

Cerințe obligatorii

1. Pattern-urile implementate trebuie să 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 depunctată cu câte 2 puncte pentru fiecare cerință 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 în 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 să aibă legătura cu subiectul primit (nu sunt acceptate denumiri generice). Funcțional, metodele vor afișa mesaje la consola care să simuleze acțiunea cerută sau vor implementa prelucrări simple.

Se dezvoltă o aplicație software destinată unei spălătorii Auto.

- 5p.** Deoarece este foarte aglomerat la spălătorie managerul dorește să fie serviți doar clienții care au abonament. Astfel se dorește implementarea unui modul care atunci când primește un client căruia să îi spele mașina, aceasta să fie spălată doar dacă clientul respectiv are abonament valabil. În cazul în care abonamentul nu este valabil, clientului i se spune că nu sunt locuri disponibile.
- 3p.** Să se testeze soluția prin crearea a cel puțin cinci obiecte și verificarea situațiilor din cadrul modulului implementat.
- 9p.** Pentru o spălătorie automată se dorește implementarea unui modul prin care clientul stabilește atunci când ajunge la spălătorie cum să fie spălată mașina: doar exterior, doar interior, exterior și interior, etc. După ce clientul a decis cum să fie spălată mașina, aceasta intră în procesul de spălare conform cerinței clientului. Să se implementeze acest modul care permite modificarea modului de spălare a unei mașini care ajunge la spălătorie.
- 3p.** Să se testeze soluția prin crearea a cel puțin cinci obiecte și spălarea în 3 moduri diferite.