

## Modell für Scheduling

# Benötigte Daten

## Beschreibung

Die Daten sind in 2 Teile aufgeteilt:

- System-Info, beinhaltet Informationen über die Umgebung in der produziert wird
- Orders, beinhaltet Kundenaufträge/Bestellungen, die in dem beschriebenen System umgesetzt werden sollen

## System-Info

Beinhaltet Informationen über die Produktions-Umgebung

**Tasks** Alle Arbeitsschritte / Aktionen, die in der Produktions-Umgebung durchgeführt werden können. (Wichtig: Tasks die z.Bsp. bei Task A als „follow up task“ angegeben werden sollen KEINEN Verweis auf Task A in den „preceding tasks“ haben). Die zeitliche Dauer jedes Tasks ist bei den Workstations( section ) abgebildet, um die Umsetzung einer Aufgabe mit verschiedenen Maschinen/Werkzeugen zu berücksichtigen.

- id – eindeutige ID des Tasks (kann fortlaufende Nummer sein, nur für interne Identifizierung)
- name – Name des Tasks (für einfacher lesbare Ergebnisse)
- resources – (liste) benötigt Ressourcen, um den Task durchzuführen (id + anzahl)
- products – (liste) alle Ressourcen (ID + Anzahl (Losgröße)) die durch den Task entstehen
- preceding\_tasks – (liste) Tasks, die gemeinsam mit diesem Task durchgeführt werden müssen (vorangestellte Tasks, müssen daher nicht extra als Arbeitsschritt übermittelt werden)
- follow\_up\_tasks – (liste) Tasks, die nach dem Task durchgeführt werden müssen (wie preceding tasks)
- independent – boolean, um festzustellen, ob der Task im Schedule verwendet werden soll (e.g. false für Tasks die immer nur als vorangestellter Task vorkommen und allein keinen Sinn ergeben)

- `prepare_time` – Zeit die zum Einrichten der Maschine/Arbeitsstation für diesen Task benötigt wird
- `unprepare_time` – Zeit, die nach der Verwendung der Maschine/Arbeitsstation für diesen Task benötigt wird

**Recipes** Die Rezepte für die verschiedenen Ressourcen, die hergestellt werden sollen (Produkte oder auch Zwischenprodukte)

- `id` – eindeutige ID des Rezepts (kann auch fortlaufende Nummer sein, nur für interne Identifizierung)
- `name` – Name des Rezepts (für einfacher lesbare Ergebnisse)
- `tasks` – (liste) IDs der Tasks, die durchgeführt werden müssen, um das Rezept umzusetzen (Ergebnis und Ergebnismenge ergeben sich aus dem letzten Task)

**Resources** Alle Ressourcen, die in der Produktions-Umgebung verwendet oder hergestellt werden können

- `id` – eindeutige ID der Ressource (kann auch fortlaufende Nummer sein, nur für interne Identifizierung)
- `name` – Name der Ressource (für einfacher lesbare Ergebnisse)
- `stock` – Anzahl der Ressource im Lagerbestand
- `price` – Preis pro Einheit, um die Ressource einzukaufen (0 wenn die Ressource nicht gekauft werden kann)
- `delivery_time` - Erwartete Wartezeit, falls die Ressource eingekauft werden muss
- `renewable` – boolean, um festzustellen, ob die Ressource verbraucht wird oder nicht (könnte z.Bsp. auch Ressource Mitarbeiter sein, der wieder verfügbar ist, wenn ein Task beendet ist)
- `recipes` – (liste) IDs der Rezepte, durch die diese Ressource hergestellt werden kann (falls vorhanden, sonst leere Liste)

**Workstations** Alle Maschinen / Arbeitsstationen, die in der Produktions-Umgebung vorhanden sind

- id – eindeutige ID der Maschine/Arbeitsstation (kann auch fortlaufende Nummer sein, nur für interne Identifizierung)
- name – Name der Maschine/Arbeitsstation (für einfacher lesbare Ergebnisse)
- basic\_resources – (liste) beinhaltet die IDs der benötigten Ressourcen + die Anzahl für jede Ressource, die (unabhängig der bevorstehenden Tasks) benötigt werden um die Maschine/Arbeitsstation zu betreiben (z.Bsp. Mitarbeiter)
- tasks – (liste) IDs aller Tasks, die auf dieser Maschine/Arbeitsstation durchgeführt werden können + die Dauer des Tasks auf dieser Maschine/Arbeitsstation

## Orders

Repräsentiert Kundenaufträge/Bestellungen, die mithilfe der Produktions-Umgebung umgesetzt werden soll.

