OOP Class

Documentation on assignment1

Computer Science
2014004411 Si Wan KIM

Short description on the Subject code(1)

Subject.java 969 Bytes

```java

public class Subject {

        private String name, tutor, room;

        public Subject(String name, String tutor, String room){
                this.name =name;
                this.tutor = tutor;
                this.room = room;
        }

        public Subject(String name){
                this.name = name;
        }

        public Subject(Subject sub){
                this.name = sub.name;
                this.tutor = sub.tutor;
                this.room = sub.room;
        }

        public String getName(){
                return this.name;
        }
        public String getTutor(){
                return this.tutor;
        }
        public String getRoom(){
                return this.room;
        }

```

It's Copy constructor and it will be used on the getSchedule method in timeTable class to copy the argument's variable to this instance variable.

Accessor method

Short description on the Subject code(2)

```
33          public void setTutor(String tutor){
34                  this.tutor = tutor;
35          }
36          public void setRoom(String room){
37                  this.room = room;
38          }
39
40
41          public boolean equals(Subject sub){
42                  if(this.name.equals(sub.name) && this.tutor.equals(sub.tutor) && this.room.equals(sub.room)) return true;
43                  else return false;
44          }
45
46          public String toString(){
47                  return this.name;
48          }
49
50          public String getDetails(){
51                  return "Name: " + this.name + "\nTutor: " + this.tutor + "\nRoom: " + this.room + "\n";
            }
}
```

Mutator method

The equals method is used for compare the contents of instance equality not reference equality

Short description on the TimeTable code(1)

TimeTable.java 1.73 KB

```java

public class TimeTable {

        public enum DAYS{MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY};

        Subject[][] timeTable = new Subject[10][5];

        private void initialize(){

                for(int i = 0; i < 10; i++){
                        for(int j = 0; j < 5; j++){
                                timeTable[i][j] = new Subject("----");
                        }
                }
                for(int i =0;i<5;i++){
                timeTable[3][i] = new Subject("BREAK");
                timeTable[7][i] = new Subject("LUNCH");
                }
        }

        public TimeTable(){
                initialize();
        }

        public Subject getSchedule(String day, int period){
                String dayUpp ="";
                for(int i = 0;i<day.length();i++){
                        char c = day.charAt(i);
                        if(c>=97 && c<=122)
                                dayUpp += String.valueOf(c).toUpperCase();
                        else dayUpp += c;
                }
                Subject sub = new Subject(timeTable[period-1][DAYS.valueOf(dayUpp).ordinal()]);

                return sub;
        }
```

timeTable is created, type as a Subject two-dimensional array for store the info about subject instance.

It's code about when the input value of 'day' is consist of small letter, have to change it to upper letter.
(For the make enum also possible to read the down letter)

Makes the subject instance with the argument day and period. And it copy the info about class in on that day, on that period.

Short description on the TimeTable code(2)

```
39
40  public boolean setSchedule(String day, int period, String name, String tutor, String room){
41      String dayUpp ="";
42      for(int i = 0;i<day.length();i++){
43          char c = day.charAt(i);
44          if(c>=97 && c<=122)
45              dayUpp += String.valueOf(c).toUpperCase();
46          else dayUpp += c;
47      }
48      DAYS classday = DAYS.valueOf(dayUpp);
49      int dayNum = classday.ordinal();
50
51      if(period!=4 && period!=8){
52      timeTable[period-1][dayNum] = new Subject(name, tutor, room);
53      return true;
54      }
55      else return false;
56
57  }
58
```

It's also same code on the getSchedule method. (Makes down letter to upper letter)

Passing an enumeration constant name as a string returns an enumeration constant.

It returns the enum's order as a ordinal number.

Short description on the TimeTable code(3)

```java
59          public String toString(){
60                  String outcome = "   ";
61                  String wordLimit;
62                  for(DAYS e : DAYS.values()){
63                          outcome += String.format("%13s", e.name());
64                  }
65
66
67                  for(int i=0;i<10;i++){
68                          outcome += "\n";
69                          outcome += String.format("%2d", i+1);
70                          for(int j=0;j<5;j++){
71                                  wordLimit = timeTable[i][j].toString();
72                                  if(wordLimit.length()>9)
73                                  wordLimit = wordLimit.substring(0, 9);
74                                  outcome += String.format("%13s", wordLimit);
75                          }
76                          if(i==9) outcome += "\n";
77                  }
78
                    return outcome;
            }
    }
```

→ Add the String that we have to return in this string

→ Add the all enumeration constant's name in outcome String

→ Make the output value of className until length 9.

Short description on the TimeTableApp code(1)

TimeTableApp.java 1.96 KB

```java
import java.util.Scanner;

public class TimeTableApp {

        public static void main(String[] args) {
                // TODO Auto-generated method stub

                Scanner sc = new Scanner(System.in);
                TimeTable table1 = new TimeTable();
                System.out.println(table1);

                int i = 1;
                do{

                System.out.println("===================Main Menu===================");
                System.out.println("(1) Add a class to my time table:");
                System.out.println("(2) View the class at a specific time:");
                System.out.println("(3) Print the time table");
                System.out.println("(4) Exit the program");
                System.out.println("===================Main Menu===================");

                int command = sc.nextInt();
                String trush = sc.nextLine();

```

→ Make instance of TimeTable class

→ It used on when make while roop finish.

