

Online-3 on Structural Design Patterns

Date: February 14, 2023

Problem Statement

Suppose that you run a company, named **Orange**, that manufactures 2 (two) different types of electronic devices, including **Notebook Computers** and **Smartphones**. Each of these two devices has the following common functionalities.

- Users can *store files* in the *Storage Module* of a device.
- Users can *interact* with the device via its *Interface Module*.
- Users can *communicate* among one another using the *Communication Module* of a device.

However, in order to facilitate these functionalities, an **Operating System** is required in the manufactured electronic device. An operating system comes with the aforementioned 3 (three) modules that enable the corresponding functionalities in a device. **Windows** and **Linux** are 2 (two) companies that manufacture operating systems in the market. The electronic devices, manufactured at Orange, must have the facility to replace the current operating system with a different operating system, at any time, to maximize the user satisfaction.

Now, you have to implement necessary classes, following an appropriate design pattern, to capture the scenario above. For the sake of simplicity, you may just print out the type of the functionality, the name of the module involved, and the name of the operating system used when implementing a method for a particular functionality of an electronic device.