- id – eindeutige ID der Bestellung (kann auch fortlaufende Nummer sein, nur für interne Identifizierung)
- arrival\_time – Zeitpunkt, zu dem die Bestellung im System eingegangen ist
- delivery\_time – Zeitpunkt, zu dem die Bestellung fertig sein soll
- latest\_acceptable\_time – Spätester Zeitpunkt, zu dem die Bestellung fertig sein soll, falls der gewünschte Lieferzeitpunkt nicht eingehalten kann
- resources – (liste) IDs + Anzahl der bestellten Ressourcen, die hergestellt werden sollen, + der Preis, den der Kunde für diese Ressource bezahlen wird
- penalty – falls die Bestellung nicht durchgeführt wird
- tardiness\_fee – falls die Bestellung nach dem gewünschten Termin, aber vor dem letzten möglichen Termin durchgeführt wird
- divisible – boolean, gibt an ob auch nur Teile der Bestellung bearbeitet werden können, oder ob alle bestellten Ressourcen geliefert werden müssen (oder gar keine)

- `customer_id` – ID des Kunden, von dem der Auftrag stammt
- `optional` - boolean, gibt an ob eine Bestellung zwingend im Schedule vorhanden sein muss oder nicht (default: true)

## Jobs

Repräsentiert die spezifische Ausführung eines Tasks für eine Bestellung

- `id` - eindeutige ID des Jobs (kann auch fortlaufende Nummer sein, nur für interne Identifizierung)
- `order_id` - ID der Bestellung, für die der Job ausgeführt werden muss
- `task_id` - ID des Tasks, der ausgeführt werden soll
- `to_id` - Spezifische Zuteilungs-ID zwischen Tasks und Bestellungen, um den Job im Schedule genau zuordnen zu können

