The code is :

```
import java.util.*;
import java.lang.*;
class Palm implements Cloneable {
    ////////////////////////
    // Data memebers
    ////////////////////////
    private int size;

    ////////////////////////////
    // Member functions
    ////////////////////////////
    // Manager function
    public Palm(int size1) {



       size = size1;




    }
    public Object clone() {



        try
        {
            return super.clone();
        }
        catch (CloneNotSupportedException e)
        {
            // This shouldn't happen, since we are Cloneable
            return null;
        }
```

```
    }

    //////////////////////////
    // Member functions
    //////////////////////////
    // Access functions
    public boolean isLarge() {              return size > 50;              } // A palm is large if
                                                                          // size is larger than 50
    public int getSize() { return size; }
    public void setSize(int size1) { size = size1; }


    // Implementor function
    public void enlarge(int sizeInc) { size += sizeInc; }
    public void shrink(int sizeDec) { size -= sizeDec; }
    public String toString() { // See the Demo class to figure out how to
                                        // implement this function
                                        return ("size of plam is " + size);



    }
}

class Finger implements Cloneable {
    //////////////////////////
    // Data members
    //////////////////////////
    private int len;

    //////////////////////////
    // Member functions
    //////////////////////////
    // Manager function
    public Finger(int len1) {


        len = len1;



    }
```

```
    public Object clone() {



        try
        {
            return super.clone();
        }
        catch (CloneNotSupportedException e)
        {
            // This shouldn't happen, since we are Cloneable
            return null;
        }



    }

    // Access functions
    public boolean isLong() {        return len > 30;        } // A finger is long if
                                                                // its length is longer than 30
    public int getLen() { return len; }
    public void setLen(int len1) { len = len1; }
    public void setFinger(Finger f1) { len =    f1.getLen(); }

    // Implementor function
    public void enlarge(int lenInc) { len += lenInc; }
    public void shrink(int lenDec) { len -= lenDec; }
    public String toString() { // See the example in the Demo class to figure
                                        // out how to implement this function
                                        return ("length of finger is " + len);



    }
}

class Hand implements Cloneable {
    ////////////////////////
    // Data members
    ////////////////////////
    private Palm palm;
    private Vector fingers;
```

```
//////////////////////////
// Member functions
//////////////////////////

// Manager function
public Hand(Palm palm1, Vector fingers1) {


       palm = (Palm)palm1.clone();
       fingers = (Vector)fingers1.clone();


}
public Object clone()
{


     try
     {
        Hand h = (Hand)super.clone();
        h.fingers = (Vector)fingers.clone();
        for (int i=0; i < fingers.size(); i++) {
           h.fingers.setElementAt(
                        ((Finger)fingers.elementAt(i)).clone(), i);
        }

          return super.clone();
     }
     catch (CloneNotSupportedException e)
     {
          // This shouldn't happen, since we are Cloneable
          return null;
     }


}

// Predicate function
public boolean isNormal() { return numOfFingers() == 5; } // A hand is normal if it
                                                          // has 5 fingers
// Access functions
public Palm getPalm() {
```

```
        return palm;


    }
    public Finger getFinger(int i) {


        return ((Finger)fingers.elementAt(i));


    }
    public Vector getFingers() {


        return fingers;


    }



    // Implementor function
    public int numOfFingers() {       return fingers.size();       }
    public                    void                    lostOneFinger(int                    index)
{ fingers.removeElementAt(index);                                    }
    public String toString() {

        StringBuffer s = new StringBuffer();
            for (int i=0; i < numOfFingers(); i++){
               s.append("Finger " + i + ": ");
               s.append(getFinger(i).toString());
               s.append("\n");


        }
        return s.toString();
     }
}

/* Will display

Experiment clone ...
Size of palm is 15
Size of palm is 20
```

```
Size of palm is 20

Display information about Hand ...
Size of palm is 20
Finger 0: Length of finger is 3
Finger 1: Length of finger is 4
Finger 2: Length of finger is 5

*/
public class Demo {
    public static void main (String argv[])
    {
        System.out.println("\nExperiment clone ...");
        Palm palm= new Palm(15);
        System.out.println(palm); // print "Size of palm is 15"
        Palm palm1= palm;
        palm1.setSize(20);
        System.out.println(palm); // print "Size of palm is 20"
        Palm palm2= (Palm)palm.clone();
        palm2.setSize(30);
        System.out.println(palm); // print "Size of palm is 20"


        Vector fingers = new Vector();
        fingers.addElement(new Finger(3));
        fingers.addElement(new Finger(4));
        fingers.addElement(new Finger(5));

        System.out.println("\nDisplay information about Hand ...");
        Hand p = new Hand(palm, fingers);
        System.out.println(p);
    }
}
```

```
Size of palm is 20

Display information about Hand ...
Size of palm is 20
Finger 0: Length of finger is 3
Finger 1: Length of finger is 4
Finger 2: Length of finger is 5

*/
public class Demo {
   public static void main (String argv[])
   {
      System.out.println("\nExperiment clone ...");
      Palm palm= new Palm(15);
      System.out.println(palm); // print "Size of palm is 15"
      Palm palm1= palm;
      palm1.setSize(20);
      System.out.println(palm); // print "Size of palm is 20"
      Palm palm2= (Palm)palm.clone();
      palm2.setSize(30);
      System.out.println(palm); // print "Size of palm is 20"


      Vector fingers = new Vector();
      fingers.addElement(new Finger(3));
      fingers.addElement(new Finger(4));
      fingers.addElement(new Finger(5));

      System.out.println("\nDisplay information about Hand ...");
      Hand p = new Hand(palm, fingers);
      System.out.println(p);
   }
}
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

Interactive

Stdin Inputs

CommandLine Arguments

Execute

Result

CPU Time: 0.27 sec(s), Memory: 35012 kilobyte(s)

compiled and executed in 1.332 sec(s)

```
Experiment clone ...
size of plam is 15
size of plam is 20
size of plam is 20

Display information about Hand ...
Finger 0: length of finger is 3
Finger 1: length of finger is 4
Finger 2: length of finger is 5
```