Examen scris la Programare Orientată Obiect – R1

Programul Postuniversitar de Pregătire în Informatică, 20.04.2019

Timp de lucru: 1 oră și 30 minute

Nume Prenume: _____

Fiecare subiect valorează 1 punct. Toate subiectele se rezolvă pe această foaie de examen.

1. Scrieți o clasă NumarRational care modelează un număr rațional de forma numărător / numitor, unde numărător și numitor sunt numere întregi. Va fi suportată operația de adunare.

2. Dați 3 exemple de perechi de clase între care ar avea sens relația de **moștenire**.

3. Scrieți o secvență de cod care folosește conceptul de **funcție lambda**.

4. Scrieți o secvență de cod în care implementați o interfață.

5. Fie trei clase A, B, C. A moștenește B, iar B este compus din mai multe C-uri. Reprezentați această situație printr-o diagramă UML.

6. Scrieți clase / interfețe care pot fi folosite pentru modelarea următorului scenariu si a altora asemănătoare: **Gigel mănâncă un măr curățat**. Argumentați pe scurt alegerile făcute. Se acceptă cod sau diagramă UML.

7. Fie o clasă **LaserPrinter** cu o metodă **getTonerLevels()** care afișează folosind System.out.println nivelul de încărcare al fiecărui toner. Scrieți cel puțin un lucru greșit în această abordare.

8. Dacă avem mai multe blocuri **catch** pentru același bloc **try**, cum trebuie ordonate acestea?

9. Evidențiați conceptul de **polimorfism** folosind următoarele clase / interfețe: **ICirculabil** și **Autostrada**.

10. Încercuiți numai variantele corecte:
    a. Șablonul **Observer** se folosește pentru a executa cod pe diferite fire de execuție (thread-uri).
    b. Un obiect este același lucru cu o referință.
    c. Java este un limbaj orientat obiect pur.
    d. O interfață este o clasă abstract pură.
    e. O clasă abstractă nu poate fi moștenită.
    f. O interfață nu poate fi instanțiată.
    g. Java suportă moștenire multiplă.
    h. O clasă poate implementa mai multe interfețe.
    i. Memoria alocată cu **new** trebuie eliberată cu **delete**.