

Software Design Patterns

Practice 1: Design principles, analysis and architectural patterns (evaluated out of 50 points)

A company asks us to design part of the software to manage the events they organize at their facilities. Events can be entertainment, conference or workshop type, depending on the client's preferences. If the event is for entertainment, you can choose the type of music to play, and for the conference specify the speaker who will give it. In their facilities there are several rooms that they can use for an event and these rooms can be distributed in various ways: in the form of a theater, with all the chairs facing the same place; in the form of a cocktail, with high tables and not many chairs; or in the form of a banquet, with tables and chairs.

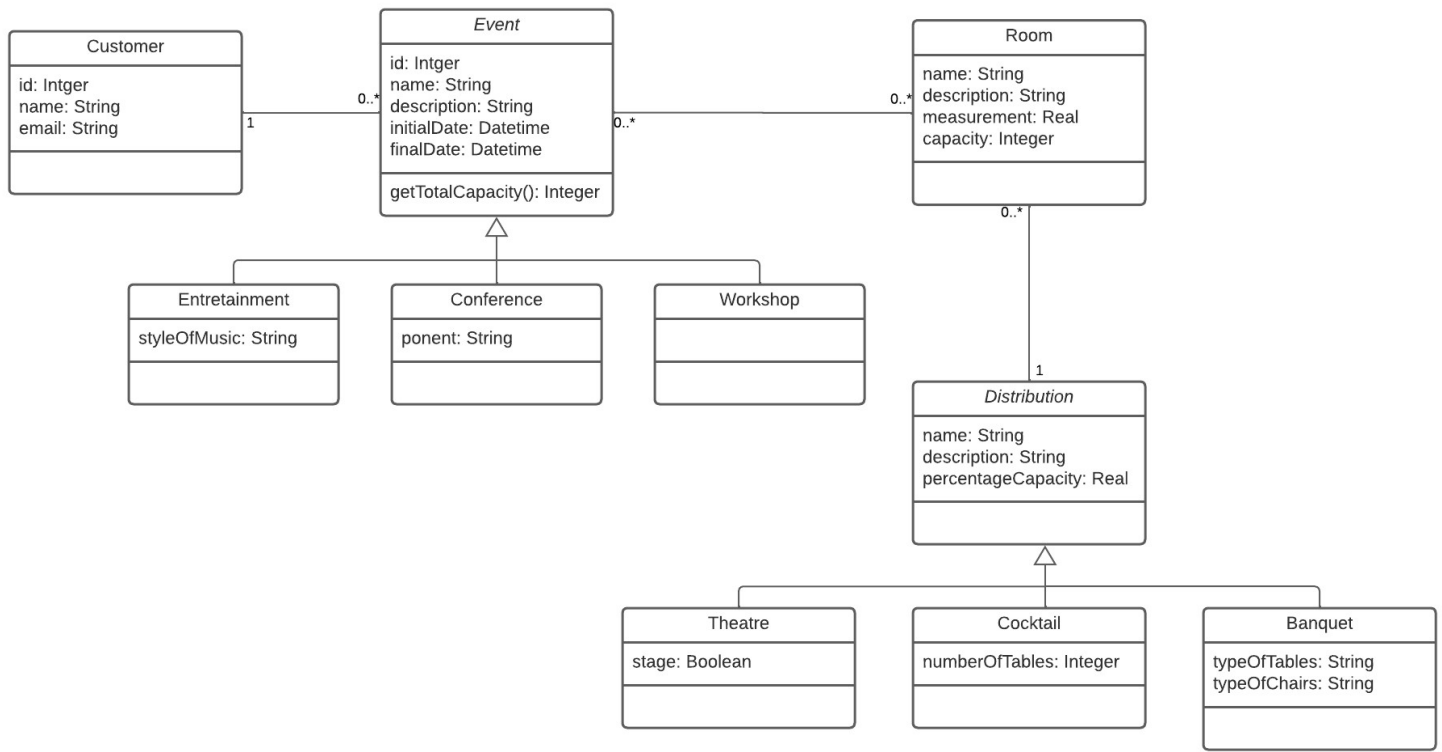
Each room has its area in m2 and the capacity in number of people, although depending on the distribution of the room its capacity changes, since having all the people standing cocktail style is not the same as having them sitting. For this reason in class *Distribution* we will have the percentage of people that fit, this percentage could vary depending on the room. Additionally, if the layout is theater type, a mini stage can be installed, which would reduce the capacity of the room by 10 people (what that stage occupies). For the cocktail type layout, you can specify how many tables we will have, and for the banquet type, you can choose the type of tables and chairs.

Another thing to keep in mind is that rooms cannot be occupied by overlapping events.

This software is a part of a larger system. We will ignore many elements that a real system would have. The reason for this is none other than to do a didactic practice, and that the expected difficulty and dedication are the desired ones.

We already have a prior analysis that we can use as a starting point, but which will have to be corrected and improved. Below you have the class diagram for this part of the software.

Diagram of classes:



Class keys:

- Customer: id
- Event: id
- Room: name
- Distribution: name

Integrity restrictions:

- The start date of an event must be earlier than its end date.
- A room cannot have overlapping events.
- All integers and reals are positive numbers.

