

Aufgabe 1: Vor- und Nachname kombinieren

Ziel: Das Einlesen von mehreren `string`-Eingaben und deren Verknüpfung üben.

- Schreibe ein Programm, das den Benutzer zuerst nach seinem Vornamen fragt.
- Lies den Vornamen ein.
- Frage den Benutzer anschließend nach seinem Nachnamen.
- Lies den Nachnamen ein.
- Kombiniere beide Eingaben zu einem vollständigen Namen (mit einem Leerzeichen dazwischen).
- Gib den vollständigen Namen in einem Satz aus, z. B.: "Ihr voller Name lautet: Max Mustermann".

Aufgabe 2: Alter in Tagen berechnen

Ziel: Typumwandlung und einfache arithmetische Operationen.

- Frage den Benutzer nach seinem Alter in Jahren.
- Wandle die Eingabe von einem `string` in einen `int` um.
- Berechne das Alter in Tagen (ignoriere dabei Schaltjahre, ein Jahr hat 365 Tage).
- Gib das Ergebnis in einem vollständigen Satz aus, z. B.: "Du bist ungefähr [Anzahl] Tage alt."

Aufgabe 3: Der Währungsrechner (Einfach)

Ziel: Mit `double` oder `float` arbeiten und das Ergebnis formatieren.

- Lege einen festen Wechselkurs in einer `const`-Variable fest (z. B. `const double EuroToDollar = 1.08;`).
- Frage den Benutzer nach einem Euro-Betrag.
- Konvertiere die Eingabe in einen `double`.
- Berechne den entsprechenden Betrag in US-Dollar.
- Gib das Ergebnis aus: "[Betrag] Euro entsprechen [Ergebnis] US-Dollar."

Aufgabe 4: Gerade oder Ungerade?

Ziel: Den Modulo-Operator (`%`) und eine `if-else`-Bedingung verwenden.

- Bitte den Benutzer, eine ganze Zahl einzugeben.
- Prüfe, ob die Zahl gerade oder ungerade ist.
 - **Tipp:** Eine Zahl ist gerade, wenn sie ohne Rest durch 2 teilbar ist.
- Gib das Ergebnis entsprechend aus.

Aufgabe 5: Ticketpreis-Ermittlung

Ziel: Logische Operatoren (`&&`, `||`) und eine `if-else if-else`-Kette nutzen.

- Frage den Benutzer nach seinem Alter.
- Bestimme den Ticketpreis basierend auf folgenden Regeln:
 - Kinder (unter 6 Jahren): frei
 - Schüler (6 bis 17 Jahre): 5€
 - Erwachsene (18 bis 64 Jahre): 10€
 - Senioren (ab 65 Jahren): 7€
- Gib den passenden Preis auf der Konsole aus.

Aufgabe 6: Einfacher Taschenrechner

Ziel: Eine `switch`-Anweisung zur Steuerung des Programmflusses verwenden.

- Frage den Benutzer nach einer ersten Zahl.
- Frage ihn nach einem Operator (`+`, `-`, `*`, `/`).
- Frage ihn nach einer zweiten Zahl.
- Verwende eine `switch`-Anweisung, um basierend auf dem Operator die richtige Berechnung durchzuführen.
- Gib das Ergebnis der Operation aus.
- Füge einen `default`-Fall hinzu, der eine Fehlermeldung ausgibt, wenn ein ungültiger Operator eingegeben wurde.

Aufgabe 7: Noten in Text umwandeln

Ziel: Eine `switch`-Anweisung für eine klare Zuordnung nutzen.

- Der Benutzer soll eine Schulnote als Zahl (1-6) eingeben.
- Das Programm soll die Note in Text umwandeln:
 - 1: "Sehr gut"
 - 2: "Gut"
 - 3: "Befriedigend"
 - 4: "Ausreichend"
 - 5: "Mangelhaft"
 - 6: "Ungenügend"
- Gib die textuelle Bewertung aus.
- Wenn eine andere Zahl eingegeben wird, soll eine Meldung wie "Ungültige Note" erscheinen.

Aufgabe 8: Schaltjahr-Rechner

Ziel: Komplexe logische Bedingungen mit `if-else` und `&&` / `||` umsetzen.

- Bitte den Benutzer, eine Jahreszahl einzugeben.
- Prüfe, ob es sich um ein Schaltjahr handelt. Die Regeln dafür sind:
 1. Ein Jahr ist ein Schaltjahr, wenn es durch 4 teilbar ist.
 2. **Ausnahme:** Wenn das Jahr auch durch 100 teilbar ist, ist es **kein** Schaltjahr.
 3. **Ausnahme von der Ausnahme:** Wenn das Jahr jedoch durch 400 teilbar ist, ist es **doch** ein Schaltjahr.
- Gib aus, ob das eingegebene Jahr ein Schaltjahr ist oder nicht.
- **Beispiele:** 2000 (ist ein Schaltjahr), 2024 (ist ein Schaltjahr), 1900 (kein Schaltjahr), 2023 (kein Schaltjahr).

Aufgabe 9: Taschenrechner mit Fehlerabfrage

Ziel: Bestehende Logik erweitern und Randfälle (Edge Cases) behandeln.

- Erweitere den "Einfachen Taschenrechner" aus Aufgabe 6.
- Füge eine `if`-Bedingung hinzu, die speziell den Fall der Division durch Null abfängt.
- Wenn der Benutzer versucht, durch `0` zu teilen, soll statt eines Fehlers eine verständliche Meldung ausgegeben werden, z. B. "Fehler: Division durch Null ist nicht erlaubt!". Die Berechnung soll dann nicht ausgeführt werden.

Aufgabe 10: Einfache Login-Simulation

Ziel: String-Vergleiche und verschachtelte `if`-Bedingungen anwenden.

- Lege einen Benutzernamen und ein Passwort als `const string`-Variablen im Code fest (z. B. `const string correctUser = "admin";`).
- Frage den Benutzer nacheinander nach seinem Benutzernamen und seinem Passwort.
- Überprüfe die Eingaben und gib unterschiedliche Meldungen aus:
 - Wenn der Benutzername falsch ist: "Benutzername nicht gefunden."
 - Wenn der Benutzername korrekt, aber das Passwort falsch ist: "Falsches Passwort."
 - Wenn beides korrekt ist: "Login erfolgreich! Willkommen."
- **Tipp:** Hierfür benötigst du eine verschachtelte `if`-Struktur. Die äußere prüft den Namen, die innere das Passwort.