

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 activității de susținere a unui examen.

- 5p.** Se dorește implementarea unui modul care atunci când are loc examenul din sesiunea normală, acest examen să fie susținut doar de către studenții care au un punctaj pe activitatea semestrială de minim 2 puncte. Funcționalitatea de susținere examen primește ca parametru un student. În cazul în care studentul nu a acumulat minim 2 puncte, acesta este înștiințat că va susține examenul în sesiunea de restanțe neputând să susțină în acest moment. În sesiunea de restanțe acest modul nu va mai fi utilizat, deoarece atunci nu mai este condiție de intrat în examen
- 3p.** Să se testeze soluția prin crearea a cel puțin cinci obiecte și verificarea situațiilor din cadrul modulului implementat.
- 9p.** Profesorul dorește să aibă posibilitatea să anunțe studenții atunci când apar modificări de ultim moment. De exemplu dorește să îi anunțe dacă a fost schimbată sala în care se susține examenul, sau dorește să îi anunțe dacă se modifică ora de susținere a examenului. De asemenea dorește să anunțe toți studenții dacă sunt clarificări cu privire la vreun subiect. Să se implementeze un modul în cadrul aplicației care permite profesorului să înștiințeze toți studenții de la cursul lui atunci când are nevoie.
- 3p.** Să se testeze soluția prin crearea a cel puțin cinci studenți și trimiterea a cel puțin 3 înștiințări.