

3 Self-Referential Data; Methods for Classes.

Portfolio Problems

- **5 bonus points for completing this task** Visit the class wiki and add a line about yourself. Does not have to be long, does not have to say much - just enough to see that you have visited the page.

Please, e-mail Professor Proulx (from your ccs account) if you are having difficulties accessing the page.

- **Methods for weather and dates**

Look at the weather data information from the Problem 3.2. Design the following methods for the weather data:

1. Problem 11.2 in the text.
2. When the forecasters predict the weather for the next day they need the method *nextDate*, defined in the class *Date*. Make it simple - ignore leap years and assume every month has 30 days.

Pair Programming Assignment

Keeping Time

All problems in this part become one Beginner ProfessorJ program. Use the data definitions and examples from your previous homework or lab whenever appropriate.

3.1 Problem

When dealing with radio shows we need to keep track of the clock time. We start by designing some methods for the class *ClockTime* that may be useful. Record the time in hours (24 hour clock) and minutes only. Design the method to solve the following problems:

1. We know the starting time for the show and its duration in minutes. Produce the *ClockTime* for when the show ends.
(Method name: *endTime*)

Remember: One task one method.

2. At other times we know the starting time and the ending time for a show and want to compute the duration in minutes.
(Method name: *durationTill*)
3. We want to know which of the two starting times is earlier.
(Method name: *isBefore*)
4. We are planning the schedule and want to make sure two shows do not overlap. For this problem we need to define a class of data to represent one show. For each show we record its name, the starting time, and the duration in minutes.
(Class name: *Show*)
5. Is the time when one show ends before the time when another show begins?
(Method name: *endsBefore*)
6. Is one show the same as another one, other than the starting time? If the shows have the same name, but not the same duration, they are considered to be different shows.
(Method name: *sameShow*)
7. We want to schedule a new show to start after this one has finished. We know the name of the show and its duration.
(Method name: *newShow*)

3.2 Problem

Methods for taxi vehicles

Look at the classes you defined for the Problem 4.6. Design the following methods for the taxi vehicles:

1. Compute the fare if you know how many passengers are taking the taxi and how far they traveled (in whole miles). The fare is zero if the taxi cannot accomodate the desired number of passengers.
(Method name: *fare*)
Remember: one task one method.
2. Compute the cost for each passenger, if the group is splitting the fare evenly. Of course, you will need the same information as in the previous case.
(Method name: *costPerRider*)

3. Determine whether it is cheaper to travel by one type of taxi or another one, for a given number of passengers and the trip length.
(Method name: *isCheaper*)