

S1 – Cinematograf

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 și complet (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 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) - se pot pierde maxim 8 puncte

1. Pentru denumirea claselor, funcțiilor, atributelor și a variabilelor se respecta convenția de nume de tip Java Mix CamelCase discutată;
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 cinematograf.

- 5p.** Un cinematograf utilizează la casierie o aplicație de generare a biletelor de acces în sălile de cinema. Biletele conțin detalii despre film (ex.: Titlu, Regie, Actori principali, Compania producătoare, O scurta descriere etc.), reclame (imagini cu plasare de produse) și detalii care indică data/ora și locurile care vor fi ocupate de posesorul biletului. Din motive tehnice, generarea primului bilet pentru un anumit film durează destul de mult (peste 10 secunde). Odată generate și completate, biletele trebuie salvate pentru a putea fi efectuată validarea la intrarea în sală. Ținând cont că pentru un film toate biletele conțin aceleași informații (detalii legate de film și reclamele) și că singurele informații diferite sunt data/ora și locul ocupat, să se implementeze o modalitate prin care să se realizeze generarea biletelor conform descrierii. Implementarea se va realiza plecând de la interfața **Ticket** atașată acestui enunț.
- 3p.** Să se testeze soluția în metoda main() prin generarea de bilete, astfel: se tastează numărul de bilete ce urmează a fi generate și apoi se citesc de la tastatură data, ora și locul pe scaun (citirea se realizează într-o buclă, data, ora și locul fiind separate cu spațiu - pentru fiecare set de date introduse se generează un bilet).
- 9p.** Cinematograful extinde activitatea prin posibilitatea rezervării online a biletelor. Aplicația de rezervare online a fost dezvoltată independent de aplicația prezentă la casierie și din această cauză imprimarea biletelor rezervate online nu poate fi realizată (formatului acestora este diferit). Implementați o modalitate prin care fără a schimba sistemul software existent, să se

S1 – Cinematograf

poată realiza imprimarea biletelor rezervate online folosind același tip de imprimantă, existent deja în toate casieriile. Implementarea se va realiza plecând de la interfața **OnlineBooking** atașată acestui enunț.

- 3p.** Pattern-ul este testat în `main()` prin definirea și utilizarea a cel puțin 3 obiecte: unul din familia **Ticket**, unul din familia **OnlineBooking** și unul ce demonstrează implementarea corectă a pattern-ului precedent (imprimarea unui bilet rezervat online se realizează corect, conform formatului acceptat de imprimantă).

```
public interface Ticket {  
    /*  
        This method returns order's details in A6 landscape printable format -  
        comma-separated values:  
        Movie title: Fast and furious 9, Date: 04.05.2022, Time: 16:45 UTC+3,  
        Seat: 3F, ...  
    */  
    public String print();  
}
```

```
public interface OnlineOrder {  
    /*  
        This method returns order's details in A4 portret printable format -  
        each info in on a different line:  
        Movie title: Fast and furious 9  
        Date: 04.05.2022  
        Time: 16:45 UTC+3  
        Seat: 3F  
        ...  
    */  
    public String showOrderDetails();  
}
```