05: Dynamic Arrays

Exercise Instructions

Goal

In this module, you will write a simple class that represents a student with a name and several other details. Dynamic arrays will be used to store several students as well as to sort or find elements within the array.

Introduction

Even when you want to use a simple fixed-length array, Symbian OS provides a class that provides you with more comfort. More advanced classes for dynamic arrays are available as well. Those exist in several different forms to suite any usage scenario you might encounter. This exercise takes a closer look at the fixed-length array as well as the dynamic `RArray` / `RPointerArray`.

Structure of this Exercise

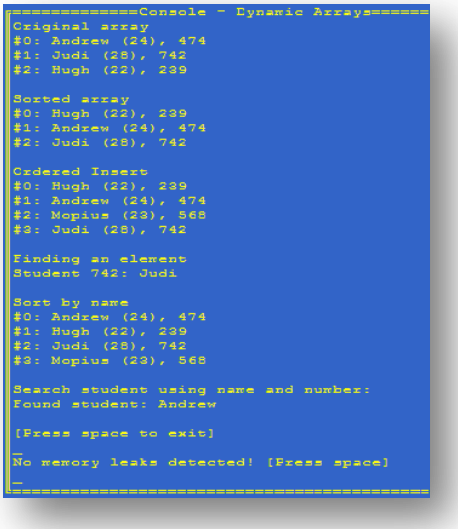
In this exercise you will create a small application that can manage students. Information about them is stored in an array, which can be sorted and allows finding elements. In the first part of the exercise, you have to create an `RArray` using `TStudent` objects, for the second part you have to rewrite the application based on the tutorial document of this module to use `CStudent` objects and an `RPointerArray`.

The first task of this exercise is to create a `TFixedArray`, which will just contain IDs of the students, stored as `TInt`s.

Next, the `RArray` has to be created and filled with data, using the ID as an automatic sorting criterion. This allows sorting the array, ordered inserts, as well as finding objects.

The third task is to write custom comparison and matching functions, which allow using the name of the student as sorting criterion instead of the ID.

The final output should look like this:



```
==============Console - Dynamic Arrays======
Original array
#0: Andrew (24), 474
#1: Judi (28), 742
#2: Hugh (22), 239

Sorted array
#0: Hugh (22), 239
#1: Andrew (24), 474
#2: Judi (28), 742

Ordered Insert
#0: Hugh (22), 239
#1: Andrew (24), 474
#2: Mopius (23), 568
#3: Judi (28), 742

Finding an element
Student 742: Judi

Sort by name
#0: Andrew (24), 474
#1: Hugh (22), 239
#2: Judi (28), 742
#3: Mopius (23), 568

Search student using name and number:
Found student: Andrew

[Press space to exit]
_
No memory leaks detected! [Press space]
_
============================================
```

After this part is finished, rewrite the application so that it uses `CStudent` objects which store the name in a dynamic `RBuf` descriptor class instead of a fixed length `TBuf`. This also requires switching to an `RPointerArray` – however, you will notice that most of the code can actually stay the same. Take a look at the tutorial document of this module for further hints on how to rewrite the application.

Detailed Description

Fixed Array

In this first and very short section, you have to create a fixed array that contains several `TInt` variables to store the IDs of the students.

Dynamic RArray

Create an `RArray` that contains several `TStudent` objects. The Symbian OS dynamic arrays allow using one of the numeric member variables to automatically sort the array – and this is what is needed here to sort them using their ID. Several other tests explore how to find an object in the array or how to compare two students based on their ID.

Custom Comparison

`RArray`s also allow implementing a custom comparison and matching function. This gives you full control over how the array elements should be sorted, without having to implement an own sorting algorithm. For this example, the objects should now be sorted based on their name. You do not have to implement the string comparison yourself; the descriptor classes do already provide this.