Academia Tehnică Militară "Ferdinand I“
Facultatea de Sisteme Informatice şi Securitate Cibernetică

Paradigme de programare (în Java)

Lab 4/Curs 4- Programare orientată pe obiecte (OOP)

Col(r) Traian Nicula

Se creează un proiect Maven cu numele ***lab4*** și se salvează în folder-ul cu numele studentului. Pentru fiecare exercițiu se va crea câte o clasă de test (Exercise1 etc.) cu metoda statică *main*, precum și alte clase cerute de exerciții.

Se rezolvă următoarele exerciții:

1. Scrieți un program Java care creează o listă. Adăugați la listă 8 elemente de tip String și tipăriți lista. (**0.5p**)
2. Creați o listă la care adăugați 8 numere întregi. Tipăriți elementele folosind bucla *for-each*. (**0.5p**)
3. Scrieți un program care creează o listă cu 5 elemente și returnează elementul de la indicele 3. (**0.5p**)
4. Scrieți un program care creează o listă cu 10 elemente de tip String. Sortați în ordine naturală și tipăriți lista. (**0.5p**)
5. Scrieți un program care conține o listă cu 5 culori. Căutați și returnați indicele uneia dintre culori. (**1p**)
6. Scrieți un program care creează o listă înlănțuită la care adaugă 8 elemente de tip String. Tipăriți prin iterare (folosind metodele *descendingIterator(), hasNext() și next()*) lista în ordine inversă. (**1p**)
7. Scrieți un program care creează o listă înlănțuită cu 3 elemente. Adăugați un element nou la capătul listei. (**0.5p**)
8. Scrieți un program care creează o listă înlănțuită cu 5 elemente. Obțineți și tipăriți primul și ultimul element din listă. (**0.5p**)
9. Scrieți un program care creează o listă înlănțuită cu 5 elemente. Tipăriți lista inițială. Amestecați elementele listei (cu metoda *shuffle()* din clasa *Collections*) și tipăriți-o din nou. (**1p**)
10. Scrieți un program care creează un ***TreeSet*** la care adaugă 4 elemente. Tipărește setul, apoi tipărește primul și ultimul element. (**1p**)
11. Scrieți un program care creează un ***HashMap*** cu 10 perechi cheie/valoare (Integer/String). Obțineți un set de vizualizare a mapărilor (*entrySet()*) și folosiți-l pentru tipărirea iterativă a perechilor cheie/valoare. (**1p**)
12. Scrieți un program care: (**2p**)
    - conține o clasă cu numele **Square** având: (**0.5p**)
        a. câmpurile *int side* și *String color*
        b. un constructor cu câmpurile *side* și *color*
        c. metode getter și setter
        d. metoda *getArea()* care returnează aria pătratului (*double*)
    - conține clasa **SquareComp** care implementează interfața *Comparator<Square>* și va fi folosită pentru compararea ariilor pătratelor (**0.5p**)

- conține o listă la care se adaugă 5 instanțe ale clasei **Square** (2 cu laturi egale, restul diferite) și tipărește lista (**0.5p**)
- sortează lista cu comparatorul **SquareComp** și afișează rezultatul. (**0.5p**)

Temă pentru acasă:

13. Scrieți un program care creează ***TreeSet*** la care adaugă 10 elemente numere întregi din intervalul 1 .. 100. Afișați elementul din set mai mic sau egal cu 50. Scoateți și afișați primul element din set. (**0.5p**)
14. Scrieți un program care creează un ***HashMap*** cu 5 perechi cheie/valoare (Integer/String). Căutați și afișați câteva valori în funcție de cheie. (**0.5p**)
15. Scrieți un program care: (**1p**)
    - conține o enumerație publică **Color** alcătuită din culorile RED, BLUE, YELLOW, GREEN.
    - conține o clasă cu numele **Circle** având:
        a. câmpurile *int radius* și *Color color*
        b. un constructor cu câmpurile radius și color
        c. metode getter și setter.
    - conține o listă cu 5 instanțe ale clasei având raze și culori diferite
    - conține o metodă statică *void printColor(Circle circle)* care folosește instrucțiunea *switch* pentru a tipări culoare cercului în litere mici
    - parcurge lista cu un ***Iterator*** și folosește metoda *printColor* pentru a tipări culoarea fiecărui cerc.