**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)
4. Denumirile de clase, metode si atribute au semnificație (au legatura cu subiectul) în contextul subiectului. NU se folosesc denumiri generice/teoretice

Se dezvoltă o aplicație software destinată administrării unui cartier de blocuri.

**10p.** Dezvolți o soluție software pentru asociațiile de proprietari dintr-un cartier de blocuri. În cadrul cartierului sunt mai multe meseriași (instalatori, electricieni, etc) care repară și îndreaptă lucrurile din tot cartierul. Atunci când apare o problemă, locatarii solicită intervenția acestora și solicitarea lor este salvată într-o coadă de așteptare, iar atunci când unul dintre meseriașii cartierului este liber, va prelua această problemă și o va rezolva. Problemele care pot apărea sunt descrise de **SolicitareRezolvareProblema**. Să se implementeze modulul care permite gestionarea cozii de comenzi de probleme către meseriașii cartierului. Clasa dată poate fi modificata astfel incat sa faciliteze implementarea soluției. Pentru meseriași se poate defini o interfata la alegere.

**5p.** Pattern-ul este testat în main() prin crearea a cel puțin patru solicitări diferite plasate către meseriași și preluarea de către doi meseriași a câte una dintre comenzile salvate.

**5p.** Independent de cerința anterioara, se dorește implementarea unei soluții care să permită locatarilor unui bloc să aleagă modul în care sunt publicate anunțurile din bloc. Asociația de proprietari poate decide ca anunțurile să fie printate la avizier sau să fie trimise prin email sau prin sms, etc. Odată soluția aleasă ea va fi menținută până la ședința următoare. Pentru a implementa soluția se va defini o interfață care că conțină minim o acțiune specifică acestui scenariu.

**5p.** Să se testeze soluția prin exemplificarea a minim 2 scenarii în care locatarii să aleagă metode diferite de informare în cadrul ședinței. Sa se testeze sistemul prin publicarea unui anunț astfel incat sa fie evidenta modalitatea de transmitere.

**6p.** Dându-se clasa *Avion* 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 *Right* pentru **setGreutateIncarcata()** **(1.5p)**
2. un unit test care să testeze o testare *Conformance* pentru **setGreutateIncarcata() (1.5p)**
3. un unit test de tip *Cardinality = 5* pentru ***getNumarLocuriOcupate* (1.5p)**
4. un unit test de verificare de tip *Performance* pentru ***getNumarLocuriOcupate***; se consideră că metoda trebuie sa verifice 1000 de locuri în mai puțin de 1 secundă **(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 *getPretBilet*() asigurând un code coverage de 100% pentru această metodă.