Στην αρχή αντιγράψαμε την κλάση TreeNode η οποία ειναι ίδια με την κλάση TreeNode του προηγούμενου μέρους. (1)

Στην συνέχεια κάναμε "import java.io.Serializable;" μαζί με "implements Serializable" στις κλάσεις μας ώστε να μπορούμε να κάνουμε το deserialization του δέντρου που φτιάξαμε στο προηγούμενο μέρος και περάσαμε με τον αντίστοιχο τρόπο ώστε να μην χαλάσει η δομή. Επίσης για την ολοκλήρωση του διαβάσματος του δέντρου από το αρχείο φτιάξαμε ένα FileInputStream και ObjectInputStream και μέσω της μεθόδου "readObject()" κάνοντας πάντα τα αντίστοιχα imports, όταν τελειώσαμε με το read του object κλείσαμε το αρχείο με "writeFile.close();" έχοντας βάλει "throws IOException, ClassNotFoundException" στην main μας . Μετά φτιάξαμε μια μεταβλητή τύπου TreeNode που την είπαμε root και περάσαμε το δέντρο. (2)

Αποφασίσαμε οτι θα χρησιμοποιήσουμε την έτοιμη στίβα της java "ArrayDeque" οπότε φτιάξαμε μια στίβα "static ArrayDeque<String> dq = new ArrayDeque<>();" κάνοντας "import java.util.ArrayDeque;" (3). Ύστερα υλοποιήσαμε μια κλάση την "printHuffmanCoding(root, writeFile);" με 2 παραμέτρους έναν τύπου TreeNode για τον κόμβο που είμαστε κάθε φορά και έναν PrintWriter για το αρχείο την οποία και καλέσαμε αναδρομικά για να γράψουμε την κωδικοποιήση στο αρχείο ,κάναμε import το PrintWriter και αφού φτιάξαμε και το αρχείο "codes.dat" ανοίξαμε το αρχείο που θέλαμε για να γράφουμε τα δεδομένα .

Στην αρχή η "printHuffmanCoding" κάνει έλεγχο με ένα if για το αν τα παιδιά του root ειναι null, δηλαδή αν ο κόμβος root είναι φύλλο. Αν είναι τοτε γράφει στο αρχείο με το writeFile την κωδικοποιήση που έχει ως τότε η στοίβα dq. Αν δεν μπεί στο if τοτε συνεχίζει και κάνει έλεγχο για το αν ο κόμβος που είμαστε εχεί αριστερό και δεξί παιδί με αθτήν την σειρά. Εφόσον μπεί σε κάποιο if τότε προσθέτει στην στοίβα μας 0 ή 1 με την "dq.add()" ανάλογα σε ποιό if θα μπεί και μέσω την αναδρομής ξανακαλεί την "printHuffmanCode" με κόμβο το αριστερό ή δεξί παιδί του προηγούμενου κόμβου. Μόλις φτάσει σε φύλλο το τυπώνει μεσα στο αρχείο και βγάζει με την "dq.removeLast()" το τελευταίο νούμερο που είχε μπεί στην στοίβα ώστε να συνεχίσει σωστά η διάσχιση του δέντρου. (4)

(1) : import java.io.Serializable;

```
public class TreeNode implements Comparable<TreeNode>, Serializable {
    private int frequency;
    private int c;
    private TreeNode left;
    private TreeNode right;

    public int getFrequency() {
        return frequency;
    }

    public void setFrequency(int frequency) {
        this.frequency = frequency;
    }

    public int getC() {
        return c;
    }

    public void setC(int c) {
        this.c = c;
    }
```

```
    public TreeNode getLeft() {
        return left;
    }

    public void setLeft(TreeNode left) {
        this.left = left;
    }

    public TreeNode getRight() {
        return right;
    }

    public void setRight(TreeNode right) {
        this.right = right;
    }

    public TreeNode(int frequency, int c, TreeNode left, TreeNode right)
{
        this.frequency = frequency;
        this.c = c;
        this.left = left;
        this.right = right;
    }

    @Override
    public int compareTo(TreeNode treeNode) {
        if (this.frequency > treeNode.frequency) {
            return 1;
        } else if (this.frequency == treeNode.frequency) {
            return 0;
        }

        return -1;
    }


}
```

(2) :

```
   import java.io.Serializable;
   import java.io.ObjectInputStream;
   import java.io.FileInputStream;
   import java.io.IOException;
   import java.util.ArrayDeque;
   import java.io.PrintWriter;


   public class HuffmanCode implements Serializable {
         public static void main(String[] args) throws IOException,
ClassNotFoundException {

         PrintWriter writeFile = new
PrintWriter("src/main/java/com/ergasia03/maven/java/codes.dat");
         FileInputStream InputStream = new
FileInputStream("src/main/java/com/ergasia03/maven/java/tree.dat");
              ObjectInputStream ObjectInputStream = new
ObjectInputStream(InputStream);

              TreeNode root;
```

```
            root = (TreeNode) ObjectInputStream.readObject();


            printHuffmanCoding(root, writeFile);

            writeFile.close();
          }
```

(3) :

```
     static ArrayDeque<String>  dq = new ArrayDeque<>();
```

(4) :

```
public static void printHuffmanCoding(TreeNode root, PrintWriter
writeFile) {


              if (
                 root.getLeft() == null
                 && root.getRight() == null
              ) {

              writeFile.println((char) root.getC() + ":" + dq );
               return;
              }



              if(root.getLeft() != null) {
                 dq.add("0");
              printHuffmanCoding(root.getLeft(), writeFile);
              dq.removeLast();
           }

            if(root.getRight() != null) {
              dq.add("1");
              printHuffmanCoding(root.getRight(), writeFile);
              dq.removeLast();
          }
```

Τα ονοματά μας :
Παναγιώτης Κωλέτσης
Παναγιώτης Πετεινάρης
Παναγιώτης Χάρος