Assignment 3 - G03P02

Group information

- Ana Inês Oliveira de Barros - `up201806593@fe.up.pt`;
- João de Jesus Costa - `up201806560@fe.up.pt`

Function Selection Process

From the previous assignment's report:

> The aim of this assignment is to perform black-box testing. This is problematic because none of the methods in the code are documented (e.g. javadoc). In order to find the purpose of each method, we needed to follow our intuition about the names of the methods, arguments, and classes. Having more extensive documentation would allow for better black-box testing.
>
> Since we aren't using mocks, we tried to test methods that didn't depend on other objects of the project. We discarded functions belonging to the `gui` package due to its dependence on *swing*. The `misc` package was also discarded since it only contains one function (not enough for the completion of the assignment). Functions related to elapsed time were also ignored.
>
> The selected package for testing was the `de.dominik_geyer.jtimesched.project` package.

In this assignment, we selected 3 methods used in the previous one, but changed some of the categories based on newly acquired knowledge.

Method 1

Method: `void setSecondsOverall(int secondsOverall)` in `Project.java` line 178.

Method's purpose: This function sets the *seconds overall* of a project as the value received as an argument (if valid).

Reason for selection: It is important that this function works as expected since other methods depend on it.

Identify the parameters

`secondsOverall` - integer (`int`) representing the number of seconds overall.

Characteristics of the parameters

The integer should represent a positive number between 0 and infinity.

Constraints

Negative time is not allowed - `secondsOverall >= 0`

Partitions

- E1 - negative number
  - `secondsOverall < 0`
- E2 - positive number (including 0)
  - `secondsOverall >= 0`

Boundaries

Partition	On-point(s)	Off-point(s)
E1	`0`	`-1`
E2	`0`	`-1`

Generate tests

Partition	Boundary	Input
E1	On-point	0
E1	Off-point	-1
E2	On-point	0
E2	Off-point	-1

4 tests.

Filter redundant tests

Partition	Boundary	Input	Expected output
E1	On-point	0	0
E1	Off-point	-1	0

Filtered down to 2 tests.

Unit Tests

We created one test with the inputs of each line on the table. The test function is the `void setSecondsOverallTest(int secondsOverall)` and the input generator is `Stream<Arguments> setSecondsOverallInputs()`. Both of these are present in the `ProjectTest.java` file of the `test` package.

Results: all the tests pass successfully.

Method 2

Method: `public static int parseSeconds(String strTime)` in `ProjectTime.java` line 36.

Purpose: This function receives a string representing time, in `hh:mm:ss` format, and returns the total number of seconds it represents.

Reason for selection: This method deals with parsing of user input, which needs to be robust.

Identify the parameters

`strTime` is a string representing time in 3 components:

- hours (hh);
- minutes (mm);
- seconds (ss).

Characteristics of the parameters

- The string should represent a valid time in the format: `hh:mm:ss`
- *hh* -- represents the hour
- *mm* -- represents the minutes
- *ss* -- represents the seconds
- It should be possible to pass single digits for each component

Constraints

- Input can't be `null`
  - `input != null`
- String can only contain digits and the `:` char
- Seconds lie within the interval [0, 59]
  - `0 <= ss <= 59`
- Minutes lie within the interval [0, 59]
  - `0 <= mm <= 59`
- Hours must be a positive number (including 0)
  - `hh >= 0`

Partitions

- E1 - null input - `null`
- E2 - empty input - `""`
- E3 - non-empty string input - `"00:00:00"`

E3 can be sub-divided into other categories:

- E4 - input containing non-digit - `"0a:00:00"`
- E5 - input containing only digits and the `:` char - `"00:00:00"`
- E6 - seconds < 0 - `"00:00:-1"`
- E7 - seconds > 59 - `"00:00:60"`
- E8 - 0 <= seconds <= 59 - `00:00:30`
- E9 - minutes < 0 - `"00:-1:00"`
- E10 - minutes > 59 - `"00:60:00"`
- E11 - 0 <= minutes <= 59 - `00:30:00`
- E12 - hours < 0 - `-1:00:00`
- E13 - seconds < 0 and minutes < 0 - `00:-1:-1`

