

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 modificare 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.

- 5p.** În cadrul unui spital, este permis accesul vizitatorilor indiferent de numărul de vizite efectuate. Odată cu apropierea sărbătorilor de iarnă, s-a observat o aglomerare în spital din cauza unui număr prea mare de vizitatori. Astfel, s-a restricționat numărul de vizitatori la doar 1 vizitator per pacient per zi. Se dorește implementarea unui modul intermediar ce permite accesul în saloane conform noilor restricții impuse. Implementarea trebuie să țină cont de interfața *ISpital* primită.
- 3p.** Pattern-ul este testat în main() prin crearea unui număr de minim 5 vizitatori care doresc să intre în saloanele cu pacienți. Se va testa funcționalitate de resetare a vizitelor la sfârșitul zilei și testarea prin intrarea în salon a unui vizitator la același pacient, dar în zile diferite.
- 9p.** În cadrul unui joc online, utilizatorii au posibilitatea să aleagă un caracter dintr-un set predefinit (enumerație) și să îl customizeze după bunul plac cu următoarele componente: culoare păr, set arme, culoare piele, etc. Odată selectat tipul de caracter, acesta nu mai poate fi modificat în acest proces de customizare. Se dorește implementarea unui modul care să permită acest lucru utilizatorilor precum și introducerea unor facilități suplimentare precum și revenirea într-o singură stare anterioară procesului de customizare pe care utilizatorul o poate alege în orice moment al jocului. Fiecare astfel de caracter este definit de interfața *ICaracter*.
- 3p.** Să se testeze soluția prin crearea unui caracter și modificarea acestuia precum și revenirea într-o stare anterioară salvată de către utilizator. Afișarea stării obiectului se va realiza folosind metoda `getDescriereCaracter()`.