

Assignment 1

Template: Download Assignment 1.zip from LMS

Due Date: April 20, 2023 until 23:59 (upload .zip file to LMS for submission)

(Note: any delayed assignments must be submitted by April 27, 2023 until 23:59)

Create a timetable application to record class schedules of students. The schedules are organized by days of the week and classes. The days of the week are from Monday to Sunday and classes are from period 1 to 10.

Course class

Create a course class to store the information of the subjects a student is taking. It should comprise of the following fields and methods.

- String name variable is declared and used to store the name of the course.
- String professor variable is declared and used to store the name of the professor of the subject.
- String roomNumber variable is declared and used to store the name of the classroom where the class is held.
- boolean isValid variable is declared and used to determine whether a valid subject slot is available or not (by default it should be true).
- Create a constructor that initializes all variables (except isValid)
- Create a constructor that initializes only the name of the subject.
- Create a copy constructor used to make copies of the subject.
- Create an accessor method that gets the values of name, tutor, and room.
- Create a mutator method to initialize the values of tutor and room
- Create an equals() method that determines whether courses are equal. However, returns true only when the three values of name, tutor, and room are the same, otherwise returns false.
- Create a toString() method that returns only the name of the course.
- Create a getDetails() method that returns the course name, professor, and roomNumber as String

```
Name : DS  
Tutor : KIM  
Room : 507
```

TimeTable class

Create a class called TimeTable to store the information of the timetable.

- Make DAYS of Enum type. It is from Monday to Friday.

- Create a `timeTable` variable to store the subjects in the students' timetables. This variable is a two-dimensional array and stores objects of the `Course` class.
- Create an `Initialize()` method. This method can only be accessed within this class and initializes the timetable as follows.
 1. This array has 5 rows (days) and 10 columns (periods).
 2. All timetables in lecture 4 (column 3) are fixed to LUNCH. Here `isValid` field of the `Course` class should be set to false.
 3. All timetables in lecture 8 (column 7) are fixed to DINNER. Here `isValid` field of the `Course` class should be set to false.
 4. All other time slots are initialized with ----. Here `isValid` field of the `Course` class should be set to true.
- Create a constructor with no parameters. This constructor initializes all the `timeTable` arrays and calls the `initialize()` method.
- Create a method called `getSchedule()`. This method receives `DAY` and `period` as arguments, stores the information about the subject in that time period in a `String`, and returns it.

At that time you have :
 Name : DS
 Tutor : KIM
 Room : 507

(Hint: Call `getDetail()` of `Subject` class after `String` with "At that time you have:" value)

- Create a method called `setSchedule()`. This method receives `DAY`, `period`, `name`, `tutor`, and `room` as arguments. This method stores the subject information at the appropriate day and time. This method returns `true` on success and `false` on failure.
 NOTE: If the user tries to save the subject during LUNCH or DINNER, `false` is returned.
- Create a method called `toString()`. This method returns the format for the timetable as a `String` (see output sample below).

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
1	----	----	----	----	----
2	----	----	----	----	----
3	----	----	----	----	----
4	LUNCH	LUNCH	LUNCH	LUNCH	LUNCH
5	----	----	----	----	----
6	----	----	----	----	----
7	----	----	----	----	----
8	DINNER	DINNER	DINNER	DINNER	DINNER
9	----	----	----	----	----
10	----	----	----	----	----

- Create a method called `oneDaySchedule()`. This method receives a day of the week as an input and returns the timetable for that day in `String` format.

MONDAY

OOP

LUNCH

DINNER

- Create a subjectSchedule() method that receives name of class as input and returns day, period, tutor information as follows

Subject: AI
Day: FRIDAY
Lecture: 10
Professor: NOH
Room No: 605

- Create a setInputDate() method. This method receives the date of the day (year month day) it wants to know as input and stores the date information in the Calendar class object. And returns an object of the Calendar class.

Hint1: The input is a String type of '20230417'. Using the Wrapper class, the year, month, and day can be separated and stored in the Calendar class object.

Hint2: See the bottom of the document for how to use the Calendar class.

TimeTableApp class

This class has a main() method, and the main() method works as follows.

- Declare a Scanner to receive input from the keyboard.
- Create a TimeTable object.
- The following menus are implemented.
 1. Add a class to my time table
 2. View the class at a specific period
 3. View schedule of a specific class
 4. Time Table corresponding to input date
 5. Exit the program

```

=====Main Menu=====
(1) Add a class to my time table
(2) View the class at a specific period
(3) View schedule of a specific class
(4) TimeTable corresponding to input date
(5) Exit the program
=====Main Menu=====

```

Case 1

- Using the Scanner, the name, day, and period of the course are input, and the professor and room number are received.
- Check whether it is okay to add a subject using the setSchedule() method
- If these processes are completed successfully, the completed timetable is output. If not successful, the timetable is output without any additions.

```

=====Main Menu=====
(1) Add a class to my time table
(2) View the class at a specific period
(3) View schedule of a specific class
(4) TimeTable corresponding to input date
(5) Exit the program
=====Main Menu=====

```

```

1
Please enter the day to add the class
MONDAY
Please enter the period to add the class
2
Please enter the name of the class
OOP
Please enter the name of the tutor
PARK
Please enter the name of the room
813

```

Class successfully added

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
1	----	----	----	----	----
2	OOP	----	----	----	----
3	----	----	----	----	----
4	LUNCH	LUNCH	LUNCH	LUNCH	LUNCH
5	----	----	----	----	----
6	----	----	----	----	----
7	----	----	----	----	----
8	DINNER	DINNER	DINNER	DINNER	DINNER
9	----	----	----	----	----
10	----	----	----	----	----

