```
class knapsack {

// Globale Gr”áen
private static final int N=5;
private static final int W=20;
private static final int[] wert= {16,15,13,12,10};
private static final int[] gewicht= {13,12,10,8,7};
private static boolean[] rucksack={false,false,false,false,false};
private static boolean[] optimalerRucksack={false,false,false,false,false};


public static void main(String [] args) {

// Aufzeichnung der Ausgangssituation
aufzeichnungRucksack(-1);

// Aufruf der rekursiven Methode
behandle(0);

// Ausgabe der Situation
aufzeichnungOptimalerRucksack();
}

private static void aufzeichnungRucksack (int akt) {
int i,j;
for (i=0; i<N;i++) {
   j=i+1;
    System.out.println("Gegenstand "+j+": "+rucksack[i]);
    }
int hilfGewicht=gesamtGewicht();
int hilfWert=gesamtWert(rucksack);
int hilfOptimalWert=gesamtWert(optimalerRucksack);
int hilfErreichbarerWert=erreichbarerWert(akt);
System.out.println("Gewicht:           "+hilfGewicht);
System.out.println("Wert:              "+hilfWert);
System.out.println("Erreichbarer Wert: "+hilfErreichbarerWert);
System.out.println("Optimalwert:       "+hilfOptimalWert);
System.out.println("---------------------------------------------------");
System.out.println();
}

private static void aufzeichnungOptimalerRucksack () {
int i,j;
System.out.println("Optimaler Rucksack:");
for (i=0; i<N;i++) {
   j=i+1;
    System.out.println("Gegenstand "+j+": "+optimalerRucksack[i]);
    }
int hilfOptimalWert=gesamtWert(optimalerRucksack);
System.out.println("Neues Optimum= "+hilfOptimalWert);
System.out.println("---------------------------------------------------");
```

```
System.out.println();
}


// Berechnung Gewicht des Rucksacks
private static int gesamtGewicht() {
int res=0;
for (int i=0; i<N; i++) {
  if (rucksack[i]) {
    res=res+gewicht[i];
    }
  }
return res;
}

//Berechnung Wert eines Rucksacks (f• r aktuellen und optimalen Rucksack)
private static int gesamtWert(boolean[] rucksack) {
int res=0;
for (int i=0; i<N; i++) {
  if (rucksack[i]) {
    res=res+wert[i];
    }
  }
return res;
}

// Berechnung des noch erreichbaren Werts,
// wenn der Gegenstand akt+1 nicht in den Rucksack aufgenommen wird
private static int erreichbarerWert(int akt) {
int res=0;
for (int i=0; i<=akt; i++) {
  if (rucksack[i]) {
    res=res+wert[i];
    }
  }
for (int i=akt+1; i<N; i++) {
  res=res+wert[i];
  }
return res;
}

// Rekursive Methode
private static void behandle(int i) {

// Gegenstand i+1 in Rucksack aufgenommen
rucksack[i]=true;
int hilfGewicht=gesamtGewicht();
int hilfWert=gesamtWert(rucksack);
int hilfOptimalWert=gesamtWert(optimalerRucksack);

// Debug Info
```

```
aufzeichnungRucksack(i);

// Gewichtsgrenze noch nicht erreicht
if (hilfGewicht<=W) {

  // alle Gegenst„nde betrachtet
  if (i==N-1) {

    // Test auf Optimalit„t
    if (hilfWert > hilfOptimalWert) {
      for(int j=0; j<N; j++) {
        optimalerRucksack[j]=rucksack[j];
        }

      // Debug Info
      aufzeichnungOptimalerRucksack();

      }
    }
  else {
    behandle(i+1);
    }
  }

// Gegenstand i+1 nicht in Rucksack aufgenommen
rucksack[i]=false;

int hilfErreichbarerWert=erreichbarerWert(i);
hilfWert=gesamtWert(rucksack);
hilfOptimalWert=gesamtWert(optimalerRucksack);

// Debug Info
aufzeichnungRucksack(i);

// Optimalwert noch zu erreichen
if (hilfErreichbarerWert >= hilfOptimalWert) {

  // alle Gegenst„nde betrachtet
  if (i==N-1) {

    // Test auf Optimalit„t
    if (hilfWert > hilfOptimalWert) {
      for(int j=0; j<N; j++) {
        optimalerRucksack[j]=rucksack[j];
        }

      // Debug Info
      aufzeichnungOptimalerRucksack();

      }
    }
```

```
  else {
       behandle(i+1);
     }

  }
}

}
```