- E14 - seconds < 0 and minutes > 59 - `00:60:-1`
- E15 - seconds < 0 and 0 <= minutes <= 59 - `00:30:-1`
- E16 - seconds < 0 and hours < 0 - `-1:00:-1`
- E17 - seconds < 0, hours < 0 and minutes < 0 - `-1:-1:-1`
- E18 - seconds < 0, minutes > 59 and hours < 0 - `-1:60:-1`
- E19 - seconds < 0, 0 <= minutes <= 59 and hours < 0 - `-1:30:-1`
- E20 - seconds > 59 and minutes < 0 - `00:-1:60`
- E21 - seconds > 59 and minutes > 59 - `00:60:60`
- E22 - seconds > 59 and 0 <= minutes <= 59 - `00:30:60`
- E23 - seconds > 59 and hours < 0 - `-1:00:60`
- E24 - seconds > 59, hours < 0 and minutes < 0 - `-1:-1:60`
- E25 - seconds > 59, minutes > 59 and hours < 0 - `-1:60:60`
- E26 - seconds > 59, 0 <= minutes <= 59 and hours < 0 - `-1:30:60`
- E27 - 0 <= seconds <= 59 and minutes < 0 - `00:-1:30`
- E28 - 0 <= seconds <= 59 and minutes > 59 - `00:60:30`
- E29 - 0 <= seconds <= 59 and 0 <= minutes <= 59 - `00:30:30`
- E30 - 0 <= seconds <= 59 and hours < 0 - `-1:00:30`
- E31 - 0 <= seconds <= 59, hours < 0 and minutes < 0 - `-1:-1:30`
- E32 - 0 <= seconds <= 59, minutes > 59 and hours < 0 - `-1:60:30`
- E33 - 0 <= seconds <= 59, 0 <= minutes <= 59 and hours < 0 - `-1:30:30`
- E34 - minutes < 0 and hours < 0 - `-1:-1:00`
- E35 - input missing hours - `:00:00`
- E36 - input missing minutes - `00::00`
- E37 - input missing seconds - `00:00:`
- E38 - input missing seconds and minutes - `00::`
- E39 - input missing seconds and hours - `:00:`
- E40 - input missing hours and minutes - `::00`
- E41 - input missing hours, seconds and minutes - `::`
- E42 - input missing both separators - `000000`
- E43 - input missing right separator - `00:0000`
- E44 - input missing left separator - `0000:00`

Boundaries

Partition	On-point(s)	Off-point(s)
E1	`null`	`""`, `"00:00:00"`
E2	`""`	`null`, `"00:00:00"`
E3	`"00:00:00"`	`""`, `null`
E4	`"0a:00:00"`	`"00:00:00"`
E5	`"00:00:00"`	`"0a:00:00"`
E6	`"00:00:00"`	`"00:00:-1"`
E7	`"00:00:59"`	`"00:00:60"`
E8	`"00:00:00"` and `"00:00:59"`	`"00:00:-1"`, `"00:00:60"`