Case 2

- Input day and period using Scanner.
- Use the getSchedule() method to get the subject of the corresponding time and output the information of the subject as shown below.

```
=====Main Menu=====
(1) Add a class to my time table
(2) View the class at a specific period
(3) View schedule of a specific class
(4) TimeTable corresponding to input date
(5) Exit the program
=====Main Menu=====
2
Please enter the day of the class
WEDNESDAY
Please enter the peroid of the class
6
At that time you have :
Name : DS
Tutor : KIM
Room : 507
```

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
1	----	----	----	----	----
2	OOP	----	----	----	----
3	----	----	----	----	----
4	LUNCH	LUNCH	LUNCH	LUNCH	LUNCH
5	----	----	----	----	----
6	----	----	DS	----	----
7	----	----	----	----	----
8	DINNER	DINNER	DINNER	DINNER	DINNER
9	----	----	----	----	----
10	----	----	----	----	AI

Case 3

- Input day and period using Scanner.
- Use the subject Schedule() method to get the subject information of the subject as shown below.

```

=====Main Menu=====
(1) Add a class to my time table
(2) View the class at a specific period
(3) View schedule of a specific class
(4) TimeTable corresponding to input date
(5) Exit the program
=====Main Menu=====

```

```

3
Please enter the class name

```

```

AI
Subject: AI
Day: FRIDAY
Lecture: 10
Professor: NOH
Room No: 605

```

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
1	----	----	----	----	----
2	OOP	----	----	----	----
3	----	----	----	----	----
4	LUNCH	LUNCH	LUNCH	LUNCH	LUNCH
5	----	----	----	----	----
6	----	----	DS	----	----
7	----	----	----	----	----
8	DINNER	DINNER	DINNER	DINNER	DINNER
9	----	----	----	----	----
10	----	----	----	----	AI

- However if the class is not present print “There is no class” as output

Case 4

- Print the entire timetable for a specific date using Scanner. EX) April 17, 2023
Enter ‘20230417’
- Using the day of the entered date, only the timetable for the corresponding day is output.
(see example)
- However, if the day of the week is a weekend (Saturday or Sunday), the string “There is no schedule” is output and exited.

```

=====Main Menu=====
(1) Add a class to my time table
(2) View the class at a specific period
(3) View schedule of a specific class
(4) TimeTable corresponding to input date
(5) Exit the program
=====Main Menu=====

```

4

Enter the date:

20230417

MONDAY

OOP

LUNCH

DINNER

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
1	----	----	----	----	----
2	OOP	----	----	----	----
3	----	----	----	----	----
4	LUNCH	LUNCH	LUNCH	LUNCH	LUNCH
5	----	----	----	----	----
6	----	----	DS	----	----
7	----	----	----	----	----
8	DINNER	DINNER	DINNER	DINNER	DINNER
9	----	----	----	----	----
10	----	----	----	----	AI

- End the program.

Calendar class (For reference)

The Calendar class is an abstract class that provides easy handling of date and time data in Java. This Calendar class includes various fields and methods for handling dates and times. To use the Calendar class, 'java.util.Calendar;' must be imported.

The fields and methods of the Calander class are shown below:

Fields	How to use	Explanation
static int YEAR	Calendar. YEAR	current year
static int MONTH	Calendar. MONTH	Current Month (Jan: 0)
static int DATE	Calendar. DATE	date of current month
static int WEEK_OF_YEAR	Calendar. WEEK_OF_YEAR	week of the current year
static int WEEK_OF_MONTH	Calendar. WEEK_OF_MONTH	wefek of the current month
static int DAY_OF_YEAR	Calendar.DAY_OF_YEAR	date of current year
static int DAY_OF_MONTH	Calendar.DAY_OF_MONTH	date of current month
static int DAY_OF_WEEK	Calendar.DAY_OF_WEEK	Current day of the week (Sunday: 1 , Saturday: 7)
static int HOUR	Calendar. HOUR	Current time (12-hour clock)
static int HOUR_OF_DAY	Calendar.HOUR_OF_DAY	Current time (24-hour clock)
static int MINUTE	Calendar. MINUTE	current minute
static int SECOND	Calendar. SECOND	current second

Methods	Explanation
static Calendar getInstance()	Creates a Calendar object with current date and time information.
boolean after(Object when)	Compared to when, it returns true if it is after the current date, or false otherwise.
boolean before(Object when)	Compared to when, returns true if it is before the current date, or false otherwise.
boolean equals(Object obj)	Compares whether the date value is the same and returns true or false.

int get(int field)	Returns a constant value corresponding to the field of the given value of the current object. This constant value has among the constants of the Calendar class.
Date getTime()	Converts the current object to a Date object.
long getTimeInMills()	Changes the object's time in units of 1/1000 of a second and returns it.
void set(int field, int value)	Sets a specific field of the current object to a different value.
void set(int year, int month, int date)	Sets the year, month, and day values of the current object to different values.
void set(int year, int month, int date, int hour, int minute, int second)	Sets the year, month, day, hour, minute, and second values of the current object to different values.
void setTime(Date date)	Creates the date and time information of the date object as the current object.
void setTimeInMills(long mills)	Sets the current object to the given parameter time in milliseconds.
int getActualMinimum(int field)	Returns the minimum value of a specific field of the current object.
int getActualMaximum(int field)	Returns the maximum value of a specific field of the current object.