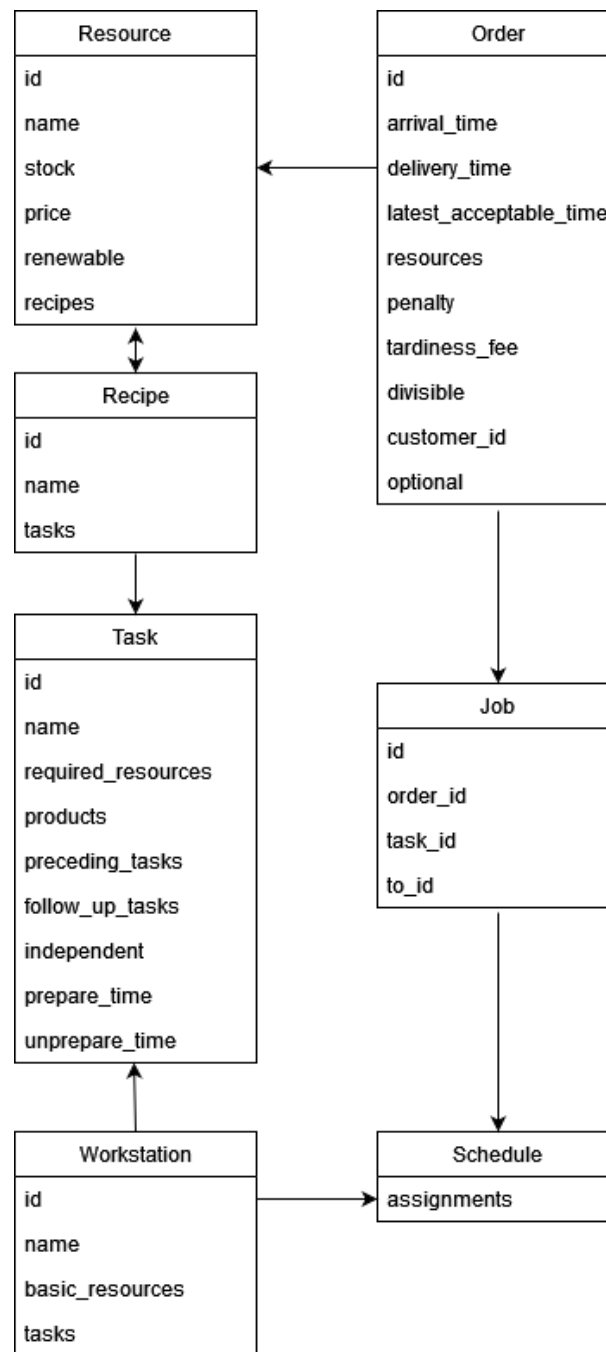


Figure 1: Dependencies

# Input / Output

Genaue Struktur der Datenübertragung mit Beispieldaten unter "JSON-Beispiel" (section )

Input:

Orders:

*OrderA*

*Task<sub>1</sub> – Task<sub>n</sub>*

*price<sub>1</sub> – price<sub>n</sub>*

*penalty*

*tardiness\_fee*

*delivery\_date*

*latest\_date*

*optional*

Mapping von Tasks zu Jobs Bsp.:

(Zuteilung zu exakten Job, Bsp. Task: Material schneiden -> Job: Material schneiden für Bestellung A)

*TaskA.1*

*TaskID*

*JobID*

Intermediate:

<Workstation ID, Task (Operation) ID, Start Time Slot>  $j_1 - (w, t, s)$

$j_2 - (w, t, s)$

$j_2 - (w, t, s)$

.

.

.

$j_n - (w, t, s)$

Output:

$$\begin{bmatrix} (j_{11}, s_{11}) & (j_{12}, s_{12}) & (j_{13}, s_{13}) & \dots & (j_{1n}, s_{1n}) \\ (j_{21}, s_{21}) & (j_{22}, s_{22}) & (j_{23}, s_{23}) & \dots & (j_{2n}, s_{2n}) \\ \dots & \dots & \dots & \dots & \dots \\ (j_{m1}, s_{m1}) & (j_{m2}, s_{m2}) & (j_{m3}, s_{m3}) & \dots & (j_{mn}, s_{mn}) \end{bmatrix}$$

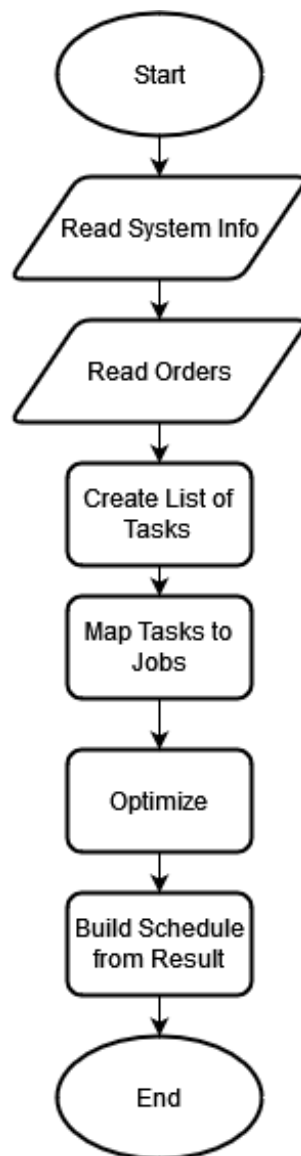


Figure 2: Ablauf

## Modell

### Variablen Definition

OLD:

$O$  - Set of all Orders

$o$  - Specific Order,  $o \in O$

$W$  - Set of all Workstations



$T$  - Set of all Tasks  
 $T_o$  - Set of all Tasks needed for Order  $o$ ,  $T_o \subset T$   
 $x$  - Specific Task,  $x \in T$   
 $T_{xp}$  - Set of all Tasks preceding Task  $x$ ,  $T_{xp} \subset T$   
 $T_{xf}$  - Set of all Tasks following Task  $x$ ,  $T_{xf} \subset T$   
 $J$  - Set of all Jobs  
 $J_w$  - Set of all Jobs assigned to Workstation  $w$ ,  $w \in W$   
 $j_x$  - Job linked to Task  $j$ ,  $x \in T$ ,  $j \in J$   
 $x_j$  - Task linked to Job  $j$ ,  $x \in T$ ,  $j \in J$   
 $W_j$  - Set of Workstations eligible for Job  $j$ ,  $W_j \subset W$   
 $R$  - Set of all Resources  
 $w$  - Specific Workstation  $w \in W$   
 $R_j$  - Set of Resources needed for Job  $j$ ,  $R_j \subset R$   
 $r$  - Specific Resource,  $r \in R$   
 $d_{xw}$  - Duration of Task  $x$  on Workstation  $w$   
 $j_w$  - Selected Workstation for a Job  $j$ ,  $w \in W_j$   
 $j_s$  - Start time slot of Job  $j$   
 $j_{dt}$  - Deliver time slot of Job  $j$   
 $j_{lt}$  - Latest allowed time slot of Job  $j$   
 $i_{rt}$  - Amount of Resource  $r$  in the Inventory at time slot  $t$   
 $l_j$  - Binary Variable, 1 if Job  $j$  is late (Tardiness Fee applies)  
 $l_o$  - Binary Variable, 1 if Order  $o$  is late (Tardiness Fee applies)  
 $u_o$  - Binary Variable, 1 if Order  $o$  can only be fulfilled partially  
 $p_o$  - Price the customer pays for Order  $o$   
 $f_o$  - Price reduction for tardy orders  
 $O_c$  - Set of all Orders which could be completed in the derived schedule,  
 $O_c \subset O$

#### NEW SUGGESTION:

$O$  - Set of all Orders  
 $P$  - Set of all Tasks  
 $W$  - Set of all Workstations  
 $R$  - Set of all Resources  
 $X$  - Set of all Jobs  
 $o$  - Specific Order,  $o \in O$   
 $p$  - Specific Task,  $p \in P$   
 $w$  - Specific Workstation  $w \in W$   
 $r$  - Specific Resource,  $r \in R$   
 $x$  - Specific Job,  $x \in X$   
 $P_o$  - Set of all Tasks needed to complete Order  $o$ ,  $P_o \subset P$

$P_{pre}$  - Set of all Tasks preceding Task  $p$ ,  $P_{pre} \subset P$   
 $P_{pf}$  - Set of all Tasks following Task  $p$ ,  $P_{pf} \subset P$   
 $X_w$  - Set of all Jobs assigned to Workstation  $w$ ,  $w \in W$ ,  $X_w \subset X$   
 $W_x$  - Set of all Workstations eligible for Job  $x$ ,  $x \in X$ ,  $W_x \subset W$   
 $R_x$  - Set of all Resources required for Job  $x$ ,  $x \in X$ ,  $R_x \subset R$   
 $d_{pw}$  - Duration of Task  $p$  on Workstation  $w$ ,  $p \in P$ ,  $w \in W$ ,  $d_{pw} \geq 0$   
 $x_w$  - Selected Workstation  $w$  for Job  $x$ ,  $x \in X$ ,  $w \in W$ ,  $x_w \in W$   
 $x_s$  - Start time slot for Job  $x$ ,  $x \in X$ ,  $x_s \geq 0$   
 $x_{dt}$  - Delivery time slot for Job  $x$ ,  $x \in X$ ,  $x_{dt} \geq 0$   
 $x_{lt}$  - Latest acceptable time slot for Job  $x$ ,  $x \in X$ ,  $x_{lt} \geq 0$ ,  $x_{lt} \geq x_{dt}$   
 $a_{r,t}$  - Amount of Resource  $r$  in stock at time slot  $t$ ,  $r \in R$ ,  $t \geq 0$ ,  $a_{r,t} \geq 0$   
 $l_x$  - Binary Variable, 1 if Job  $x$  is late (Tardiness Fee applies),  $x \in X$   
 $l_o$  - Binary Variable, 1 if Order  $o$  is late (Tardiness Fee applies),  $o \in O$   
 $u_o$  - Binary Variable, 1 if Order  $o$  can only be fulfilled partially,  $o \in O$ ,  
 $z_o$  - Price the customer pay for Order  $o$ ,  $o \in O$ ,  $z_o \geq 0$   
 $f_o$  - Price reduction for tardy orders,  $o \in O$ ,  $f_o \geq 0$   
 $o_{opt}$  - Binary Variable, 1 if Order  $o$  is optional,  $o \in O$   
 $o_{lt}$  - Latest acceptable time for Order  $o$ ,  $o \in O$ ,  $o_{lt} \geq 0$   
 $b_o$  - Penalty, if Order  $o$  is not scheduled to meet  $o_{lt}$ ,  $o \in O$ ,  $b_o \geq 0$   
 $O_c$  - Set of all Orders which could be completed in the schedule,  $O_c \subset O$