Partition	On-point(s)	Off-point(s)
E9	`"00:00:00"`	`"00:-1:00"`
E10	`"00:59:00"`	`"00:60:00"`
E11	`"00:00:00"` and `"00:59:00"`	`"00:-1:00"`, `"00:60:00"`
E12	`"00:00:00"`	`"-1:00:00"`
E13	`"00:00:00"`	`"00:-1:-1"`
E14	`"00:59:00"`	`"00:60:-1"`
E15	`"00:00:00"` and `"00:59:00"`	`"00:-1:-1"`, `"00:60:-1"`
E16	`"00:00:00"`	`"-1:00:-1"`
E17	`"00:00:00"`	`"-1:-1:-1"`
E18	`"00:59:00"`	`"-1:60:-1"`
E19	`"00:00:00"` and `"00:59:00"`	`"-1:-1:-1"`, `"-1:60:-1"`
E20	`"00:00:59"`	`"00:-1:60"`
E21	`"00:59:59"`	`"00:60:60"`
E22	`"00:00:59"` and `"00:59:59"`	`"00:-1:60"`, `"00:60:60"`
E23	`"00:00:59"`	`"-1:00:60"`
E24	`"00:00:59"`	`"-1:-1:60"`
E25	`"00:59:59"`	`"-1:60:60"`
E26	`"00:00:59"` and `"00:59:59"`	`"-1:-1:60"`, `"-1:60:60"`
E27	`"00:00:00"` and `"00:00:59"`	`"00:-1:60"`, `"00:-1:-1"`
E28	`"00:59:00"` and `"00:59:59"`	`"00:60:60"`, `"00:60:-1"`
E29	`"00:00:00"`, `"00:00:59"`, `"00:59:00"`, `"00:59:59"`	`"00:-1:-1"`, `"00:-1:60"`, `"00:60:-1"`, `"00:60:60"`
E30	`"00:00:00"` and `"00:00:59"`	`"-1:00:-1"`, `"-1:00:60"`
E31	`"00:00:00"` and `"00:00:59"`	`"-1:-1:-1"`, `"-1:-1:60"`
E32	`"00:59:00"` and `"00:59:59"`	`"-1:60:-1"`, `"-1:60:60"`
E33	`"00:00:00"`, `"00:00:59"`, `"00:59:00"`, `"00:59:59"`	`"-1:-1:-1"`, `"-1:-1:60"`, `"-1:60:-1"`, `"-1:60:60"`
E34	`"00:00:00"`	`"-1:-1:-1"`
E35	`"00:00:00"`	`":00:00"`

Partition	On-point(s)	Off-point(s)
E36	`"00:00:00"`	`"00::00"`
E37	`"00:00:00"`	`"00:00:"`
E38	`"00:00:00"`	`"00::"`
E39	`"00:00:00"`	`":00:"`
E40	`"00:00:00"`	`"::00"`
E41	`"00:00:00"`	`"::"`
E42	`"00:00:00"`	`"000000"`
E43	`"00:00:00"`	`"00:0000"`
E44	`"00:00:00"`	`"0000:00"`

Generate tests

Partition	Boundary	Input
E1	On-point	`null`
E1	Off-point 1	`""`
E1	Off-point 2	`"00:00:00"`
E2	On-point	`""`
E2	Off-point 1	`null`
E2	Off-point 2	`"00:00:00"`
E3	On-point	`"00:00:00"`
E3	Off-point 1	`""`
E3	Off-point 2	`null`
E4	On-point	`"0a:00:00"`
E4	Off-point	`"00:00:00"`
E5	On-point	`"00:00:00"`
E5	Off-point	`"0a:00:00"`
E6	On-point	`"00:00:00"`
E6	Off-point	`"00:00:-1"`
E7	On-point	`"00:00:59"`
E7	Off-point	`"00:00:60"`

Partition	Boundary	Input
E8	On-point 1	`"00:00:00"`
E8	On-point 2	`"00:00:59"`
E8	Off-point 1	`"00:00:-1"`
E8	Off-point 2	`"00:00:60"`
E9	On-point	`"00:00:00"`
E9	Off-point	`"00:-1:00"`
E10	On-point	`"00:59:00"`
E10	Off-point	`"00:60:00"`
E11	On-point 1	`"00:00:00"`
E11	On-point 2	`"00:59:00"`
E11	Off-point 1	`"00:-1:00"`
E11	Off-point 2	`"00:60:00"`
E12	On-point	`"00:00:00"`
E12	Off-point	`"-1:00:00"`
E13	On-point	`"00:00:00"`
E13	Off-point	`"00:-1:-1"`
E14	On-point	`"00:59:00"`
E14	Off-point	`"00:60:-1"`
E15	On-point 1	`"00:00:00"`
E15	On-point 2	`"00:59:00"`
E15	Off-point 1	`"00:-1:-1"`
E15	Off-point 2	`"00:60:-1"`
E16	On-point	`"00:00:00"`
E16	Off-point	`"-1:60:-1"`
E17	On-point	`"00:00:00"`
E17	Off-point	`"-1:-1:-1"`
E18	On-point	`"00:59:00"`
E18	Off-point	`"-1:60:-1"`

