

Στην αρχή της εργασίας κάνουμε τα import. Αρχικά 'import java.io.PrintWriter' ώστε να μπορούμε να γράφουμε στο αρχείο εξόδου. Μετά με 'import java.io.FileInputStream' έχουμε την δυνατότητα να διαβάζουμε το αρχείο εισόδου, με 'import java.io.File' έχουμε πρόσβαση codes.dat για να μπορέσουμε να το διαβάσουμε. Μετά με το 'import java.io.IOException' για να κάνουμε έπειτα throw IOException στην main. Μετά με τα 'import java.io.Bitset' και 'import java.io.HashMap' γιατί κάνουμε χρήση Bitset και HashMap. Τέλος με το 'import java.io.Scanner' σκανάρουμε το αρχείο codes.dat (1).

Στην αρχή κανουμε στην main κάνουμε ένα throw IOException. Στην συνέχεια φτιάχνουμε, ένα FileInputStream δίνοντας του το αρχείο εισόδου ως args[0], δηλαδή ως την πρώτη παράμετρο από την γραμμή εντολών, για να μπορέσουμε να το διαβάσουμε και ένα PrintWriter δίνοντας του για αρχείο εξόδου το args[1], δηλαδή την δεύτερη παράμετρο από την γραμμή εντολών, για να μπορέσουμε να γράψουμε στο αρχείο. Μετά χρησιμοποιούμε το File και το scanner για να διαβάσουμε το αρχείο codes.dat (2).

Στην συνέχεια φτιάχνω ένα HashMap με key το String και value το BitSet (3). Μέσα σε ένα while με συνθήκη 'scanner.hasNextLine()', δηλαδή για όσο υπάρχει επόμενη γραμμή στο αρχείο codes.dat, κάνουμε τους απαραίτητους ελέγχους για να πάρουμε τα στοιχεία που χρειαζόμαστε από το codes.dat. Στην αρχή φτιάχνω ένα String fileLine που θα έχω κάθε φορά την κίνηση γραμμή του αρχείου. Μετά φτιάχνω ένα πίνακα encodedData[] από String (4). Στο πρώτο if κάνουμε έλεγχο με τη συνθήκη 'fileLine.contains("::")', δηλαδή αν στην γραμμή που βρίσκεται μέσα στο fileLine κάθε φορά υπάρχει το ':' δύο φορές συνεχόμενες. Αυτό είναι ιδιαίτερη περίπτωση για τον χαρακτήρα ':'. Μέσα στο if χρησιμοποιούμε το 'encodedData = fileLine.split("::");', δηλαδή να χωρίσει την γραμμή με το '::' σε δύο στοιχεία και να τα βάλει στον πίνακα encodedData και μετά βάζουμε στο encodedData[0] το ':' (5). Στην συνέχεια με ένα else κάνουμε πάλι 'split()' αλλά μόνο με ένα ':'. Όλο αυτό γίνεται για να πάρουμε από το codes.dat ξεχωριστά τις κωδικοποιήσεις και ξεχωριστά τους χαρακτήρες. Οπότε μέσα στο encodedData στην πρώτη θέση έχει τον χαρακτήρα και στην δεύτερη την κωδικοποίηση (6).

Στην συνέχεια με if κάνουμε ελέγχους για να δούμε ότι ο πίνακας encodedData έχει πάρει σωστά τα δεδομένα από το codes.dat. Μετά κάνουμε 'encodedData[1] = encodedData[1].replaceAll("[\\[,\\], ]", "");' όπου στην ουσία χρησιμοποιούμε το replaceAll για να αντικαταστήσουμε τα '[', ']' , ', ' με κένο για να έχουμε την κωδικοποίηση σε σωστή μορφή (7).

