Class Project I

CIS 476/566 Software Architecture and Design patterns (Fall 2014)

Posted: 10/20/2014

Due: 11/10/2014

Suppose you are writing a program that performs remote diagnostics on computers made by a computer manufacturer called Richie Supersystems. Over time, Richie has produced four (4) generations of his computer having substantially different architectures: Richie, Advanced Richie, Super Richie, Ultra Richie. The core components you are interested in are: CPU, MMU (Memory Management Unit), and Motherboard. The core components used in these generations perform similar functions but are different. Each generation has its own programs (objects) for testing the CPU, MMU, and motherboard. To know which tests to run and how to interpret the results, you need to instantiate objects that correspond to each one of the core components in the computer being diagnosed. You can assume that the generation of the computer to be diagnosed is stored in a configuration file (text file). Because this situation fits the Abstract Factory pattern so well, you can use that pattern to organize the creation of objects that correspond to core computer components. You will also need to use the singleton pattern to ensure that the instance of each concrete factory is unique.

Turn-ins

1) Problem Statement – paragraph describing problem being solved.
2) Requirement specification – e.g. use case – describing scenarios to solve the problem. List the specifications, functional and non-functional requirements, assumptions, constraint
3) Testing Strategy – categories of what to test
4) Design - Give the UML diagram for the Richie Supersystems Diagnosis software. You should have one single diagram that includes both abstract factory and singleton patterns.
5) Algorithm
6) Test Plan – test cases within the categories of the test strategy
7) Give the code (in any language) of the Richie Supersystems Diagnosis software based on the UML class diagram and algorithm above. As output, you need to display three different messages (e.g., “Richie CPU”, “Richie MMU”, and “Richie Motherboard”) for the computer generation being diagnosed.
8) Test plan with results
9) Output screen shots (compile/build and execution output from test plan)
10) Final Status – what work and what doesn’t

Upload to class project 1 folder in Canvas before midnight of due date.