

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 în totalitate corect 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 modificarea claselor 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 8 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.nume.prenume.gNrGrupa.denumire_pattern*, *cts.nume.prenume.gNrGrupa.main* (studenții din anul suplimentar trec "as" în loc de gNrGrupa);
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 unui magazin online.

- 5p.** În cadrul unui magazin online de pantofi, se dorește implementarea unui modul de filtrare a produselor în funcție de criteriile de selecție adăugate de client (interfața ICautare). Implementarea trebuie să permită în timp extinderea modulului prin adăugarea de noi criterii de selecție. De asemenea, trebuie să existe posibilitatea de rearanjare a ordinii de aplicare a criteriilor. Dacă un client nu parametrizează un anumit criteriu de selecție, atunci nu se va ține cont de acesta. Ordinea criteriilor de selecție se stabilește la nivel de magazin, nu la nivel de fiecare client în parte. Însa, clientul poate parametriza cautarea prin alegerea sau nu a unui criteriu de a fi adăugat.
- 3p.** Pattern-ul este testat în main() prin adăugarea unui număr de minim 7 produse în cadrul magazinului. Ulterior, se aplică minim 2 căutări în cadrul magazinului și se afișează rezultatul furnizat de căutări.
- 1p.** Să se motiveze în scris alegerea design pattern-ului care modelează cel mai bine situația enunțată. Motivarea trebuie să conțină elementele cheie care denotă utilizarea respectivului design pattern.
- 5p.** Se dorește implementarea unui modul suplimentar care să permită clientului să aleagă între generarea unei parametrizări implicite de căutare de produse sau opțiunea de a-și seta singur parametrii de căutare. Parametrizarea clientului se va realiza folosind citirea de la tastatură. Se utilizează interfața IClient și ICautare.
- 3p.** Pattern-ul este testat în main() prin utilizarea unui client și setarea la nivelul sau a celor două opțiuni de parametrizare a unei căutări.

1p. Sa se motiveze in scris alegerea design pattern-ului care modeleaza cel mai bine situatia enuntata. Motivarea trebuie sa contina elementele cheie care denota utilizarea respectivului design pattern.