

# Übungsblatt 1 zum Tutorium Grundlagen der Programmierung im WS23/24

## 1. Übung

Implementieren Sie ein Glücksspiel, die geheime Zahl lautet 42. Lesen Sie eine geratene Zahl auf der Konsole ein. Prüfen Sie anschließend, ob die geratene Zahl der geheimen Zahl entspricht. Sofern die geratene Zahl kleiner ist, soll eine entsprechende Ausgabe auf der Konsole folgen. Ist die geratene Zahl größer soll ebenfalls eine entsprechende Ausgabe folgen. Wenn die Zahl gleich der geheimen Zahl ist, soll eine dritte Ausgabe erfolgen.

Zur Überprüfung Ihres Programms folgende Testausgaben:

### Test-Case 1:

```
Raten Sie die geheime Zahl: 42  
Die geheime Zahl war 42, Sie haben richtig geraten
```

### Test-Case 2:

```
Raten Sie die geheime Zahl: 13  
Ihre geratene Zahl war kleiner als die geheime  
Zahl
```

### Test-Case 3:

```
Raten Sie die geheime Zahl: 60  
Ihre geratene Zahl war größer als die geheime Zahl
```

## 2. Übung:

Sie sind Zulieferer für einen Rohstoff, Schreiben Sie ein Programm, welches die bestellte Menge vom Rohstoff entgegennimmt. Ab einer Menge von 100 ME gibt es 3% Rabatt. Ab einer Menge von 500 ME gibt es 5% Rabatt. Ab einer Menge von 1000 ME gibt es 7% Rabatt, der Rabatt bezieht sich auf die gesamte Mengeneinheit. Eine Mengeneinheit kostet 0,5€. Geben Sie die zu bezahlende Summe auf der Konsole aus.

Zur Überprüfung Ihres Programms folgende Testausgaben:

**Test-Case 1:**

```
Geben Sie die bestelle ME ein: 30  
Der Kunde muss folgenden Preis bezahlen: 15€
```

**Test-Case 2:**

```
Geben Sie die bestelle ME ein: 400  
Der Kunde muss folgenden Preis bezahlen: 194€
```

**Test-Case 3:**

```
Geben Sie die bestelle ME ein: 1200  
Der Kunde muss folgenden Preis bezahlen: 558€
```

### 3. Übung:

Erstellen Sie ein Programm bei dem der Benutzer eine Temperatur (warm, kalt) und Wetter (regnerisch, verschneit, sonnig) eingeben kann, beachten Sie, dass die Eingabe case sensitive ist. Jede Temperatur und Wetter Kombination soll eine entsprechende Kleidungsempfehlung ausgeben. Kombinationen, die logisch nicht möglich sind, sollen einen Hinweis bekommen, dass diese Kombination nicht möglich ist.

**Test-Case 1:**

```
Geben Sie die Temperatur an: warm  
Geben Sie das Wetter an: sonnig  
Ich empfehle ein T-Shirt und eine kurze Hose
```

**Test-Case 2:**

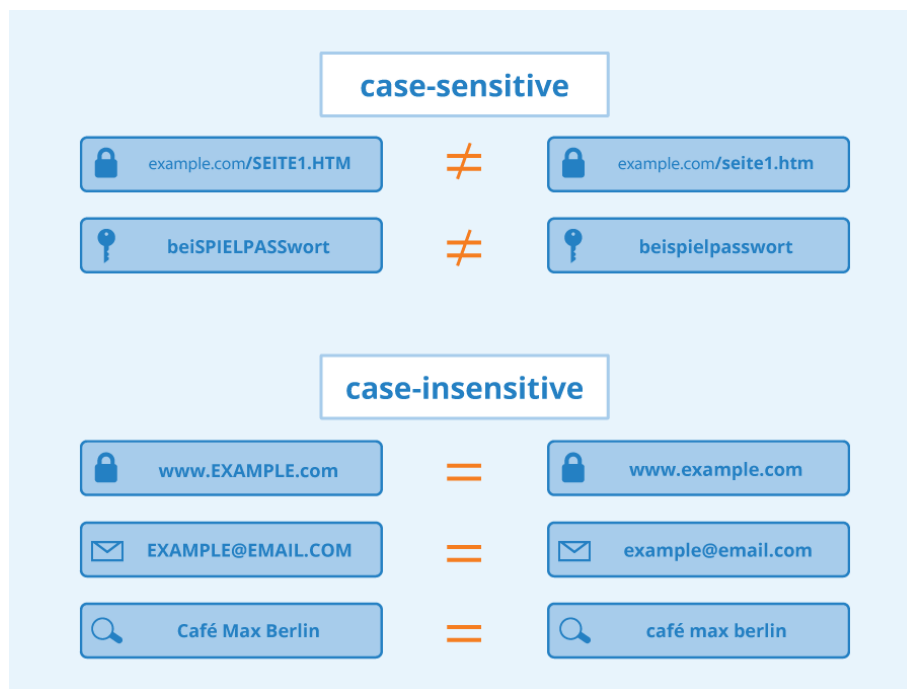
```
Geben Sie die Temperatur an: kalt  
Geben Sie das Wetter an: verschneit  
Ich empfehle eine dicke Jacke und festes Schuhwerk
```

**Test-Case 3:**

```
Geben Sie die Temperatur an: warm  
Geben Sie das Wetter an: verschneit  
Diese Wetter Kombination existiert nicht.
```

Hinweise:

- Eine Übersicht, was case sensitivity bedeutet:

Abbildung: Case Sensitivity - Autor: [Seobility](#)

## 4. Übung:

In Python können Sie die gesamte Logik mit AND, NOT und OR abbilden, bilden Sie für folgende Logikgatter die entsprechenden Wahrheitstabellen und wie die Logikgatter in Python umgesetzt werden können.

Gatter	Wahrheitstabelle	Python-Implementierung
<b>OR</b>	Wahr <b>OR</b> Wahr $\rightarrow$ Wahr	boolA or boolB
	Wahr <b>OR</b> Falsch $\rightarrow$ Wahr	
	Falsch <b>OR</b> Wahr $\rightarrow$ Wahr	
	Falsch <b>OR</b> Falsch $\rightarrow$ Falsch	

Gatter                      Wahrheitstabelle                      Python-Implementierung

**NAND**

Wahr **NAND** Wahr →  
Wahr **NAND** Falsch →  
Falsch **NAND** Wahr →  
Falsch **NAND** Falsch →

Gatter                      Wahrheitstabelle                      Python-Implementierung

**NOR**

Wahr **NOR** Wahr →  
Wahr **NOR** Falsch →  
Falsch **NOR** Wahr →  
Falsch **NOR** Falsch →

Gatter                      Wahrheitstabelle                      Python-Implementierung

**XNOR**

Wahr **XNOR** Wahr →  
Wahr **XNOR** Falsch →  
Falsch **XNOR** Wahr →  
Falsch **XNOR** Falsch →

Hinweise:

- Wenn Sie Schwierigkeiten beim Lösen des XNOR-Gatters haben, dann lösen Sie die Übungsaufgabe 4. Des zweiten Übungsblattes zuerst.
- XNOR ist die verneinte Version von XOR