## Objective Function + Nebenbedingungen

### Objective Functions

Equation 1 Minimize tardy jobs

Equation 2 Maximize earning

Equation 3 Minimize deviation from the expected delivery date

Equation 4 Maximize Set Size of scheduled jobs

Equation 5 Maximize Set Size of scheduled jobs with least tardy jobs

Equation 6 Maximize Set Size of scheduled jobs with least deviations

$$\begin{aligned}
 j &:= J_i \\
 \text{minimize } &\sum_{i=0}^J l_j
 \end{aligned} \tag{1}$$

$$\begin{aligned}
 o &:= O_{c,i} \\
 \text{maximize } &\sum_{i=0}^{O_c} p_o - (l_o * f_o)
 \end{aligned} \tag{2}$$

$$\begin{aligned}
j &:= J_i \\
y &:= x_j \\
\text{minimize} \sum_{i=0}^J |(j_s + d_{y,w}) - j_{dt}| &
\end{aligned} \tag{3}$$

$$\text{maximize} |O_c| \tag{4}$$

$$\begin{aligned}
j &:= J_i \\
\text{maximize} |O_c| - \sum_{i=0}^J l_j &
\end{aligned} \tag{5}$$

$$\begin{aligned}
j &:= J_i \\
y &:= x_j \\
z & \dots \text{threshold for acceptable deviation} \\
x &:= \sum_{i=0}^J (|(j_s + d_{y,w}) - j_{dt}| > z) \\
\text{maximize} |O_c| - x &
\end{aligned} \tag{6}$$

### Nebenbedingungen

Equation 7 to make sure jobs finish before last possible time slot

Equation 8 to make sure jobs have sufficient resources to start

Equation 9 checks if tardiness fee applies

Equation 10 jobs can't start before preceding tasks finish

Equation 11 follow up jobs can't start before job finished

Equation 12 only one active job at one time slot for each Workstation

Equation 13 starting times need to be after timeslot 0

$$j_s \leq j_{lt} - d_{j,w} \tag{7}$$

$$\sum_{m=0}^{R_j} (i_{r_m,t} - r_m \leq 0) < 1 \tag{8}$$

$$l_j := j_{dt} \leq j_s \leq j_{lt} \tag{9}$$

$$\begin{aligned}
&y \in T_{j_{xp}} \\
\sum_{m=1}^y ((j_{y_m,s} + d_{y_m,w}) - (j_{y_{m-1}} + d_{y_{m-1},w}) < 0) < 1 &
\end{aligned} \tag{10}$$

