

```

1 import my_time
2 from operator import attrgetter
3 from collections import namedtuple
4
5 Competitor = namedtuple('Competitor', ['name', 'tstart', 'tfinal', 'elapsed'])
6
7 def is_valid_line(data):
8     return len(data)>0
9
10 def create_competitor(dataline):
11     values = dataline.split(";")
12     name = values[0]
13     tstart = my_time.Time(values[1])
14     tfinal = my_time.Time(values[2])
15     elapsed = tfinal - tstart
16     return Competitor(name, tstart, tfinal, elapsed)
17
18 class CompetitorsList:
19     def __init__(self, complist=None):
20         if complist:
21             self.complist = complist
22         else:
23             self.complist = []
24
25     def read_scorecard(self, fname):
26         with open(fname, mode="rt", encoding="utf-8") as f:
27             for data_line in f:
28                 if is_valid_line(data_line):
29                     # self.complist.append(Competitor(data_line))
30                     self.complist.append(create_competitor(data_line))
31
32     def __str__(self):
33         format_string = "{:<15} {:<10} {:<10} {:<10}\n"
34         message = format_string.format('Name', 'Start', 'Finish', 'Elapsed')
35         for c in self.complist:
36             message += format_string.format(str(c.name), str(c.tstart),
37                                             str(c.tfinal), str(c.elapsed))
38         return message
39
40     def sort_by(self, sorting_attribute):
41         return CompetitorsList(sorted(self.complist, key=attrgetter(sorting_attribute)))
42

```