Έπειτα φτιάχνουμε το BitSet με μέγεθος το μήκος της κωδικοποίησης κάθε φορά αφού είμαστε ακόμα στο while που διαβάζουμε γραμμή-γραμμή το codes.dat. Μετά με ένα 'for(int i = 0; i < encodedData[1].length(); i++)' βάζουμε τις κωδικοποιήσεις στο BitSet. Αναλυτικότερα με ένα if με συνθήκη 'encodedData[1].charAt(i) == '1'' περνάμε την κωδικοποίηση χαρακτήρα-χαρακτήρα και αν είναι 1 πηγαίνει και βάζει στο BitSet τον χαρακτήρα ως true δηλαδή 1, αλλιώς αν δεν είναι 1 το βάζει ως false δηλαδή 0 (8). Τέλος το while κλείνει αφού βάλουμε στο HashMap όλα τα key, δηλαδή τους χαρακτήρες ως encodedData[0], και όλες τα values, δηλαδή τις κωδικοποιήσεις ως BitSet (9).

Μετά με ένα while διαβάζουμε ανά χαρακτήρα το αρχείο εισόδου. Με το 'String key = Character.toString((char) c);' κάνουμε το χαρακτήρα που διαβάζει String ώστε να το βάλουμε στο HashMap. Με το 'if(!encodedDataMap.containsKey(key))' ελέγχουμε αν το συγκεκριμένο key υπάρχει μέσα στο HashMap αλλιώς να τυπώσει μήνυμα Error (10). Στην συνέχεια πηγαίνουμε και βάζουμε στο bitsLength να προσθέτει το μήκος του BitSet κάθε φορά για να ξέρουμε το συνολικό αριθμό από bits. Με το 'for

(int i = 0; i < encodedDataMap.get(key).length(); i++)' βάζουμε στο 'String bit' την κωδικοποίηση. Ουσιαστικά αν στο BitSet έχει true στην θέση i θα βάλει 1, αλλιώς 0. Και μετά με το 'outputResult += bit' φτιάχνουμε την συνολική κωδικοποίηση (11).

Τέλος βάζουμε στο String finalResult το bitsLength και το outputResult και το εκτυπώνουμε στο outputFile με το 'outputFile.write(finalResult)' και κλείνουμε τα stream προς τα αρχεία εισόδου και εξόδου (12).

```
(1) : import java.io.PrintWriter;
      import java.io.FileInputStream;
      import java.io.File;
      import java.io.IOException;
      import java.util.BitSet;
      import java.util.HashMap;
      import java.util.Scanner;

(2) : FileInputStream inputFile = new FileInputStream(args[0]);
      PrintWriter outputFile = new PrintWriter(args[1]);

      File codesDatFile = new File("./codes.dat");
      Scanner scanner = new Scanner(codesDatFile);

(3) : HashMap<String, BitSet> encodedDataMap = new HashMap<>();

(4) : while (scanner.hasNextLine()) {
      String fileLine = scanner.nextLine();

      String encodedData[];

(5) : if (fileLine.contains("::")) {
      encodedData = fileLine.split("::");
      encodedData[0] = ":";

(6) : else {
      encodedData = fileLine.split(":");
      }

(7) : if (encodedData.length != 2) {
      continue;
      }

      if (encodedData[0].length() <= 0) {
      continue;
      }

      if (encodedData[1].length() <= 0) {
      continue;
      }
```

```

        }

        encodedData[1] = encodedData[1].replaceAll("[\\[,\\], ]",
""");

(8) :   BitSet bitset = new BitSet(encodedData[1].length());
        for (int i = 0; i < encodedData[1].length(); i++) {
            if (encodedData[1].charAt(i) == '1') {
                bitset.set(i, true);
            } else {
                bitset.set(i, true);
                bitset.set(i, false);
            }
        }

(9) :

        encodedDataMap.put(encodedData[0], bitset);

(10) :   while ((c = inputFile.read()) != -1) {

        String key = Character.toString((char) c);

        if (!encodedDataMap.containsKey(key)) {
            System.out.println("Error,no value for the key :" + key);
            continue;
        }

(11) :   bitsLength += encodedDataMap.get(key).length();
        for (int i = 0; i < encodedDataMap.get(key).length(); i++) {

            String bit = encodedDataMap.get(key).get(i) ? "1" : "0";

            outputResult += bit;
        }

(12) :   String finalResult = bitsLength + "\n" + outputResult;
        outputFile.write(finalResult);

        outputFile.close();
        inputFile.close();

```

Τα ονοματά μας:

Παναγιώτης Κωλέτσης  
Παναγιώτης Πετεινάρης  
Παναγιώτης Χάρος