Εργαστηριακή άσκηση #10

Θέμα: Διαπροσωπείες(interfaces) και αφηρημένες κλάσεις

Η εργαστηριακή αυτή άσκηση αποσκοπεί στην εξοικείωση με την χρήση αφηρημένων κλάσεων και διαπροσωπειών (interfaces). Κεντρικές εξακολουθούν να είναι οι έννοιες του πολυμορφισμού και της κληρονομικότητας.

Θα κατασκευαστεί μία απλή διαπροσωπεία (SetInt) η οποία ορίζει τις λειτουργίες ενός συνόλου από ακεραίους. Κατόπιν, η διαπροσωπεία θα υλοποιηθεί μέσω ενός διανύσματος από την κλάση ArraySetInt. Η κλάση ArraySetInt θα υλοποιεί και τις Διαπροσωπείες Comparable και Cloneable της βιβλιοθήκης java.lang. Η άσκηση ολοκληρώνεται με την υλοποίηση μιας απλής εφαρμογής που χρησιμοποιεί την διαπροσωπεία SetInt και την υλοποίηση ArraySetInt.

Ένα σύνολο του τύπου SetInt αποτελείται αποκλειστικά από ακέραιους που προέρχονται από το σύνολο [1..maxSize], όπου η τιμή του maxSize προσδιορίζεται κατά την κατασκευή του συνόλου από τον κατασκευαστή μίας από τις κλάσεις που το υλοποιεί. Κάθε στοιχείο εμφανίζεται μόνο μία φορά στο σύνολο.

1. Χρησιμοποιώντας το BlueJ, δημιουργήστε το έργο setOfInt.
2. Δημιουργήστε την διαπροσωπεία SetInt η οποία δηλώνει τις παρακάτω μεθόδους με την επιθυμητή λειτουργία που περιγράφεται:

`void add(int k)`	Προσθέτει τον ακέραιο k στο σύνολο. $1 \le k \le$ maxSize
`boolean isIn(int k)`	Ελέγχει εάν ο ακέραιος k ανήκει στο σύνολο. $1 \le k \le$ maxSize.
`void remove(int k)`	Διαγράφει τον ακέραιο k από το σύνολο. $1 \le k \le$ maxSize.
`int cardinality()`	Επιστρέφει την πληθικότητα του συνόλου.
`void union(SetInt s)`	Ενώνει το σύνολο με ένα άλλο σύνολο από ακεραίους. Το σύνολο s πρέπει να έχει τον ίδιο μέγιστο αριθμό στοιχείων με το τρέχων σύνολο για να είναι η ένωση σωστά ορισμένη.
`void setEmpty()`	Ελέγχει εάν το σύνολο είναι άδειο.
`int getMaxSize()`	Επιστρέφει την μέγιστη δυνατή πληθικότητα του συνόλου.

3. Να οριστεί η κλάση ArraySetInt η οποία υλοποιεί τη διαπροσωπεία SetInt. Αρχικά, εισάγετε τον παρακάτω κώδικα που περιγράφει τα πεδία και τον κατασκευαστή.

```
public  class ArraySetInt implements SetInt
{
    private boolean set[];
    private int maxSize;
    private int cardinality;


    ArraySetInt(int size)
    {
        maxSize=size;
        set=new boolean[maxSize];
        for (int i=0; i<maxSize; i++)
            set[i]=false;
        cardinality=0;
    }
}
```

4. Μεταφράστε την κλάση ArraySetInt. Παρατηρήστε ότι ο μεταφραστής βρίσκει συντακτικά λάθη και δίνει το διαγνωστικό μήνυμα: «ArraySetInt is not abstract and does not override abstract method getMaxSize() in SetInt». Το μήνυμα αυτό είναι αναμενόμενο λόγω του ότι δεν έχουμε προσδιορίσει ακόμη την υλοποίηση των μεθόδων που ορίζονται στην διαπροσωπεία SetInt. Προσθέστε την λέξη “abstract” στην δήλωση της κλάσης (μετά το public). Τώρα η κλάση μεταφράζεται κανονικά.
5. Να υλοποιηθεί η μέθοδος add(int k) όπως παρακάτω. Παρατηρήστε ότι όπως προκύπτει από την υλοποίηση, το γεγονός ότι το στοιχείο k ανήκει στο σύνολο υποδηλώνεται από το γεγονός ότι το στοιχείο set[k-1] του διανύσματος έχει την τιμή true.

```
public void add(int k)
{
    if ( ! set[k-1])
    {
       set[k-1]=true;
```

```
            cardinality++;
        }
    }//add
```