→ It used to erase the buffer because nextInt store the Enter in buffer.

Short description on the TimeTableApp code(2)

```
26              switch(command){
27
28              case 1:
29                      System.out.println("Please enter the day to add the class:");
30                      String daySet = sc.nextLine();
31                      System.out.println("Please enter the period to add the class:");
32                      int periodSet = sc.nextInt();
33                      String trush2 = sc.nextLine();
34                      System.out.println("Please enter the name of class");
35                      String nameSet = sc.nextLine();
36                      System.out.println("Please enter the name of tutor");
37                      String tutorSet = sc.nextLine();
38                      System.out.println("Please enter the name of Room");
39                      String roomSet = sc.nextLine();
40
41                      boolean isSuccess = table1.setSchedule(daySet, periodSet, nameSet, tutorSet, roomSet);
42
43                      if(isSuccess)
44                              System.out.println("Class successfully added");
45                      else
46                              System.out.println("Class was NOT successfully added");
47                      break;
48
49
50              case 2:
51                      System.out.println("Please enter the day to add the class:");
52                      String dayGet = sc.nextLine();
53                      System.out.println("Please enter the period to add the class:");
54                      int periodGet = sc.nextInt();
55                      String trush3 = sc.nextLine();
56                      System.out.println("At that time you have:");
57
58                      System.out.println(table1.getSchedule(dayGet, periodGet).getDetails());
59                      break;
60
```

Same with trush1

It store the info on the table1 and return true or false.

Same with trush1

Copy to the new instance of Subject class and call the getdetails method

Short description on the TimeTableApp code(3)

```
61
62                  case 3:
63                          System.out.println(table1.toString());
64                          break;
65
66                  case 4:
67                          i = 0;
68                          break;
69                  }
70          }while(i==1);
71
72
            }

}
```

To finish the while roop

Screen shot of program's output

- Main menu (when do not put the info.)

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

===================Main Menu===================
(1) Add a class to my time table:
(2) View the class at a specific time:
(3) Print the time table
(4) Exit the program
===================Main Menu===================
```

Screen shot of program's output

- Add a class

```
==================Main Menu==================
(1) Add a class to my time table:
(2) View the class at a specific time:
(3) Print the time table
(4) Exit the program
==================Main Menu==================
1
Please enter the day to add the class:
wednesday
Please enter the period to add the class:
6
Please enter the name of class
data structure
Please enter the name of tutor
james
Please enter the name of Room
room b
Class successfully added
==================Main Menu==================
(1) Add a class to my time table:
(2) View the class at a specific time:
(3) Print the time table
(4) Exit the program
==================Main Menu==================
```

```
==================Main Menu==================
(1) Add a class to my time table:
(2) View the class at a specific time:
(3) Print the time table
(4) Exit the program
==================Main Menu==================
1
Please enter the day to add the class:
monDay
Please enter the period to add the class:
3
Please enter the name of class
java
Please enter the name of tutor
Eric
Please enter the name of Room
room a
Class successfully added
==================Main Menu==================
(1) Add a class to my time table:
(2) View the class at a specific time:
(3) Print the time table
(4) Exit the program
==================Main Menu==================
```

```
==================Main Menu==================
(1) Add a class to my time table:
(2) View the class at a specific time:
(3) Print the time table
(4) Exit the program
==================Main Menu==================
1
Please enter the day to add the class:
Tuesday
Please enter the period to add the class:
8
Please enter the name of class
Mathmetics
Please enter the name of tutor
Morrison
Please enter the name of Room
room Edison
Class was NOT successfully added
==================Main Menu==================
(1) Add a class to my time table:
(2) View the class at a specific time:
(3) Print the time table
(4) Exit the program
==================Main Menu==================
```

Screen shot of program's output

- View the class at a specific time

```
===================Main Menu===================
(1) Add a class to my time table:
(2) View the class at a specific time:
(3) Print the time table
(4) Exit the program
===================Main Menu===================
2
Please enter the day to add the class:
wednesday
Please enter the period to add the class:
6
At that time you have:
Name: data structure
Tutor: james
Room: room b
```

```
===================Main Menu===================
(1) Add a class to my time table:
(2) View the class at a specific time:
(3) Print the time table
(4) Exit the program
===================Main Menu===================
2
Please enter the day to add the class:
MONDAy
Please enter the period to add the class:
3
At that time you have:
Name: java
Tutor: Eric
Room: room a
```

- Print the time table

```
===================Main Menu===================
(1) Add a class to my time table:
(2) View the class at a specific time:
(3) Print the time table
(4) Exit the program
===================Main Menu===================
3
        MONDAY      TUESDAY    WEDNESDAY     THURSDAY       FRIDAY
 1        ----         ----         ----         ----         ----
 2        ----         ----         ----         ----         ----
 3        java         ----         ----         ----         ----
 4       BREAK        BREAK        BREAK        BREAK        BREAK
 5        ----         ----         ----         ----         ----
 6        ----         ----    data stru         ----         ----
 7        ----         ----         ----         ----         ----
 8       LUNCH        LUNCH        LUNCH        LUNCH        LUNCH
 9        ----         ----         ----         ----         ----
10        ----         ----         ----         ----         ----
```