Cerințe obligatorii

1. Pattern-urile implementate trebuie sa respecte definiția din GoF discutată în cadrul cursurilor și laboratoarelor. Nu sunt acceptate variații sau implementări incomplete.
2. Pattern-ul trebuie implementat corect în totalitate pentru a fi luat în calcul.
3. Soluția nu conține erori de compilare.
4. Pattern-urile pot fi tratate distinct sau pot fi implementate pe același set de clase.
5. Implementările care nu au legătura funcțională cu cerințele din subiect NU vor fi luate în calcul (preluare unui exemplu din alte surse nu va fi punctată).
6. NU este permisă modificare claselor/interfetelor primite.
7. Soluțiile vor fi verificate încrucișat folosind MOSS. Nu este permisă partajarea de cod între studenți. Soluțiile care au un grad de similitudine mai mare de 30% vor fi anulate.

Cerințe Clean Code obligatorii (soluția este depunctata cu câte 2 puncte pentru fiecare cerința ce nu este respectată) - maxim se pot pierde 4 puncte

1. Pentru denumirea claselor, funcțiilor, testelor unitare, atributelor și a variabilelor se respecta convenția de nume de tip Java Mix CamelCase.
2. Pattern-urile și clasa ce conține metoda main() sunt definite in pachete distincte ce au forma *cts.nume.prenume.gGrupa.pattern.model*, *cts.nume.prenume.Grupa.pattern.main* (studenții din anul suplimentar trec “as” în loc de gGrupa).
3. Clasele și metodele sunt implementate respectând principiile KISS, DRY și SOLID (atenție la DIP).
4. Denumirile de clase, metode, variabile, precum și mesajele afișate la consola trebuie sa aibă legătura cu subiectul primit (nu sunt acceptate denumiri generice). Funcțional, metodele vor afișa mesaje la consola care sa simuleze acțiunea cerută sau vor implementa prelucrări simple.

Se dezvoltă o aplicație software destinată activității gestionarilor unor săli de calculatoare.

4p. Se dorește implementarea unei aplicații care să ajute la gestionarea imaginilor virtuale folosite pentru calculatoarele din săli. Crearea unei imagini durează foarte mult deoarece trebuie instalate anumite aplicații în cadrul acestei imagini. Clasele vor implementa interfata ImagineVirtuala. Managerul sălii de calculatoare dorește ca în momentul în care o imagine a fost setată să aibă posibilitatea să fie copiată aceasta pe toate celelalte calculatoare fără a fi nevoie să mai apeleze constructorul de creare a unei imagini.

1p. Să se testeze soluția prin crearea a cel puțin patru obiecte prin intermediul modulului implementat

4p. Deoarece în aceste săli vor lucra studenți, se dorește ca soluția să creeze obiecte de tipul ImagineVirtuala intr-un mod facil, tinand cont de faptul ca atunci cand se creeaza o imagine sunt instalate anumite aplicatii iar altele sunt optionale. Astfel atunci cand obiectul de tipul ImagineVirtuala este creat nu mai pot fi instalate noi aplicatii trebuie ca acest obiect sa fie final. Realizati acest modul prin utilizarea unui design pattern adecvat.

1p. Să se testeze soluția prin crearea a cel puțin patru obiecte pentru prezentarea avantajelor design pattern-ului implementat.