

8. Übungsaufgaben zur LV Algorithmen & Datenstrukturen

Abgabetermin: Mo, 17.05.04

Aufgabe:

- Implementation eines Programms zum Vergleich von Such-Algorithmen:

```
public class Dialog_Such extends JDialog {
    //...
    int sanz=7;
    String[][] Tab = new String[5][sanz+1];
    String[] TabKopf = new String[]{" ", "SortListe", "SortListeKett", "HashKett",
        "HashCoal", "HashDopp", "AVLBaum", "BBaum"};
    int dLg=1000;
    JTable jT = new JTable(Tab, TabKopf);
    JOptionPane jOptPan = new JOptionPane();

    private void jbInit() throws Exception {
        //...
        for(int i=1; i<=5; i++)
            jT.setValueAt("n="+dLg*i, i-1, 0);
    }

    void OKBtn_actionPerformed(ActionEvent ae) {
        int n=dLg, anzExp=1000, i, j, k; //anzExp dem Prozessor anpassen
        double start, ende;

        //##### Instanzen der SuchStruk-turen erzeugen #####
        SuchStruk sl, slK, hk, hc, hd, ab, bb;
        sl=new SortListe(dLg*5); //max. dLg*5 Elemente
        slK=new SortListeKett();
        hk=new HashKett();
        hc=new HashCoal();
        hd=new HashDopp();
        ab=new AVLBaum();
        bb=new BBaum(50); //Vorl.: "meist wird m=50 gewählt"; hier also auch

        //##### SuchStruk's mit Werten 1,2,...,dLg*5 füllen #####
        for(j=1; j<=dLg*5; j++){ //SuchStruk's mit Werten 1,2,...,dLg*5 füllen
            ElementtypI e=new ElementtypI(j);
            sl.einfuegen(e); slK.einfuegen(e); hk.einfuegen(e); hc.einfuegen(e);
            hd.einfuegen(e); ab.einfuegen(e); bb.einfuegen(e);
        }

        //##### Suchen & Operationen ausführen #####
        for(i=1; i<=5; i++, n=dLg*i) { //von n=dLg bis n=5*dLg
            //jOptPan.showMessageDialog(this, "n="+n);
            for(j=0; j<sanz; j++){
                start=Calendar.getInstance().getTime().getTime();
                for(k=0; k<anzExp; k++){
                    ElementtypI y=null, x=new ElementtypI((int)Math.floor(Math.random()*n+1));
                    switch(j){
                        case 0: y=(ElementtypI) sl.suchen(x.getSchluessel()); break;
                        case 1: y=(ElementtypI) slK.suchen(x.getSchluessel()); break;
                        case 2: y=(ElementtypI) hk.suchen(x.getSchluessel()); break;
                        case 3: y=(ElementtypI) hc.suchen(x.getSchluessel()); break;
                        case 4: y=(ElementtypI) hd.suchen(x.getSchluessel()); break;
                        case 5: y=(ElementtypI) ab.suchen(x.getSchluessel()); break;
                        case 6: y=(ElementtypI) bb.suchen(x.getSchluessel()); break;
                        default: jOptPan.showMessageDialog(this, "Algorithmus nicht vorhanden!");
                    }
                }
            }
        }
    }
}
```