To calculate the total capacity of an event, we have been asked to design the operation (`getTotalCapacity(): Integer`) of the **Event** class that returns the capacity of the event based on the rooms it has reserved and their distribution. Below is the pseudocode of a possible implementation of this operation:

```
public abstract class Event
```

```
{
    private Integer id;
    private String name;
    private String description;
    private Datetime initialDate;
    private Datetime finalDate;
    private Customer customer;
    private List<Room> rooms;

    public Integer getTotalCapacity()
    {
        Integer capacity = 0;
        foreach (Room r in rooms)
        {
            Integer roomCapacity = r.getCapacity();
            Distribution d = r.getDistribution();

            if(typeof(d)==Theatre)
            {
                Real percentCapacity = d.getPercentageCapacity()/100;
                capacity = capacity + (roomCapacity * percentCapacity);
                if(d.hasStage())
                {
                    capacity = capacity - 10
                }
            }
            else
            {
                Real percentCapacity = d.getPercentageCapacity()/100;
                capacity = capacity + (roomCapacity * percentCapacity);
            }
        }

        return capacity;
    }
}
```

Question 1 (9 points)

Statement

We reviewed the pseudocode for the `getTotalCapacity()`: Integer operation of the Event class and were concerned that some of the design principles were not met. Specifically, it asks:

- a) (3 points) Indicate whether or not this design satisfies the Law of Demeter and give reasons for your answer.
- b) (3 points) Indicate whether or not this design satisfies the No Repetition (DRY) design principle and give reasons for your answer.
- c) (3 points) Indicate whether or not this design satisfies the Open-Closed design principle and give reasons for your answer.

Question 2 (10 points)

Statement

Please propose a complete solution (also in pseudocode) that corrects the problems of the operation `getTotalCapacity()`: Integer of the Event class that you detected in the previous question and that, therefore, complies with the violated principles identified in question 1. Specifically, it asks:

- a) (4 points) Modify the class diagram by adding all the operations that you think are necessary for your solution.
- b) (6 points) Pseudocode of the `getTotalCapacity()` operation: Integer of the Event class complying with all design principles.

Question 3 (6 points)

Statement

Many clients have asked the company that they would like to manage several events in one, that is, having an event that is made up of different events. For example, an event that lasts three days is made up of three conference-type events and two work session events. Thus managing to integrate several events into one to manage them better. As this is a recurring request, the company proposes that we find a solution that allows us to represent this new event option that we will call congress in our system (you can start from the static analysis diagram of the statement). The conference event type cannot contain other conferences, only events of the other types.

Specifically, it asks:

- a) (2 points) Apply one of the analysis patterns explained in the materials to represent this information, explain which one and reason its convenience.
- b) (4 points) Propose a detailed solution in the form of a static analysis diagram (including new attributes that appear, multiplicities of associations and integrity constraints). Show only the classes that change with respect to the statement diagram.

Question 4 (6 points)

Statement

The company acquires a new space outside the country and realizes that depending on the country where it is located, clients do not fully understand the size of the room, since it is expressed in m2. This measurement is not the usual one in all countries, they find that they have to translate m2 into ft2 more and more for Anglo-Saxon clients who use the Anglo-Saxon system of units.

As there is increasing customer potential, they decide to be able to represent the unit of measurement in the system we are designing (you can start from the static analysis diagram of the statement).

Specifically, it asks:

- a) (2 points) Apply one of the analysis patterns that have been explained to the materials to represent this information, explain which one and reason its convenience.
- b) (4 points) Propose a detailed solution in the form of a static analysis diagram (including the new attributes that appear, the multiplicities of associations and integrity restrictions). Show only the classes that change with respect to the statement diagram.

Question 5 (8 points)

Statement

In order for the company to use this software, it needs a web application to access the data, enter new ones or modify existing ones. To define this presentation layer of our system, we decided to apply the Model, View, Controller (MVC) architecture pattern.

The use case proposed for this exercise is “Modify Event Description”. This use case allows the company from the “Event Data” screen to modify any of the attributes, save it and on the same screen the system returns the message “Successfully registered”.

It is requested:

- a) (5 points) MVC class diagram for this use case including the operations that are necessary (the operations must contain the parameters they need and the return types, if they have them) and a short description of what each of the operations do.
- b) (3 points) Sequence diagram or pseudocode of the use case, from the data of an event is displayed on the screen so that the company can modify any data until the system shows you the modified event and the message “Successfully registered”.

Question 6 (5 points)

Statement

We continue designing the architecture of the system, in the previous exercise we have used the Model-View-Controller pattern, which helps us connect the Presentation layer with the Domain layer. Now we want to continue designing that use case but reaching the Technical Services layer (Data Layer), from where we can access the database. Suppose we have a single database and we want to connect the Event class of our domain layer to the database to store the change described in the previous question.

We must generate the class diagram (corresponding to the domain layer and to the data layer) for this use case including the operations that are necessary (the operations have to contain the parameters they need and the return types, if they have them). The class diagram and operations have to be specific to this use case.

Question 7 (6 points)

Statement

To finish designing the 3-layer architecture of the system, using the three classic layers of many information systems:

1. Presentation
2. Domain
3. Technical services (Data)

We want to reflect on which layer each of the classes belongs to (and interfaces, if there was) that have appeared in the statement (and the solution you have developed so far):

- Classes in the class diagram of exercise 2.
 - Classes in the class diagram of exercise 5.
 - Classes in the class diagram of exercise 6.
- a) (2 points) Indicate which classes (and interfaces, if any) belong to the technical services layer (in our case, data layer). Explain your answer.
 - b) (2 points) Indicate which classes (and interfaces, if any) belong to the domain layer. Explain your answer.
 - c) (2 points) Indicates which classes (and interfaces, if any) belong to the presentation layer. Explain your answer.