

```
//element jednostruko spregnute liste//
public class Cvor {

    int info; //podatak u cvoru//
    Cvor next; //pokazivac na sledeci cvor//

}

//metoda koja vraca koliko elemenata ima jslista//
public int broj (Cvor c){
    if (c==null)
        return 0;
    return 1+ broj(c.next); /*metoda poziva samu
sebe, prebrojava po svakom sledecem clanu*/
}

//metoda koja proverava da li se element nalazi u listi//
public boolean proverava (Cvor c, int b){
    if (c==null)
        return false;
    else if (c.info == b) return true;
    else
        return proverava (c.next,b);
}

public Cvor ubaci (int i, Cvor k){
    Cvor p = new Cvor();
    p.info = i;
    p.next = k;
    return p;
}

//metoda koja vraca presek dva skupa//
public Cvor presek (Cvor c1, Cvor c2){
    if (c1==null || c2==null)
        return null; /*ako je bilo koja prazna,presek je
prazan skup*/
    if (!proverava(c2, c1.info))
        return presek (c1.next,c2); /*ako ne pripada
cvoru c2, proverava da li sledeci pripada*/
    else
        return ubaci (c1.info, presek(c1.next,c2));
}

//ako pripada, ubacuje taj element//
}

//metoda koja vraca uniju dva skupa//
public Cvor unija (Cvor c1, Cvor c2){
    if (c1 == null)
        return c2;
    if (c2 == null)
        return c1; /*ako je bilo koja prazna,unija je
drugi skup*/
    if (proverava (c2,c1.info))
        return unija (c1.next,c2); /*ako pripada c2,
proverava se da li sledeci pripada*/
    else
```

```
        return ubaci (c1.info,unija (c1.next,c2)); //ako
ne,vraca se taj element, i proverava se sledeci//
    }

//metoda koja vraca zbir clanova liste//
    public int zbir (Cvor c){
        if (c == null)
            return 0;
        else
            return c.info + zbir (c.next);
    }

//metoda koja vraca broj pozitivnih clanova liste//
    public int brPoz (Cvor c){
        if (c == null)
            return 0;
        else
            if(c.info>0)
                return 1+brPoz(c.next);
            else return brPoz(c.next);
    }

//metoda za binarno pretrazivanje, rekurzivno//
    public int binarnoPretrazi (int [] niz, int element, int
dg, int gg){
        if (niz.length == 0)
            return -1;
        int s;
        s= (dg+gg)/2; //srednji clan niza//
        if (niz [s]==element)
            return s; /*ako je element jednak srednjem
clanu,vrati srednji clan*/
        if (niz [s]>element)
            return binarnoPretrazi (niz,element,dg,s-1);
/*ako je element manji,pretrazice levu stranu,ponovo deliti niz na
pola*/
        if (niz[s]<element)
            return binarnoPretrazi (niz,element,s+1,gg);
//ako je element veci,pretrazice desnu stranu//
    }
}
```