Partition	Boundary	Input
E19	On-point 1	`"00:00:00"`
E19	On-point 2	`"00:59:00"`
E19	Off-point 1	`"-1:-1:-1"`
E19	Off-point 2	`"-1:60:-1"`
E20	On-point	`"00:00:59"`
E20	Off-point	`"00:-1:60"`
E21	On-point	`"00:59:59"`
E21	Off-point	`"00:60:60"`
E22	On-point 1	`"00:00:59"`
E22	On-point 2	`"00:59:59"`
E22	Off-point 1	`"00:-1:60"`
E22	Off-point 2	`"00:60:60"`
E23	On-point	`"00:00:59"`
E23	Off-point	`"-1:00:60"`
E24	On-point	`"00:00:59"`
E24	Off-point	`"-1:-1:60"`
E25	On-point	`"00:59:59"`
E25	Off-point	`"-1:60:60"`
E26	On-point 1	`"00:00:59"`
E26	On-point 2	`"00:59:59"`
E26	Off-point 1	`"-1:-1:60"`
E26	Off-point 2	`"-1:60:60"`
E27	On-point 1	`"00:00:00"`
E27	On-point 2	`"00:00:59"`
E27	Off-point 1	`"00:-1:60"`
E27	Off-point 2	`"00:-1:-1"`
E28	On-point 1	`"00:59:00"`
E28	On-point 2	`"00:59:59"`

Partition	Boundary	Input
E28	Off-point 1	`"00:60:60"`
E28	Off-point 2	`"00:60:-1"`
E29	On-point 1	`"00:00:00"`
E29	On-point 2	`"00:00:59"`
E29	On-point 3	`"00:59:00"`
E29	On-point 4	`"00:59:59"`
E29	Off-point 1	`"00:-1:-1"`
E29	Off-point 2	`"00:-1:60"`
E29	Off-point 3	`"00:60:-1"`
E29	Off-point 4	`"00:60:60"`
E30	On-point 1	`"00:00:00"`
E30	On-point 2	`"00:00:59"`
E30	Off-point 1	`"-1:00:-1"`
E30	Off-point 2	`"-1:00:60"`
E31	On-point 1	`"00:00:00"`
E31	On-point 2	`"00:00:59"`
E31	Off-point 1	`"-1:-1:-1"`
E31	Off-point 2	`"-1:-1:60"`
E32	On-point 1	`"00:59:00"`
E32	On-point 2	`"00:59:59"`
E32	Off-point 1	`"-1:60:-1"`
E32	Off-point 2	`"-1:60:60"`
E33	On-point 1	`"00:00:00"`
E33	On-point 2	`"00:00:59"`
E33	On-point 3	`"00:59:00"`
E33	On-point 4	`"00:59:59"`
E33	Off-point 1	`"-1:-1:-1"`
E33	Off-point 2	`"-1:-1:60"`

Partition	Boundary	Input
E33	Off-point 3	`"-1:60:-1"`
E33	Off-point 4	`"-1:60:60"`
E34	On-point	`"00:00:00"`
E34	Off-point	`"-1:-1:-1"`
E35	On-point	`"00:00:00"`
E35	Off-point	`":00:00"`
E36	On-point	`"00:00:00"`
E36	Off-point	`"00::00"`
E37	On-point	`"00:00:00"`
E37	Off-point	`"00:00:"`
E38	On-point	`"00:00:00"`
E38	Off-point	`"00::"`
E39	On-point	`"00:00:00"`
E39	Off-point	`":00:"`
E40	On-point	`"00:00:00"`
E40	Off-point	`"::00"`
E41	On-point	`"00:00:00"`
E41	Off-point	`"::"`
E42	On-point	`"00:00:00"`
E42	Off-point	`"000000"`
E43	On-point	`"00:00:00"`
E43	Off-point	`"00:0000"`
E44	On-point	`"00:00:00"`
E44	Off-point	`"0000:00"`

124 tests.

