**Cerințe obligatorii**

1. Pattern-urile implementate trebuie sa respecte definitia din GoF discutată în cadrul cursurilor și laboratoarelor. Nu sunt acceptate variații sau implementării incomplete.
2. Pattern-ul trebuie implementat corect în totalitate corect pentru a fi luat în calcul
3. Soluția nu conține erori de compilare
4. Testele unitare sunt considerate corecte doar dacă sunt implementate conform cerințelor și dacă metodele sunt corectate corespunzător pe baza lor
5. Pattern-urile pot fi tratate distinct sau pot fi implementate pe același set de clase

**Cerințe Clean Code obligatorii (soluția este depunctata cu câte 5 puncte pentru fiecare cerința ce nu este respectată**) - maxim se pot pierde 15 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, Test Case-urile, Excepțiile și clasa ce contine metoda main() sunt definite in pachete distincte ce au forma *cts.nume.prenume.gNrGrupa.teste*, *cts.nume.prenume.gNrGrupa.patternX*, *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)

Se dezvoltă o aplicație software destinată unei platforme de mesagerie de tipul WhatsApp, messenger, etc.

**10p.** Dezvolți o soluție software care să permită profesorilor și studenților să comunice prin intermediul mesajelor în cadrul unor grupuri de discuții. În momentul înregistrării, utilizatorii trebuie să își definească un profil definit de o serie de atribute: nume (obligatoriu), prenume (obligatoriu), tip (obligatoriu și poate fi profesor sau student), vârsta (opțional), gen (opțional), grupa (opțional), facultatea (opțional), cursul coordonat (opțional). Găsiți o soluție care să le permită programatorilor să gestioneze eficient crearea acestor obiecte complexe.

De asemenea, având în vedere că în aplicație vor exista 2 tipuri de profil predominante (profesor sau student), soluția trebuie să faciliteze crearea rapidă a acestor tipuri standard prin utilizarea unui catalog/director de profile standard. Pentru profesor setarea cursului coordonat este obligatorie, iar pentru student este obligatorie inițializarea grupei și a facultății. Pentru un profil particular de profesor sau student aplicația permite utilizatorilor să îl configureze după cum doresc.

**5p.** Pattern-ul este testat în main() prin crearea a cel puțin un profil de profesor și a unui profil de student, cu diferite setări. De asemenea, este exemplificată și crearea rapidă a unui profil standard de profesor și a unui profil standard de student.

**5p.** Soluția permite participanților să comunice prin mesaje text ce sunt transmise pe diferite grupuri. Mesajele sunt gestionate la nivel de grup (de ex WhatsApp) și transmiterea unui nou mesaj va genera afișarea acestuia pe interfața tuturor clienților abonați la acel grup. Un client/utilizator abonat la un grup are interfața ***ITransmitereMesaj***. Clienții au posibilitatea să se aboneze la diferite grupuri și de asemenea să iasă când doresc dintr-un grup existent.

**5p.** Să se testeze soluția prin definirea a minim 2 grupuri la care să fie abonați diferiți clienți. Un client trebuie să fie abonat la minim 2 grupuri simultan. Transmiteți câte un mesaj pe fiecare grup. Dezabonați un client și apoi retrimiteți un alt mesaj.

**6p.** Dându-se clasa *TestGrila* și restricțiile definite prin comentarii să se implementeze teste unitare (toate testele unitare aferente unei metode sunt implementate într-un TestCase) care să cuprindă:

1. un unit test care să realizeze o testare *Boundary* pentru **setDificultate()** **(1.5p)**
2. un unit test care să testeze o testare *Error Condition* pentru **setPunctajMaxim() (1.5p)**
3. un unit test de tip *Existence* pentru metoda ***calculTotalPuncte()* (1.5p)**
4. un unit test de verificare de tip *Cardinality = 3* pentru metoda ***calculTotalPuncte*()**;**(1.5p)**

**2p.** Să se implementeze o suită de teste care să conțină DOAR câte o metodă, la alegere, din fiecare test case

**2p.** Prin testele implementate sau prin adăugarea de teste noi sa se testeze *setDificultate* ()asigurând un code coverage de 100% pentru această metodă.