

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.ume.prenume.gGrupa.pattern.model*, *cts.ume.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 unei spălătorii Auto.

- 5p.** În cadrul unei spălătorii Auto se dorește automatizarea procesului de spălare a mașinilor. În acest moment persoana care își spală mașina trebuie să folosească degresatorul pentru degresarea masinii, apoi foloseste un obiect de tip Spumant pentru adaugarea spumei pe mașină, după care foloseste laveta pentru ștergere. Să se implementeze un modul care va simplifica pentru client spălarea mașinii, el apelând doar functionalitati de tipul: doarSpălare(), spalareCuDegresare(), SpălareCuSpuma() etc.
- 3p.** Să se testeze soluția prin crearea a cel puțin cinci obiecte și 3 apeluri diferite din cadrul modulului implementat.
- 9p.** Spălătoria dorește să anunțe clienții care stau la coadă atunci când se eliberează un slot de spălat mașina, astfel încât aceștia să știe că se pot duce să ocupe acel slot. În cadrul mesajului transmis clienților li se aduce la cunoștință și numărul de persoane care se află la coadă. Să se implementeze modulul care permite înștiințarea tuturor clienților de la coadă cu numărul de persoane care așteaptă să spele mașina.
- 3p.** Să se testeze soluția prin crearea a cel puțin cinci clienți și trimiterea a cel puțin 3 înștiințări.