

# Paket-Zuordnung im Lager



Die CodeSolutions Neumarkt GmbH wurde beauftragt einen **Software-Prototypen** für die **Zuordnung** von **Paketen** zu **Fächern** in einem Lagersystem zu entwickeln.

Im Rahmen eines **Kundenworkshops** wurde in Zusammenarbeit mit dem Kunden eine **Technische Beschreibung** verfasst, welche die Grundlage für das Projektvorhaben darstellt.

Außerdem wurde während des Kundenworkshops ein rudimentäres **Funktionsmuster** entwickelt, welches die **Funktionsweise** und die **Interaktion** mit der Software grundsätzlich demonstriert.

# Ihr Projektleiter überreicht Ihnen die Technische Beschreibung...

Das Projekt besteht aus  
drei Arbeitspaketen.

Sie sollten bis zum  
**Dienstag, 28.10.2025**  
mit der Bearbeitung des  
Arbeitspakets 1 fertig  
sein.

Beginnen Sie also mit  
**Arbeitspaket 1 (AP1).**



# Handlungsauftrag AP1: Informieren

Lesen Sie die folgenden Kapitel der **Technischen Beschreibung** durch:

- *1.1 Aufgabenstellung bis 1.5.1 Arbeitsinhalte von Arbeitspaket 1*

Machen Sie sich Gedanken zu folgenden **Fragestellungen**:

- Welche **Arbeitsinhalte** müssen für AP1 durchgeführt werden?
- Wie könnten Sie die **Zuordnungs-Regeln** im Programm umsetzen?



**Eigenarbeit, 12 Minuten**



# Welche **Arbeitsinhalte** müssen für AP1 durchgeführt werden?

1. Entwurf des Programmablaufplans
2. Entwicklung der Software
3. Testen der Software
4. Dokumentation des Programmablaufs

Seite 2

## 1.5 ARBEITSINHALTE

### 1.5.1 AP1: Entwurf & Entwicklung einer Software zur einmaligen Zuordnung von Paketen

Die Bearbeitung von AP1 umfasst folgende Inhalte:

1. Entwurf des Programmablaufplans
2. Entwicklung der Software
3. Testen der Software
4. Dokumentation des Programmablaufs



# Wie könnten Sie die Zuordnungs-Regeln im Programm umsetzen?

Nr.	Regel	Fach
R1	Die Paketnummer ist restlos durch 4 <b>oder</b> 5 teilbar.	1
R2	Die Paketnummer ist zusätzlich restlos durch 100 teilbar.	2
R3	Die Paketnummer ist zusätzlich restlos durch 400 teilbar.	3
R4	R1, R2 und R3 treffen nicht zu.	4



# Logische Operatoren and und or

## - Wahrheitstabelle

Bedingung A		Bedingung B	UND (and)	ODER (or)
0	or	0	0	0
0	or	1	0	1
1	or	0	0	1
1	or	1	1	1

0 = False    1 = True



# Logische Operatoren and und or

## - Wahrheitstabelle

Bedingung A		Bedingung B	UND (and)	ODER (or)
0	and	0	0	0
0	and	1	0	1
1	and	0	0	1
1	and	1	1	1

0 = False    1 = True



# Logische Operatoren **and** und **or**

## - Beispiele

```
paketnummer = 10
```

```
if paketnummer % 4 == 0 or paketnummer % 5 == 0:  
    # Wird der Anweisungsblock ausgeführt?  
    print("Der Anweisungsblock wird ausgeführt.")
```



Ja, wird ausgeführt, da linke Bedingung nicht erfüllt, aber rechte Bedingung erfüllt.

Bedingung A	Bedingung B	ODER
0	1	1





# Logische Operatoren **and** und **or**

## - Beispiele

```
paketnummer = 10
```

```
if paketnummer % 4 == 0 or paketnummer % 5 == 0:  
    # Wird der Anweisungsblock ausgeführt?  
    print("Der Anweisungsblock wird ausgeführt.")
```



Ja, wird ausgeführt, da linke Bedingung nicht erfüllt, aber rechte Bedingung erfüllt.

Bedingung A	Bedingung B	ODER (or)
0	1	1



# Logische Operatoren **and** und **or**

## - Beispiele

```
paketnummer = 7  
  
if paketnummer % 4 == 0 or paketnummer % 5 == 0:  
    # Wird der Anweisungsblock ausgeführt?  
    print("Der Anweisungsblock wird ausgeführt.")
```



Nein, wird nicht ausgeführt, da beide Bedingungen nicht erfüllt.



# Logische Operatoren **and** und **or**

## - Beispiele

```
a = 15
```

```
if a > 10 and a < 20:
```

```
    # Wird der Anweisungsblock ausgeführt?
```

```
    print("Der Anweisungsblock wird ausgeführt.")
```



Ja, wird ausgeführt, da beide Bedingungen erfüllt.



# Logische Operatoren **and** und **or**

## - Beispiele

```
if False and True:  
    # Wird der Anweisungsblock ausgeführt?  
    print("Der Anweisungsblock wird ausgeführt.")
```



Nein, wird nicht ausgeführt, da linke Bedingung nicht erfüllt.



# Wie könnten Sie die Zuordnungs-Regeln im Programm umsetzen?

Nr.	Regel	
R1	Die Paketnummer ist restlos durch 4 oder 5 teilbar.	
R2	Die Paketnummer ist <b>zusätzlich</b> restlos durch 100 teilbar.	2
R3	Die Paketnummer ist <b>zusätzlich</b> restlos durch 400 teilbar.	3
R4	R1, R2 und R3 treffen nicht zu.	4

Was bedeutet  
zusätzlich?



# Wie könnten Sie die Zuordnungs-Regeln im Programm umsetzen?

Nr.	Regel	Fach
R1	Die Paketnummer ist restlos durch 4 oder 5 teilbar.	
R2	Die Paketnummer ist <b>zusätzlich</b> restlos durch 100 teilbar.	
R3	Die Paketnummer ist <b>zusätzlich</b> restlos durch 400 teilbar.	3
R4	R1, R2 und R3 treffen nicht zu.	4

Was bedeutet  
zusätzlich?



# Handlungsauftrag AP1: Entwickeln

- **Schreiben** Sie das entsprechende **Python-Skript** in einer neuen Code-Datei „Paketzuordnung\_AP1.py“.
- **Testen** Sie Ihr Programm während und nach der Entwicklung mit den **Testfällen** aus der Technischen Beschreibung.
- **Laden** Sie Ihre **Codedatei** (.py) mit Ihrem Nachnamen in MS Teams unter *Dateien > Schülerlösungen > Paket-Zuordnung AP1 hoch*.



**Zusammenarbeit mit Nachbar, 15 Minuten**



# Welche Arbeitsinhalte mussten für AP1 durchgeführt werden?

1. Entwurf des Programmablaufplans
2. Entwicklung der Software
3. Testen der Software
4. Dokumentation des Programmablaufs

Haben wir alles erledigt?





# Ihr Projektleiter ist zufrieden mit Ihrer Bearbeitung von AP1



Sie haben **gute Arbeit** geleistet in AP1.

Der **Kunde** wird sehr zufrieden sein.

Bearbeiten Sie als nächstes bitte die Arbeitsinhalte aus AP2.

