CISP 401

Object Oriented Programming - Java

Dixon

==

Class Diagram (Generic Data and Generic Sort)

```
GenericContainer
 (-) GenericData[] gd
 (+) abstract Sort(GenericData[] data)
 (+) ... (all standard container services)

GenericData
  (+) abstract boolean isLess(GenericData gd)
  (+) abstract boolean isEqual(GenericData gd)
  (+) abstract boolean isGreater(GenericData gd)

InsertionSort extends GenericSort
  (+) Sort(GenericData[] data){ insertionSort(data, ...)}

QuickSort extends GenericSort
  (+) Sort(GenericData[] data){ quickSort(data, ...) }

IntData extends GenericData
  (-) int data
  (+) boolean isLess(GenericData gd)
  (+) boolean isEqual(GenericData gd)
  (+) boolean isGreater(GenericData gd)
  (+) get(), set(d)

CharData extends GenericData
  (-) char data
  (+) boolean isLess(GenericData gd)
  (+) boolean isEqual(GenericData gd)
  (+) boolean isGreater(GenericData gd)
  (+) get(), set(c)
```

```
public void quickSort(GenericData[] gd,int start,int finish)
{
     // quicksort initialization step(start, finish, pivot)
     while (left < right) // quicksort pass loop
       // find left swap candidate
       // find right swap candidate
       if left <= right
          swap(gd[left],gd[right])
          left++, right—
       end if
     wend
     if (start < right) quickSort(gd,start,right)
     if (left < finish) quickSort(gd,left,finish)
}
```

Java Generic Translation:

```
public void quickSort(GenericData[] gd,int start,int finish)
{
     // quicksort initialization step(start, finish, pivot)
     while (left < right) // quicksort pass loop
       if (gd[left] < pivot) left++ // find left swap candidate
          restate: if (gd[left].isLess(pivot)) left++
       if (gd[right] > pivot) right++ // find right swap candidate
          restate: if (gd[right].isGreater(pivot)) left++
       if left <= right
          Swap(gd[left],gd[right])
          left++, right—
       end if
     wend
     if (start < right) quickSort(gd,start,right)
     if (left < finish) quickSort(gd,left,finish)
}
```