Exercise 2a. (Container implementation is independent of the itemtype to be stored)

Modify your class from the exercise 1 in such a way that it works with the given simpleList.java program (e.g. it must implement `Comparable<Time>` interface). Overload also the toString() method in order to be able to print those time values easily and implement a constructor with initial values as parameters, e.g. `Time(int hour, int min)`.

At the lectures, we studied the simple list implementation using templates. The application was using list to store characters. Now we use the same list (available as a simpleList.java file from Tuubi) to store Time values.

simpleList.java contains also the source code for a small application to test the Time valued list. Test first your Time class with the given test main method. Then implement a new operation function for the simpleList container, `bool insert_to_begin(T item)` which adds a new item to the beginning of the list container. After that change the line `list.insert_to_end(item)` to `list.insert_to_begin(item)` in order to test your new function.

Remark. Use your time as a component. This means that you should have the separate Time.java file.

Extra exercise 2b. (Ordered list, 0.25p)

In this exercise we add an another operation function, `list.insert(item)`, to the list. This operation function insert an item to the list in such a way that the list is always ordered (smallest item first). Verify that the list is always in order.