**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 care să le permită să gestioneze detaliile legate de costurile de întreținere aferente unei luni pentru fiecare proprietar de apartament (interfața *IProprietarApartament*). Aceste date sunt stocate în fiecare lună într-o clasă ce stochează numărul apartamentului, costul apei calde, costul caldurii, costul gazelor, etc. Soluția trebuie să permită salvarea acestor date la finalul lunii astfel încât să poată oferi în orice moment informații cu privire la istoricul acestor costuri pentru fiecare proprietar/apartament. Soluția trebuie să permită recuperarea situației pentru un apartament și o lună indicată.

**5p.** Pattern-ul este testat în main() prin simularea soluției pentru cel puțin 2 apartamente pentru care s-a salvat situația (cel puțin 2 luni diferite). Soluția trebuie demonstreze ca se poate recupera situația costurilor pentru un anumit apartament, pentru o lună indicată.

**5p.** Independent de cerința anterioara, blocul are la intrare un interfon prin care este solicitat/controlat accesul conform interfeței *IInterfon*. Deoarece copiii din cartier abuzează de acest serviciu și deranjează locatarii, se dorește adăugarea în sistem a unei camere video care să verifice dacă apelantul este adult sau copil. Această camera va fi conectată la sistemul existent (nu se modifica interfonul) și va controla dacă se poate fi folosit interfonul sau nu. Dacă camera detectează un copil atunci interfonul nu va suna la apartamentul apelat.

**5p.** Să se testeze soluția prin exemplificarea scenariului în care sistemul funcționează fără cameră și a scenariului în care se adaugă camera.

**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.