



## Übungsblatt 6

22.05., 23.05. und 26.05

### Problem 6.1: Funktionen mit Rückgabewert

Nun kommen Funktionen, die einen Rückgabewert berechnen.

1. Schreiben Sie eine Funktion `berechne_BMI` mit zwei Parametern `groesse` und `gewicht` vom Typ `double`, deren Rückgabetyt ebenfalls `double` ist und deren Rückgabewert der wie in Aufgabe 2.1 als

$$\text{BMI} = \frac{\text{Gewicht [kg]}}{(\text{Körpergröße [m]})^2}$$

berechnete BMI ist (ohne die Plausibilitätsüberprüfungen aus 2.1.2 und 2.1.3).

In der Funktion `main` soll der Benutzer Größe und Gewicht eingeben können, zu dem dann (durch Aufruf von `berechne_BMI`) der BMI berechnet und ausgegeben wird.

2. Die nächste Funktion soll `berechne_BMI_Text` heißen, einen Parameter BMI vom Typ `double` und Rückgabetyt `string` haben. Sie soll den Text wie in Aufgabe 2.1.4 zurückgeben – also *nicht* selber mittels `cout` ausgeben.

(Copy&Paste aus Ihrem Programm zu Aufgabe 2.1 bietet sich hier an – wenn Sie das nicht mehr zur Hand haben, erfinden Sie einfach eine vereinfachte Klassifizierung; alle Fälle neu einzutippen, dürfte hier nicht hilfreich sein. Wie immer, wenn wir den Datentyp `string` verwenden, muss ggf. noch die `string`-Bibliothek eingebunden werden.)

Ergänzen Sie die Funktion `main` so, dass der zu den Eingaben passende Text (durch Aufruf von `berechne_BMI_Text`) berechnet, in einer Variablen gespeichert und mit ausgegeben wird.

Probieren sie zwei Varianten aus: Eine, bei der Sie in `main` den berechneten Wert des BMI in einer Variablen speichern und eine, in der Sie den Wert innerhalb des Aufrufs von `berechne_BMI_Text` noch einmal berechnen (der übergebene Aktualparameter kann ein beliebiger Ausdruck passenden Typs sein, insbesondere ein Aufruf von `berechne_BMI`).

### Problem 6.2: Referenzparameter

1. Schreiben Sie eine Funktion `tausch` ohne Rückgabewert mit zwei `int`-Referenzparametern `a` und `b` (zu Referenzparametern s. Breymann, Kap. 2.2.2).

In der Funktion sollen die beiden Werte auf `cout` ausgegeben werden, dann sollen die Inhalte der beiden Variablen vertauscht werden (`a` bekommt den ursprünglichen Inhalt von `b` und umgekehrt – eine lokale Hilfsvariable ist dabei nützlich) und noch einmal ausgegeben werden.

In der Funktion `main` soll der Benutzer zwei `int`-Variablen `x` und `y` eingeben können, sie sollten ausgegeben werden, `tausche(x, y)` soll aufgerufen werden und zum Schluss sollen beide Variablen noch einmal ausgegeben werden.

Ein Dialog mit dem Programm könnte dann so aussehen (Benutzereingaben unterstrichen):

`x: 6`

`y: 2021`

In `main()`, vor dem Tausch: `x = 6` , `y = 2021`

In `tausch()`, vor dem Tausch: `a = 6`, `b = 2021`

In `tausch()`, nach dem Tausch: `a = 2021`, `b = 6`

In `main()`, nach dem Tausch: `x = 2021`, `y = 6`

Und nun noch folgendes Experiment: Entfernen Sie die `&` aus der Parameterliste von `tausch`. Wird das Programm noch fehlerfrei übersetzt? Falls ja: Was tut es nun?

2. Um den Inhalt von drei Variablen `a`, `b` und `c` (z.B. vom Typ `int`) zu sortieren (`a` soll hinterher den kleinsten, `b` den mittleren und `c` den größten Wert enthalten), kann man so vorgehen:
  - Falls  $a > b$ , vertauschen Sie die Inhalte von `a` und `b`.
  - Falls nun (ggf. nach dem Tausch vom vorigen Schritt)  $b > c$  ist, vertauschen Sie die Inhalte von `b` und `c`.
  - Falls Sie im letzten Schritt vertauscht haben, und nun  $a > b$  ist, vertauschen Sie die Inhalte von `a` und `b`.

Zeichnen Sie das Struktogramm zu diesem Verfahren!

Wieso funktioniert das Verfahren – d.h., warum sind die Zahlen am Ende sortiert?

Implementieren Sie das Verfahren nun mit einer Funktion `sortiere` ohne Rückgabewert mit drei `int`-Referenzparametern `a`, `b` und `c`. Verwenden Sie zum Tauschen die Funktion `tausch` aus 1.

Probieren Sie die Funktion mit passenden Testfällen aus!