Filter redundant tests

Partition	Boundary	Input	Expected outcome
E1	On-point	`null`	Thrown exception
E1	Off-point 1	`""`	Thrown exception
E1	Off-point 2	`"00:00:00"`	0
E4	On-point	`"0a:00:00"`	Thrown exception
E6	Off-point	`"00:00:-1"`	Thrown exception
E7	On-point	`"00:00:59"`	59
E7	Off-point	`"00:00:60"`	Thrown exception
E9	Off-point	`"00:-1:00"`	Thrown exception
E10	On-point	`"00:59:00"`	3540
E10	Off-point	`"00:60:00"`	Thrown exception
E12	Off-point	`"-1:00:00"`	Thrown exception
E13	Off-point	`"00:-1:-1"`	Thrown exception
E14	Off-point	`"00:60:-1"`	Thrown exception
E16	Off-point	`"-1:60:-1"`	Thrown exception
E17	Off-point	`"-1:-1:-1"`	Thrown exception
E20	Off-point	`"00:-1:60"`	Thrown exception
E21	Off-point	`"00:60:60"`	Thrown exception
E23	Off-point	`"-1:00:60"`	Thrown exception
E24	Off-point	`"-1:-1:60"`	Thrown exception
E25	Off-point	`"-1:60:60"`	Thrown exception
E30	Off-point 1	`"-1:00:-1"`	Thrown exception
E32	On-point 2	`"00:59:59"`	3599
E35	Off-point	`":00:00"`	Thrown exception
E36	Off-point	`"00::00"`	Thrown exception
E37	Off-point	`"00:00:"`	Thrown exception
E38	Off-point	`"00::"`	Thrown exception
E39	Off-point	`":00:"`	Thrown exception

Partition	Boundary	Input	Expected outcome
E40	Off-point	`"::00"`	Thrown exception
E41	Off-point	`"::"`	Thrown exception
E42	Off-point	`"000000"`	Thrown exception
E43	Off-point	`"00:0000"`	Thrown exception
E44	Off-point	`"0000:00"`	Thrown exception

Filtered down to 31 tests.

Unit Tests

We created one test with the inputs of each line on the table.

The test function:

- `parseSecondsValidTest(String hours, String minutes, String seconds)`
- `parseSecondsInvalidTest(String timeStr)`
- `parseSecondsNullTest()`

The input generators (respectively):

- `parseSecondsValidInputs()`
- `parseSecondsInvalidInputs()`
- *No input generator*

All of these are present in the `ProjectTimeTest.java` file of the `test` package.

Results: All the test cases pass successfully.

Method 3

Method: `void adjustSecondsToday(int secondsToday)` in `Project.java` line 192.

Method's purpose: This function sets the *seconds overall* of a project as the value it receives as an argument (if valid).

Reason for selection: It is important that this function works as expected since other methods depend on it.

Identify the parameters

`secondsToday` - integer (`int`) representing the number of seconds today.

Characteristics of the parameters

The integer should represent a positive number between 0 and infinity.

Constraints

Negative time is not allowed - `secondsToday >= 0`

Partitions

- E1 - negative number
  - `secondsToday < 0`
- E2 - positive number (including 0)
  - `secondsToday >= 0`

Boundaries

Partition	On-point(s)	Off-point(s)
E1	`0`	`-1`
E2	`0`	`-1`

Generate tests

Partition	Boundary	Input
E1	On-point	0
E1	Off-point	-1
E2	On-point	0
E2	Off-point	-1

4 tests.

Filter redundant tests

Partition	Boundary	Input	Expected output
E1	On-point	0	0
E1	Off-point	-1	0

Filtered down to 2 tests.

Unit Tests

We created one test with the inputs of each line on the table.

The test function:

- `void adjustSecondsValidTest(int secondsToday)`
- `void adjustSecondsInvalidTest(int secondsToday)`

The input generators (respectively):

- `Stream<Arguments> adjustSecondsValidInputs()`
- `Stream<Arguments> adjustSecondsInvalidInputs()`

All of these are present in the `ProjectTest.java` file of the `test` package.

Results: all the tests pass successfully.