Poo Laboratoare[1]

Contents

[1] http://www.google.ro/

Laborator4

Programare Orientata pe Obiecte: Laborator 4

1 Clase noi si clase derivate

```
class ListAsVector
{
   Vector v;
   public ListAsVector()
   {
      v = new Vector();
   }
   public boolean add(Object o)
   {
      return v.add(o);
   }
   ....
}

class ListAsVectorDerivat  extends Vector
{
   public boolean add(Object o)
   {
      return super.add(o);
   }
...
}
```

1.1 Principii POO

1.1.1 Moştenirea

Permine definirea unor noi clase care pastreaza caracteristicile, datele, funcţiile unei alte clase (denumita clasa de baza). Pornind de la aceasta proprietate putem defini clase noi care extind comportamentul unei clase de baze, facand dintr-o clasa generala, una particulara pentru ceea ce avem nevoie.

1.1.2 Supraîncarcarea

Ofera posibilitatea de a redefini metode din clasele de baza. Se observa astfel, cum am redefinit în clasa SetAsVectorDerivat metoda add, în schimb am putut utiliza metoda add din clasa de baza Vector datorita lui super (super.add(o) apeleaza metoda add din Vector).

1.1.3 Observaţii

Clasele Vector si Hashtable se gasesc în pachetul java.util. Pentru a vedea metodele acestor clase accesaţi documentaţia!

2 Probleme de laborator

2.1 Problema 1

(2 puncte) Sa se defineasca o clasa SetAsVector pentru o mulţime de obiecte realizata ca vector neordonat cu elemente distincte, în doua variante:

- clasa SetAsVector derivata din clasa Vector (pachetul java.util)
- clasa SetAsVector conţine un obiect de tip Vector

Metode din clasa Vector care vor fi implementate în clasa SetAsVector:

- boolean add(Object)
- boolean remove(Object)
- boolean contains(Object)
- String toString()

Rezultatul metodelor add şi remove este true daca operaţia ceruta a reusit.

Sa se scrie un program pentru crearea unei mulţimi de şiruri prin adaugari succesive, eliminarea unor elemente şi afişare dupa fiecare operaţie, folosind clasa SetAsVector.

2.2 Problema 2

(2 puncte) Sa se defineasca o clasa SortedVector derivata din clasa Vector pentru vectori ordonaţi de obiecte, cu urmatoarele metode redefinite:

- addElement(Object) : adauga un element la sfarşitul vectorului şi apoi ordoneaza cu metoda Collections.sort
- insertElementAt(Object, int) : adauga într-o poziţie data şi reordoneaza

Sa se scrie un program pentru operaţii cu un vector ordonat de şiruri: adaugari succesive de elemente (obiecte Întreger) cu afişare dupa fiecare modificare.

2.3 Problema 3

(2 puncte) Sa se defineasca o clasa ÎntSet pentru o mulţime de întregi, cu metodele add, remove, contains şi "toString" care apeleaza metode din clasa BitSet(care reprezinta un vector de biţi; astfel bitul i este true daca i aparţine mulţimii). Clasa ÎntSet conţine o variabila de tip BitSet şi apeleaza metode ale acestei clase.

Metode din clasa BitSet

- boolean get(int bitÎndex) : întoarce true sau false daca bitul cu indexul bitÎndex a fost setat
- void set(int bitÎndex, boolean value) : seteaza bitul de la indexul bitÎndex la valoarea value

Pentru mai multe detalii despre clasa BitSet consultaţi APÎ-ul Java.

2.4 Problema 4

(2 puncte) Sa se defineasca o clasa Graph pentru grafuri reprezentate prin liste de adiacenţe, ca o clasa derivata din clasa Vector. Listele de adiacenţe sunt tot vectori (obiecte Vector). Nodurile grafului se numeroteaza de la 1. Se va defini un constructor cu argument numar de noduri din graf.

Metode:

- size : numarul de noduri din graf)
- addArc : adauga un arc la graf
- isArc : verifica daca exista arc intre doua noduri date

- toString : lista de arce din graf

Sa se scrie un program pentru creare de graf prin adaugari succesive de arce (pe baza unor perechi de numere date in program), afişare arce si afişare grad interior şi exterior pentru fiecare nod (numar de arce în şi din nod).

2.5 Problema 5

(1 punct) a) Sa se defineasca o clasa HSet pentru o mulţime realizata ca tabel de dispersie, cu metodele: add, contains, toString, size. Clasa HSet va fi derivata din clasa Hashtable (pachetul java.util), în care cheia şi valoarea asociata vor fi egale (cheile sunt elementele mulţimii).

(1 punct) b) Sa se defineasca o clasa HSet pentru o mulţime de obiecte realizata ca tabel de dispersie, folosind clasa Hashtable (pachetul java.util). Metode care vor fi implementate: add, remove, contains, toString. Clasa HSet conţine o variabila de tip Hashtable şi apeleaza metodele acesteia.

Sa se scrie un program pentru adaugarea unor numere la mulţime şi afişarea mulţimii.