)
   -- Time the second sort
   Call getRandArray with (a, size)
   Set startTime = jjMillisSinceStart()
   Call swapSort2 with (a, size) -- TIME!!
   Set finishTime = jjMillisSinceStart()
   Output "Time duration of swap2 sort is "
   Outputln (finishTime - startTime)
   -- Test some values
   -- Outputln "Some last values, sorted "
   -- Call outArray with (a, 30) -- to test
   Outputln " " -- gap
EndRoutine mainRoutine

Routine getRandArray (b,m) is public
   Slot b ofType int[]
   Slot m ofType int
   -- Fill array with random values (percents)
   Box i ofType int
   Set i = 0
   Repeat
      ExitOn (i == m)
      Set b[i] = RealToInt (101.0 * Random())
      Inc i by 1
   EndRepeat
EndRoutine getRandArray

Routine outArray (c,n) is public
   Slot c ofType int[]
   Slot n ofType int
   -- Output the array 10 entries to a line
   Box i ofType int
   Set i = 0
   Repeat
      ExitOn (i == n)
      If ((i % 10) == 0) then
         Outputln " " -- new line
      EndIf
      Output IntToStr (c[i]) + " "
      Inc i by 1
   EndRepeat
   Outputln " " -- gap
EndRoutine outArray

Routine swapSort (A, n) is public
   Slot A ofType int[] -- array
   Slot n ofType int -- size
   Boxes i, j, temp ofType int
   -- Does sort with about n * n loops
   Set i = 0
   Repeat
      ExitOn (i == n - 1)
      Set j = 0
      Repeat
         ExitOn (j == n - i - 1)
         If (A[j] < A[j+1]) then
            -- swap adjacent values
            Set temp = A[ j ]
            Set A[j+1] = temp
         EndIf
         Inc j by 1
      EndRepeat
      Inc i by 1
   EndRepeat
EndRoutine swapSort

Routine swapSort2 (A, n) is public
   Slot A ofType int[] -- array
   Slot n ofType int -- size
   Boxes i, j, temp ofType int
   Box done ofType bool
   -- Does sort with halt when done
   Repeat
      Set done = true
      Set j = 0
      Repeat
         ExitOn (j == n - 1)
         If (A[j] < A[j+1]) then
            -- swap adjacent values
            Set temp = A[ j ]
            Set done = false
         EndIf
         Inc j by 1
      EndRepeat
      ExitOn done
   EndRepeat
EndRoutine swapSort2

EndClass SortTimer

Enter the size of array 1000
Time duration of swap1 sort is 59
Time duration of swap2 sort is 85
Enter the size of array 1000
Time duration of swap1 sort is 58
Time duration of swap2 sort is 86
Enter the size of array 10000
Time duration of swap1 sort is 6025
Time duration of swap2 sort is 9190
Enter the size of array 100000
Time duration of swap1 sort is 623646
Time duration of swap2 sort is 968467