



SOFT2201/COMP9201

Tutorial 11

Singleton & Decorator & Façade

Question 1: Singleton Pattern

Singleton is one of the easiest patterns to implement once you know it, but is one of the easiest to use poorly - it effectively creates a 'fancy' global variable(s). Consider our stickman scenario, what opportunities might exist for the use of the singleton pattern? Model the possible use of singleton pattern in stickman scenario in a UML class diagram, and try to determine what the actual positive and negative consequences of the pattern are in this context.

Assignment Project Exam Help

Question 2: Decorator Pattern

A very interesting and common interview question related to decorator pattern is to identify the use of this design pattern in the Java API. The answer to this question is in the java.io package. Go to java.io package in Java API and see if you can identify the participants of decorator pattern there and explain why this pattern was used.

Consider our stickman scenario, see if you can come up with your own entity decorator. This could be an increased movement speed or a double points reward, etc. Model the possible use of decorator pattern in stickman scenario in a UML class diagram, and try to determine what the actual positive and negative consequences of the pattern are in this context.

Question 3: Façade Pattern

Façade is an easy pattern to implement. It is required not so much for low-scale design but modular architectural design. There might be a place for it in our stickman scenario - some of you naturally gravitated towards it due to the need for encapsulation and proper separation of concerns. See if you can identify any possible use of the façade pattern in stickman scenario. Model the possible use of this pattern in a UML class diagram, and try to determine what the actual positive and negative consequences of the pattern are in this context.