$$y \in T_{j_x f}$$

$$\sum_{m=1}^y ((j_{y_{m-1},s} + d_{y_{m-1},w}) < j_{y_m,s}) < 1 \quad (11)$$

$$\forall w \in W$$

$$i = J_{wm}$$

$$j = J_{wn}$$

$$x = x_i \quad (12)$$

$$x' = x_j$$

$$\sum_{m=0}^{J_w} (\sum_{n=0}^{J_w-1} ((1 - (j_s + d_{x',w} < i_s)) + (1 - (j_s > i_s + d_{x,w})))) < 1$$

$$j = J_m$$

$$\sum_{m=0}^J (j_s < 0) < 1 \quad (13)$$

## JSON-Beispiel

```

1 {
2   "system-info":
3     {
4       "tasks":
5         [
6           {
7             "id": 0,
8             "name": "Task 1",
9             "resources":
10              [
11                {
12                  "id": 0,
13                  "amount": 10
14                }
15              ],
16             "products":
17              [
18                {
19                  "resource_id":
20                    1,

```

```

20         "amount": 1
21     }
22 ],
23 "preceding_tasks":
24 [],
25 "follow_up_tasks":
26 [
27     1
28 ],
29 "independent": true,
30 "prepare_time": 5,
31 "unprepare_time": 5
32 },
33 {
34     "id": 1,
35     "name": "Task 2",
36     "resources":
37     [
38         {
39             "id": 1,
40             "amount": 1
41         }
42     ],
43     "products":
44     [
45         {
46             "resource_id":
47                 2,
48             "amount": 1
49         }
50     ],
51     "preceding_tasks":
52     [],
53     "follow_up_tasks":
54     [],
55     "independent": false,
56     "prepare_time": 0,
57     "unprepare_time": 5
58 },
59 {

```

```

59         "id": 2,
60         "name": "Task 3",
61         "resources":
62         [
63             {
64                 "id": 3,
65                 "amount": 20
66             },
67             {
68                 "id": 4,
69                 "amount": 10
70             }
71         ],
72         "products":
73         [
74             {
75                 "resource_id":
76                     5,
77                 "amount": 2
78             },
79             {
80                 "resource_id":
81                     6,
82                 "amount": 1
83             }
84         ],
85         "preceding_tasks":
86         [
87             3, 4
88         ],
89         "follow_up_tasks":
90         [],
91         "independent": true,
92         "prepare_time": 10,
93         "unprepare_time": 5
94     },
95     {
96         "id": 3,
97         "name": "Task 4",
98         "resources":

```

```

97         [
98             {
99                 "id": 7,
100                 "amount": 5
101             }
102         ],
103         "products":
104         [
105             {
106                 "resource_id":
107                     3,
108                 "amount": 20
109             }
110         ],
111         "preceding_tasks":
112         [],
113         "follow_up_tasks":
114         [],
115         "independent": false,
116         "prepare_time": 5,
117         "unprepare_time": 5
118     },
119     {
120         "id": 4,
121         "name": "Task 5",
122         "resources":
123         [
124             {
125                 "id": 8,
126                 "amount": 1
127             }
128         ],
129         "products":
130         [
131             {
132                 "resource_id":
133                     4,
134                 "amount": 5
135             }
136         ],

```

```

135         "preceding_tasks":
136         [],
137         "follow_up_tasks":
138         [],
139         "independent": true,
140         "prepare_time": 1,
141         "unprepare_time": 5
142     },
143     {
144         "id": 5,
145         "name": "Task 6",
146         "resources":
147         [
148             {
149                 "id": 9,
150                 "amount": 20
151             },
152             {
153                 "id": 10,
154                 "amount": 1
155             }
156         ],
157         "products":
158         [
159             {
160                 "resource_id":
161                     8,
162                 "amount": 2
163             }
164         ],
165         "independent": true,
166         "preceding_tasks":
167         [],
168         "follow_up_tasks":
169         [],
170         "prepare_time": 5,
171         "unprepare_time": 5
172     }
173 ],
    "recipes":

