

# Notierhilfe für prioritätsregelbasierte Maschinenbelegung

Jobnummer $j$	Bearbeitungszeit $p_j$	gewünschter Fertigstellungstermin $d_j$
1	4	11
2	3	9
3	2	7
4	3	20
5	5	10
6	6	5

## SPT-Regel

Jobnummer $j$	Bearbeitungszeit $p_j$	gewünschter Fertigstellungstermin $d_j$	tatsächlicher Fertigstellungstermin $c_j$	Verspätung $t_j$
3	2	7	2	-
2	3	9	5	-
4	3	20	8	-
1	4	11	12	1
5	5	10	17	7
6	6	5	23	18

TT(SPT) = 3 -> 2 -> 4 -> 1 -> 5 -> 6

## EDD-Regel

Jobnummer $j$	Bearbeitungszeit $p_j$	gewünschter Fertigstellungstermin $d_j$	tatsächlicher Fertigstellungstermin $c_j$	Verspätung $t_j$
6	6	5	6	1
3	2	7	8	1
2	3	9	11	2
5	5	10	16	6
1	4	11	20	9
4	3	20	23	3

TT(EDD) = 6 -> 3 -> 2 -> 5 -> 1 -> 4

## Aufgabe 1)

```
1 package com.MirkoCordes.OOP;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         // write your code here
7
8         Heuristik spt_regel = new Heuristik();
9         spt_regel.DoSomething( aufBearbeitungSetzen: false);
10    }
11 }
```

```
1 package com.MirkoCordes.OOP;
2
3 public class Heuristik {
4     int[] j = {1, 2, 3, 4, 5, 6}; // j = Jobnumber
5     int[] pj = {4, 3, 2, 3, 5, 6}; // pj = Bearbeitungszeit
6     int[] dj = {11, 9, 7, 20, 10, 5}; // dj = Fertigstellungstermin
7     int[] tt = {7,7,7,7,7,7};
8     int[] cj = {7,7,7,7,7,7};
9
10    Heuristik() {
11
12    }
13
14    public void setJ(int[] j) {
15        this.j = j;
16    }
17
18    public void setPj(int[] pj) {
19        this.pj = pj;
20    }
21
22    public void setDj(int[] dj) {
23        this.dj = dj;
24    }
```

```

26 void DoSomething(boolea aufBearbeitungSetzen){
27     int tempJ, tempPj, tempDj;
28
29     if(aufBearbeitungSetzen){
30         //TODO: Priorisiere Pj
31
32         for(int i=0; i<6; i++){
33             for(int b=0; b<6; b++){
34                 if(pj[i] <= pj[b]){
35                     //sortiere nach pj
36                     tempPj = pj[i];
37                     pj[i] = pj[b];
38                     pj[b] = tempPj;
39
40                     tempJ = j[i];
41                     j[i] = j[b];
42                     j[b] = tempJ;
43
44                     tempDj = dj[i];
45                     dj[i] = dj[b];
46                     dj[b] = tempDj;
47                 }
48             }
49         }
50
51     } else {
52         //TODO: Priorisiere Dj
53         for(int i=0; i<6; i++){
54             for(int b=0; b<6; b++){
55                 if(dj[i] <= dj[b]){
56                     //sortiere nach pj
57                     tempPj = pj[i];
58                     pj[i] = pj[b];
59                     pj[b] = tempPj;
60
61                     tempJ = j[i];
62                     j[i] = j[b];
63                     j[b] = tempJ;

```

```

65         tempDj = dj[i];
66         dj[i] = dj[b];
67         dj[b] = tempDj;
68     }
69 }
70 }
71 }
72
73     cjBerechnen();
74     ttBerechnen();
75     gebeSomethingAus();
76 }
77
78 void cjBerechnen(){
79     for(int t=0; t<6; t++){
80         if(t!=0){
81             cj[t]= pj[t] + cj[t-1];
82         } else {
83             cj[t]= pj[t];
84         }
85     }
86 }
87 }
88
89 void ttBerechnen(){
90     for(int t=0; t<6; t++){
91         tt[t] = cj[t] - dj[t];
92         if(tt[t] < 0){
93             tt[t] = 0;
94         }
95     }
96 }

```

```

98 void gebeSomethingAus(){
99     System.out.println("Jobnummer, Bearbeitungszeit, Fertigstellungstermin, tatsächliche Fertigstellung, Verspaetung");
100
101     for(int i=0; i<6; i++){
102         System.out.println(j[i] + ", " + pj[i] + ", " + dj[i] + ", " + cj[i] + ", " + tt[i]);
103     }
104 }
105
106 }
107 }

```