

Observații

1. Implementările de Design Patterns trebuie să fie conform definiției GoF discutată în cadrul cursului și laboratoarelor. Implementările incomplete sau variațiile non-GoF ale patternurilor nu sunt acceptate și nu vor fi punctate.
2. Sunt luate în considerare doar implementările complete și corecte, implementările parțiale nu se punctează.
3. Soluțiile ce conțin erori de compilare nu vor fi evaluate.
4. Patternurile pot fi implementate separat sau utilizând același set de clase.
5. Implementările generale de design patterns, care nu sunt adaptate cerinței și nu rezolvă problema cerută nu vor fi luate în considerare.
6. Clasele/interfețele primite nu pot fi modificate.
7. Soluțiile vor fi verificate cu software antiplagiat. Partajarea de cod sursă între studenți nu este permisă. Soluțiile cu un nivel de similitudine peste 30% vor fi anulate.

Cerințe Clean Code (nerespectarea lor va duce la depunctarea cu 2 puncte pentru fiecare cerință) - se pot pierde 8 puncte în total

1. Clasele, funcțiile, atributele și variabilele vor fi denumite conform convenției Java Mix CamelCase.
2. Fiecare pattern precum și clasa ce conține metoda `main()` vor fi definite în pachete diferite de forma `cts.num.prenume.g<numarul_grupeii>.pattern.<denumire_pattern>`, respectiv `cts.num.prenume.g<numarul_grupeii>.main` (studenții de la recuperare vor utiliza „recuperare” în loc de numărul grupei).
3. Clasele și metodele nu trebuie să încalce principiile KISS, DRY, YAGNI sau SOLID.
4. Numele claselor, metodelor, variabilelor și mesajele afișate la consolă trebuie să fie strict legate de subiectul curent (numele generice nu sunt acceptate). Mesajele afișate trebuie să simuleze scenariul cerut.
5. Nu sunt permise „magic numbers” sau valori hardcodate. Formatarea codului sursă trebuie să fie cea standard.

Realizați o aplicație software pentru o televiziune de știri.

3p. Televiziunea deține o aplicație prin care poate transmite live buletine de știri pe diverse platforme de social media. Cu toate că același conținut este transmis pe toate platformele, redacția se plânge de faptul că pierde mult timp cu crearea fiecărei transmisiuni în parte. Știind că o transmisiune live implementează interfața `Live`, utilizați un design pattern ce rezolvă problema agenției. Țineți cont și de faptul că odată create, transmisiunile live pot avea detalii diferite (exemplu: altă listă de comentarii).

2p. Testați implementarea prin crearea a 3 transmisiuni live aferente aceluiași eveniment pe 3 platforme de streaming diferite. Demonstrați faptul că adăugarea unui comentariu pe una dintre platforme nu va face ca acesta să apară pe altă platformă.

3p. Televiziunea postează pe site știri din 3 mari categorii: economie, politică și sport. Utilizați un design pattern ce permite crearea de știri din cele 3 categorii știind că toate tipurile de știri derivează interfața `NewsArticle` și că tipul de știre este dat la momentul execuției programului împreună cu titlul știrii și textul ei. După o perioadă televiziunea dorește modificarea aplicației astfel încât să poată posta știri din cele 3 categorii și pe rețelele de socializare. O știre postată pe o rețea de socializare va avea titlu, un text de maxim 100 de caractere (dacă textul furnizat depășește această dimensiune, se vor păstra primele 100 de caractere) și link-ul spre site.

2p. Testați în main implementarea prin crearea a minim 5 știri, din categorii diferite, postate atât pe site, cât și pe rețelele de socializare. Categoria, titlul, textul și link-ul (dacă e cazul) vor fi introduse de la tastatură.