Programming Exercises - SDJ - Session 09
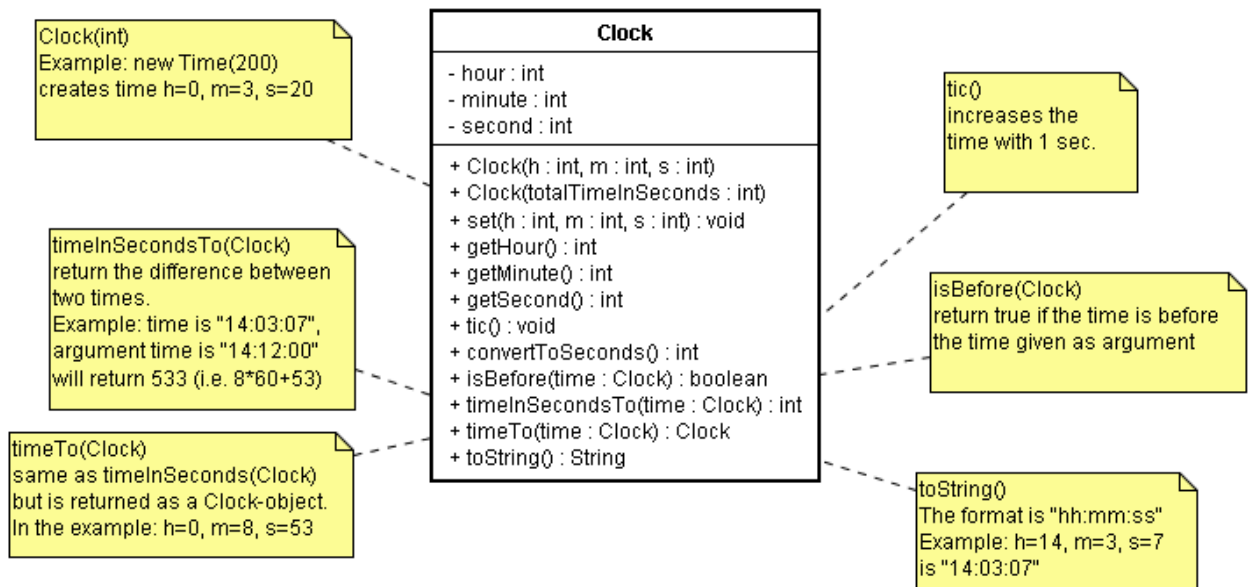
Exercise 09.01

Write a program that generates a random number between 1 and 10, and then asks the user to try and guess it. When the user enters a guess, compare it with the random generated number, and tell the user whether or not the guess was correct.

Exercise 09.02

Add a method `getMonthName()` to the MyDate class, that uses a switch statement to return the name of the current month (1 = January, 2 = February, etc.).

Exercise 09.03

Implement a class `Clock` holding information of a time as `hour, minute` and `second`. The following class diagram has to be implemented:


```
Clock(int)
Example: new Time(200)
creates time h=0, m=3, s=20
```

```
timeInSecondsTo(Clock)
return the difference between
two times.
Example: time is "14:03:07",
argument time is "14:12:00"
will return 533 (i.e. 8*60+53)
```

```
timeTo(Clock)
same as timeInSeconds(Clock)
but is returned as a Clock-object.
In the example: h=0, m=8, s=53
```

Clock

- hour : int
- minute : int
- second : int

+ Clock(h : int, m : int, s : int)
+ Clock(totalTimeInSeconds : int)
+ set(h : int, m : int, s : int) : void
+ getHour() : int
+ getMinute() : int
+ getSecond() : int
+ tic() : void
+ convertToSeconds() : int
+ isBefore(time : Clock) : boolean
+ timeInSecondsTo(time : Clock) : int
+ timeTo(time : Clock) : Clock
+ toString() : String

```
tic()
increases the
time with 1 sec.
```

```
isBefore(Clock)
return true if the time is before
the time given as argument
```

```
toString()
The format is "hh:mm:ss"
Example: h=14, m=3, s=7
is "14:03:07"
```

Then write a test application named `ClockTest` that demonstrates the functionality of the `Clock` class.

Exercise 09.04

Modify class `Clock` so that you:

a) Add a `boolean` instance variable called `timeFormat24`, that can hold the value `true` to represent that the clock is set to a 24-hour time format and `false` if it is set to a 12-hour time format.

b) Add a get method called `isTimeFormat24()` that returns the value of `timeFormat24`.

c) Add a set method `setTimeFormat(int hourFormat)` that takes either 12 or 24 as argument and sets the time format according to this. If any other value is put as argument it should make no changes.

d) Change the `toString()` method so that time is returned in 12-hour format if `timeFormat24` is `false`, and in 24-hour format if `timeFormat24` is `true`. Example: h=20, m=5, s = 40 will return "20:05:40" if `timeFormat24` is `true` and "8:05:40 PM" if `timeFormat24` is `false`.