

Programmieren Tutorium

Arbeitsblatt 5

Aufgabe 1: Schleifen - Quadrierte Zahlen

Schreibe ein Programm, das den Benutzer auffordert, eine Zahl einzugeben. Das Programm soll die Zahl so lange quadrieren und das Ergebnis ausgeben, bis das Ergebnis größer als 10.000 ist. Sobald das Ergebnis diesen Wert überschreitet, soll das Programm die Schleife beenden und eine Nachricht ausgeben, dass das Limit erreicht wurde.

Beispiel:

Eingabe: 3

Ausgabe: 3 9 81 6561

Aufgabe 2: Listen - Einkaufsliste, Element kopieren

Anforderungen:

- Erstelle eine Liste von string-Elementen, die als Einkaufsliste dient (z. B. ["Milch", "Brot", "Eier"]).
- Kopiere das erste Element der Liste und füge es ans Ende der Liste hinzu.
- Gib die aktualisierte Einkaufsliste auf der Konsole aus.

Beispiel:

Eingabe: Einkaufsliste: ["Äpfel", "Bananen", "Orangen"]

Ausgabe: ["Äpfel", "Bananen", "Orangen", "Äpfel"]

Hinweis: Nutze `.Add(element)` um ein Element einer Liste hinzuzufügen

Hinweis: Überprüfe, ob die Liste mindestens ein Element enthält, bevor du das erste Element hinzufügst, um Fehler zu vermeiden.

Aufgabe 3: Zeichenketten - Ersetze Zeichen im String

Die Variable `text` ist ein string, der mehrere Zeichen enthält. Ersetze in diesem string jedes Minuszeichen (-) durch ein Pluszeichen (+) und gib den neuen string zurück.

Anforderungen:

- Verwende den Variablennamen `text` statt `txt` für eine bessere Lesbarkeit.
- Iteriere über alle Zeichen im string und ersetze jedes - durch ein +.
- Gib den veränderten string am Ende aus.

Beispiel:

Eingabe: `text = "Hallo-Welt-Text"`

Ausgabe: `"Hallo+Welt+Text"`

***Hinweis:** Du kannst den string in ein char-Array umwandeln, um über die einzelnen Zeichen zu iterieren und Änderungen vorzunehmen. In C# geht das zum Beispiel mit `text.ToCharArray()`.*

Aufgabe 4: Funktionen - Ausgabe mit Print

Erstelle eine Funktion namens `Print`, die einen string-Parameter als Eingabe erhält und diesen in einer Zeile auf der Konsole ausgibt. Die Funktion soll keine Rückgabe haben und lediglich den übergebenen Text anzeigen.

Anforderungen:

- Die Funktion `Print` soll einen string-Parameter akzeptieren.
- Die Funktion soll den übergebenen string in einer Zeile auf der Konsole ausgeben.
- Es dürfen keine weiteren Funktionen oder Methoden verwendet werden, die das Ausgabeverhalten verändern.

Beispiel:

Eingabe: `Eingabe: "Hallo, Welt!"`

Ausgabe: `Hallo, Welt!`

Aufgabe 5: Funktionen - Addiere und Runde

Schreibe ein Programm, das zwei `double`-Zahlen einliest, ihre Summe berechnet und das Ergebnis auf die nächste ganze Zahl rundet. Das Programm soll den gerundeten Wert als Ganzzahl zurückgeben.

Anforderungen:

- Erstelle eine Funktion, die zwei `double`-Parameter akzeptiert.
- Berechne die Summe der beiden `double`-Werte.
- Runde das Ergebnis auf die nächste ganze Zahl.
- Gib das gerundete Ergebnis als `int` zurück.

Beispiel:

Eingabe: `AddiereUndRunde(3.7, 2.5)`

Ausgabe: 6

***Hinweis:** Verwende eine Methode zum Runden auf die nächste ganze Zahl (z. B. `Math.Round` in C#).*

Aufgabe 6: Durchschnitt der Zahlen über einem Schwellenwert berechnen

Schreibe eine Funktion `DurchschnittSchwelle(int[] array, int schwelle)`, die den Durchschnitt der Zahlen berechnet, die über einem bestimmten Schwellenwert liegen.

Anforderungen:

- Die Funktion `DurchschnittSchwelle` soll ein Array von Ganzzahlen und einen Schwellenwert als Parameter akzeptieren.
- Summiere nur die Zahlen, die den Schwellenwert überschreiten.
- Berechne und gib den Durchschnitt dieser Zahlen zurück.

Beispiel:

Eingabe: Array = [10, 20, 30, 40], Schwelle = 15

Ausgabe: 30 // da nur 20, 30 und 40 größer als 15 sind und deren Durchschnitt 30 ist.

***Hinweis:** Summiere nur die Zahlen, die den Schwellenwert überschreiten, und zähle sie, um am Ende den Durchschnitt zu berechnen.*