

Computer Science 3620

Assignment 2 - Due Friday March 17 @11:55pm

For each of these problems you must provide a working IntelliJ project with the solution and a UML diagram describing the class design that you implemented.

Factory Method Pattern

We are implementing a Maze game that will contain multiple maze rooms. We will create an abstract *MazeRoom* class to be the superclass of a hierarchy of different maze rooms. Such a room could contain a variety of methods and fields such as a *description()* method that returns a string, briefly describing the room.

Rooms could also contain a variety of objects, but for simplicity let's focus on doors. Any given room will contain a List of *MazeDoors*, which is going to be an abstract superclass for a hierarchy of doors. There will be different types of doors and new types could be conceived as the designers of the game come up with new crazy ideas.

For now, we know that there will be simple doors which don't have a lock, but there will also be locked doors, and to unlock them you must provide a numeric code (integer) or a password (string). And we also know there will be doors with quizzes printed on them... and the solution to the quiz (lets for simplicity say it will be a string) will need to be provided to unlock the door. The doors with quizzes will need to include a functionality to 'show' the quiz. In the case of the doors without quizzes the answer will be hidden in other parts of the room.

There will also be many type of rooms, for now we have 'wooden rooms' which will contain normal doors, there will be 'steel rooms' which will contain locked doors that open with numeric code, there will be 'brick rooms' which will contain locked doors that open with a password (string) and there will be 'glass rooms' which will contain doors with quizzes.

We don't know which other rooms and doors could be created in the future thus we want to make our design as flexible as possible using the Factory Method pattern. The idea is that each subclass in the rooms hierarchy will know what type of doors it will contain and will be able to create them through a *createDoor()* method.

With the provided information design and implement class hierarchies for the rooms and doors. Rooms should be able to create their own type of doors in the constructor given the information of where (which wall, e.g. North, South, etc.) to put them. Depending on the type of door, doors need to provide the functionality to open (all doors), unlock (doors with locks and quizzes) and show the quiz (only doors with quizzes). In our example the functionality basically prints the action being executed, for instance, "The door opens". In a real game this functionality could be an animation or sound. What is important to take in consideration is that the functionality is the same for all the doors that can perform it. This means that all doors open in the same way or all unlockable doors unlock in the same way. Take this into consideration when efficiently designing the hierarchy for the doors.

As you may notice, you have only been given very general information about the game and what design decisions to make. Feel free to incorporate your own design decision and elements to the game as you see fit. For extra points feel free to apply other patterns. With this question you may include a document describing any relevant design decision and a description of any other pattern that you applied.

Submission Guidelines:

Submit your assignment as a single zip file through the Moodle link provided. Use the following naming format for your zip file:

lastname-as2.zip

The zip file should contain two folders, “problem1” and “problem2”, corresponding to the two problems above. Each of these folders should contain an IntelliJ project with the solution. **The folder must also include a UML class diagram describing the design solution.**