Student: Frunzescu Vlad-Teodor Grupa: 223 Mail: [frunzescu.vlad@gmail.com](mailto:frunzescu.vlad@gmail.com)

Alegeti afirmatiile (raspunsuri multiple) corecte privind codul de mai jos:

**public class** Decan{

**private** Decan() {}  
**static final private** Decan ***instance*** = **new** Decan();

**static final** Decan getDecan()

{

**return *instance***;

}

**public void** aruncaExceptie(){

throw new ValidationException();

*//presupunem ca nu avem acces la implementarea clasei ValidationException*

}

}

1. Obiectul este incarcat in memorie doar atunci cand este nevoie de acesta.
2. Obiectul este incarcat in memorie in cazul in care poate va fi folosit.
3. Metoda aruncaExceptie() poate fi folosita in afara pachetului in care a fost definita.
4. Clasa poate fi instantiata cel mult o data.
5. Clasa custom ValidationException poate mosteni Error.
6. Clasa custom ValidationException poate mosteni oricare dintre Exception si RuntimeException.

Corect: b), d), e)

Explicatie: Metoda getDecan() returneaza o unica instanta a clasei si nu are modificatorul de acces “public” specificat in antet, deci metoda va fi de tipul “default” sau “package-private”. Asta inseamna ca si metoda aruncaExceptie(), chiar daca este declarata “public”, se poate utiliza doar folosind obiectul de tip Decan. In concluzie, si metoda aruncaExceptie() poate fi apelata doar in interiorul pachetului.

In antetul functiei afiseazaExceptie() nu apare clauza “throws” insotita de numele exceptiei, deci se intelege ca ValidationException apartine categoriei exceptiilor neverificate (unchecked) si poate mosteni RuntimeException sau Error. Raspunsul f) este gresit pentru ca ValidationException NU poate mosteni Exception.