```



```

174     [
175         {
176             "id": 0,
177             "name": "Recipe 1",
178             "tasks":
179             [
180                 5
181             ]
182         },
183         {
184             "id": 1,
185             "name": "Recipe 2",
186             "tasks":
187             [
188                 2
189             ]
190         },
191         {
192             "id": 2,
193             "name": "Recipe 3",
194             "tasks":
195             [
196                 0
197             ]
198         },
199         {
200             "id": 3,
201             "name": "Recipe 4",
202             "tasks":
203             [
204                 4
205             ]
206         }
207     ],
208     "resources":
209     [
210         {
211             "id": 0,
212             "name": "Resource 1",
213             "stock": 50,

```

```

214         "price": 20,
215         "delivery_delay": 0,
216         "renewable": false,
217         "recipes":
218             []
219     },
220     {
221         "id": 1,
222         "name": "Resource 2",
223         "stock": 0,
224         "price": 300,
225         "delivery_delay": 0,
226         "renewable": false,
227         "recipes":
228             []
229     },
230     {
231         "id": 2,
232         "name": "Resource 3",
233         "stock": 10,
234         "price": 1000,
235         "delivery_delay": 0,
236         "renewable": false,
237         "recipes":
238             [
239                 2
240             ]
241     },
242     {
243         "id": 3,
244         "name": "Resource 4",
245         "stock": 100,
246         "price": 10,
247         "delivery_delay": 0,
248         "renewable": false,
249         "recipes":
250             []
251     },
252     {
253         "id": 4,

```

```

254         "name": "Resource 5",
255         "stock": 10,
256         "price": 100,
257         "delivery_delay": 0,
258         "renewable": false,
259         "recipes":
260         [
261             3
262         ]
263     },
264     {
265         "id": 5,
266         "name": "Resource 6",
267         "stock": 4,
268         "price": 50,
269         "delivery_delay": 0,
270         "renewable": false,
271         "recipes":
272         [
273             1
274         ]
275     },
276     {
277         "id": 6,
278         "name": "Resource 7",
279         "stock": 3,
280         "price": 70,
281         "delivery_delay": 0,
282         "renewable": false,
283         "recipes":
284         [
285             1
286         ]
287     },
288     {
289         "id": 7,
290         "name": "Resource 8",
291         "stock": 15,
292         "price": 15,
293         "delivery_delay": 0,

```

```

294         "renewable": false,
295         "recipes":
296             []
297     },
298     {
299         "id": 8,
300         "name": "Resource 9",
301         "stock": 6,
302         "price": 700,
303         "delivery_delay": 0,
304         "renewable": false,
305         "recipes":
306             [
307                 0
308             ]
309     },
310     {
311         "id": 9,
312         "name": "Resource 10",
313         "stock": 40,
314         "price": 200,
315         "delivery_delay": 0,
316         "renewable": false,
317         "recipes":
318             []
319     },
320     {
321         "id": 10,
322         "name": "Resource 11 (Employee
323             )",
324         "stock": 10,
325         "price": 0,
326         "delivery_delay": 0,
327         "renewable": true,
328         "recipes":
329             []
330     },
331     {
332         "id": 11,
333         "name": "Resource 12",

```

```

333         "stock": 1000,
334         "price": 5,
335         "delivery_delay": 0,
336         "renewable": false,
337         "recipes":
338             []
339     },
340 ],
341 "workstations":
342 [
343     {
344         "id": 0,
345         "name": "Workstation 1",
346         "basic_resources":
347             [
348                 {
349                     "id": 10,
350                     "amount": 1
351                 }
352             ],
353         "tasks":
354             [
355                 {
356                     "task_id": 0,
357                     "duration": 10
358                 },
359                 {
360                     "task_id": 2,
361                     "duration": 5
362                 }
363             ]
364     },
365     {
366         "id": 1,
367         "name": "Workstation 2",
368         "basic_resources":
369             [
370                 {
371                     "id": 10,
372                     "amount": 1

