

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.numa.prenume.gGrupa.pattern.model*, *cts.numa.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 unui service auto.

4p. În cadrul aplicației de gestiune a activității unui service auto de tunare mașini, se dorește implementarea unui modul pentru facilitarea procesului de colantare mașini. Crearea unui obiect de tip Sticker este un proces consumator întrucât sunt necesare informații suplimentare deținute de producătorii mașinilor iar dimensiunile exacte diferă de la un model la altul. Unicitatea dimensiunilor este reprezentată de modelul mașinii alături de anul de fabricație. Constatându-se faptul că service-ul gestionează de multe ori modele de mașini identice, să se orienteze implementarea modulului în scopul optimizării timpului alocat creării specificațiilor necesare unui Sticker. Crearea unui Sticker primește ca parametru un obiect de tipul unei clase care implementează interfața IMasina. Consultând producătorii de mașini, este construit obiectul de tip Sticker cu dimensiunile impuse.

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

3p. În cadrul aplicației service-ului, angajații trebuie să creeze obiecte pentru componentele auto colantate: Usa, Capota, Bara, Portbagaj etc. Pentru fiecare tip de componentă este folosită o clasă aferentă componentei respective care implementează interfața IComponentă. Să se implementeze modulul care îi va ajuta pe inginerii service-ului în procesul de creare a obiectelor din familia componentelor auto.

1p. Să se testeze soluția prin crearea a cel puțin patru obiecte din cel puțin două tipuri diferite din familia componentelor auto.

Timp de lucru: 50 minute