6. Να υλοποιηθούν και κατόπιν να ελεγχθεί η σωστή λειτουργία των υπολοίτων μεθόδων που ορίστηκαν στην διαπροσωπεία. Για τον έλεγχο της σωστής λειτουργίας επιβάλλεται η δημιουργία αντικειμένων. Το γεγονός ότι η κλάση ArraySetInt δηλώθηκε ως abstract (για να υλοποιηθεί και να μεταφραστεί τμηματικά) δεν επιτρέπει την δημιουργία αντικειμένων. Διαγράφτε την λέξη abstract από την δήλωση της κλάσης έτσι ώστε η δημιουργία αντικειμένων να καταστεί δυνατή.
   Σημείωση: Με τον τρόπο που περιγράφηκε, ο έλεγχος γίνεται αφού υλοποιηθούν όλες οι μέθοδοι. Αυτό δεν είναι ικανοποιητικό μιας και απαιτείται ο έλεγχος κάθε μεθόδου να γίνεται αμέσως μετά την δημιουργία της. Αυτό μπορεί να επιτευχθεί υλοποιώντας όλες τις μεθόδους έτσι δώστε το σώμα τους να είναι άδειο, αλλά να επιστρέφουν τον τύπο του αποτελέσματος που ορίζουν. Έτσι, η πραγματική ανάπτυξη του κώδικα και ο έλεγχος ορθότητας γίνεται για κάθε μέθοδο αμέσως μετά την ανάπτυξη της.

7. Να υλοποιηθεί η μέθοδος toString() (η οποία κληρονομείται από την κλάση Object) όπως παρακάτω:

```
public String toString()
{
    String result="{";
     boolean foundFirst=false;
     int i=1;
     for (i=1 ; i<=maxSize && (!foundFirst); i++)
         if (set[i-1]==true)
         {
            result=result + i;
            foundFirst=true;
         }

     for (; i<=maxSize ; i++)
         if (set[i-1]==true)
            result=result + "," + i ;
     result= result + "}";
     return result;
}//toString
```

8. Επιθυμητό είναι να μπορεί να οριστεί η διάταξη των συνόλων. Η διάταξη τους καθορίζεται με όμοιο τρόπο με αυτόν της διάταξης των String. Έτσι, το σύνολο {1,13,26} είναι μικρότερο από τα σύνολα {2,5}, {1,15,17}, {1,13,32} και {1,13,26, 45}. Το κενό σύνολο είναι μικρότερο από όλα τα αλλά σύνολα. Η java υποστηρίζει την σύγκριση αντικειμένων μέσω της διαπροσωπεία Comparable της βιβλιοθήκης java.lang. Να επεκτείνετε τη δήλωση της κλάσης ArraySetInt έτσι ώστε να υλοποιεί (implements) τη διαπροσωπεία Comparable. Να υλοποιήσετε την μέθοδο compareTo().

9. Επιθυμητό είναι επίσης να μπορούμε να δημιουργήσουμε αντίγραφα (deep cloning) ενός αντικειμένου τύπου ArraySetInt. Η java υποστηρίζει την δημιουργία αντιγράφων αντικειμένων μέσω της διαπροσωπεία Cloneable της βιβλιοθήκης java.lang. Να επεκτείνετε τη δήλωση της κλάσης ArraySetInt έτσι ώστε να υλοποιεί (implements) τη διαπροσωπεία Cloneable. Να υλοποιήσετε την μέθοδο clone() η οποία κληρονομήθηκε από την κλάση Object.

10. Θα χρησιμοποιήσουμε την διαπροσωπεία SetInt και την κλάση ArraySetInt για να λύσουμε το παρακάτω απλό πρόβλημα: Να εκτιμηθεί το πόσες φορές η γεννήτρια τυχαίων αριθμών (java.util.Random) πρέπει να κληθεί έτσι ώστε να δημιουργήσει όλους τους ακέραιους από το 1 έως το 1000. Η χρήση της γεννήτριας γίνεται μέσω της κλήσης της μεθόδου nextInt(). Να γραφεί η κλάση TestRandom η οποία περιέχει την μέθοδο estimate() η οποία επιστρέφει τον αριθμό των κλήσεων της μεθόδου nextInt() που απαιτήθηκαν. Στον κώδικα της TestRandom το όνομα της κλάσης ArraySetInt εμφανίζεται μόνο μια φορά, όταν καλείται ο κατασκευαστής της. Όλες οι μεταβλητές δηλώνονται ώστε να είναι τύπου SetInt.

11. Ο τρόπος εκτίμησης του αριθμού των κλήσεων της nextInt() να τροποποιηθεί έτσι ώστε:

    i. Η μέθοδος estimate() να δέχεται μια παράμετρο (έστω n) η οποία προσδιορίζει το μέγεθος του συνόλου, και
    ii. για κάθε σύνολο μεγέθους n, το «γέμισμα» του να επαναλαμβάνεται 1000 φορές και να υπολογίζεται ο μέσος όρος των κλήσεων ανά γέμισμα.

12. Οργανώστε ένα πείραμα όπου το μέγεθος του συνόλου (n) είναι μεταβλητό. Πειραματιστείτε για διαδοχικές τιμές του n που διαφέρουν κατά 10 μεταξύ τους.

13. Να μελετήσετε την διαπροσωπεία java.util.Set και τις σχετικές με αυτή κλάσεις. Προσπαθήστε να καταλάβετε σε όσο το δυνατό περισσότερο βάθος την λειτουργία των κλάσεων.