```

```

373         },
374         {
375             "id": 11,
376             "amount": 2
377         }
378     ],
379     "tasks":
380     [
381         {
382             "task_id": 1,
383             "duration": 20
384         },
385         {
386             "task_id": 2,
387             "duration": 7
388         }
389     ]
390 },
391 {
392     "id": 2,
393     "name": "Workstation 3",
394     "basic_resources":
395     [],
396     "tasks":
397     [
398         {
399             "task_id": 3,
400             "duration": 1
401         },
402         {
403             "task_id": 4,
404             "duration": 2
405         },
406         {
407             "task_id": 5,
408             "duration": 15
409         }
410     ]
411 },
412 {

```

```

413         "id": 3,
414         "name": "Workstation 4",
415         "basic_resources":
416         [
417             {
418                 "id": 10,
419                 "amount": 1
420             }
421         ],
422         "tasks":
423         [
424             {
425                 "task_id": 4,
426                 "duration": 1
427             },
428             {
429                 "task_id": 5,
430                 "duration": 10
431             }
432         ]
433     }
434 ]
435 },
436 "orders":
437 [
438     {
439         "id": 0,
440         "arrival_time": "2022-02-07 00:00:00",
441         "delivery_time": "2022-02-20 14:30:00"
442         ,
443         "latest_acceptable_time": "2022-02-22
444             00:00:00",
445         "resources":
446         [
447             {
448                 "id": 2,
449                 "amount": 5,
450                 "price": 700

```

```

451         "id": 4,
452         "amount": 10,
453         "price": 300
454     },
455     ],
456     "penalty": 300,
457     "tardiness_fee": 100,
458     "divisible": false,
459     "customer_id": 1,
460     "optional": true
461 },
462 {
463     "id": 1,
464     "arrival_time": "2022-02-17 00:00:00",
465     "delivery_time": "2022-02-21 15:30:00"
466     ,
467     "latest_acceptable_time": "2022-02-24
468         00:00:00",
469     "resources":
470     [
471         {
472             "id": 4,
473             "amount": 20,
474             "price": 600
475         },
476         {
477             "id": 5,
478             "amount": 10,
479             "price": 2000
480         }
481     ],
482     "penalty": 500,
483     "tardiness_fee": 100,
484     "divisible": false,
485     "customer_id": 1,
486     "optional": true
487 },
488 {
489     "id": 2,
490     "arrival_time": "2022-02-10 00:00:00",

```



```

489         "delivery_time": "2022-02-19 14:00:00"
490         ,
491         "latest_acceptable_time": "2022-02-26
492         00:00:00",
493         "resources":
494         [
495             {
496                 "id": 2,
497                 "amount": 20,
498                 "price": 2700
499             },
500             {
501                 "id": 5,
502                 "amount": 10,
503                 "price": 2000
504             }
505         ],
506         "penalty": 1000,
507         "tardiness_fee": 100,
508         "divisible": false,
509         "customer_id": 1,
510         "optional": true
511     },
512     {
513         "id": 3,
514         "arrival_time": "2022-02-08 00:00:00",
515         "delivery_time": "2022-02-23 17:00:00"
516         ,
517         "latest_acceptable_time": "2022-02-26
518         00:00:00",
519         "resources":
520         [
521             {
522                 "id": 5,
523                 "amount": 17,
524                 "price": 3500
525             },
526             {
527                 "id": 6,
528                 "amount": 10,

```

```

525         "price": 1200
526     }
527 ],
528 "penalty": 700,
529 "tardiness_fee": 200,
530 "divisible": false,
531 "customer_id": 1,
532 "optional": true
533 },
534 {
535     "id": 4,
536     "arrival_time": "2022-02-15 00:00:00",
537     "delivery_time": "2022-02-23 12:00:00"
538     ,
539     "latest_acceptable_time": "2022-02-26
540         00:00:00",
541     "resources":
542     [
543         {
544             "id": 4,
545             "amount": 34,
546             "price": 800
547         },
548         {
549             "id": 8,
550             "amount": 10,
551             "price": 3000
552         }
553     ],
554     "penalty": 200,
555     "tardiness_fee": 100,
556     "divisible": false,
557     "customer_id": 1,
558     "optional": true
559 }

```