

## Wann wird welche Methode verwendet:

- 1) Wenn man eine Variable durch die Funktion ändern will, geht **nur** die lvalue-Referenz.
- 2) Wenn man einen Parameter nicht ändern will, verwendet man die Wertübergabe (wenn das Argument kopiert werden kann und nicht viel Speicher belegt), in den übrigen Fällen die `const lv`-Referenzmethode (diese benötigt immer genau 8 Bytes Speicher für das Argument = Größe einer Speicheradresse) und kann auch Objekte übergeben, die nicht kopierbar sind.
- 3) Die rvalue-Referenz verwendet man, wenn man ein temporäres Objekt übergeben und dieses in der Funktion „ausrauben“ will !?

## Funktionen: mehrere Argumente

Natürlich kann eine Funktion mehr als 1 Argument definieren und für jedes einzelne von ihnen die Wert-, Referenz- (lvalue oder rvalue) oder `const-lv`-Referenz vereinbaren. Bei jedem Aufruf müssen entsprechend viele Argumente angegeben sein!

```
void addiere(double a, const double& b, double& summe)
{
    summe = a + b;
}
```

```
...
double x;
addiere(1, 2, x); // korrekt: x hat jetzt den Wert 3.0
```

1 wird in 1.0 umgewandelt und per Wert übergeben

2 wird in 2.0 umgewandelt, temporär gespeichert und als konstante lv-Referenz übergeben

x wird per lv-Referenz übergeben und dient hier eigentlich als Rückgabewert der Summe, es ist ein sog. Outparameter.

Auf diese Weise lassen sich von einer Funktion mehrere Resultate zurückgeben: Eines (oder keines) davon als Rückgabewert und die restlichen mittels lv-Referenz-Argumenten!

Allerdings wird diese Methode nicht gerne gesehen. Manche Firmen verbieten sogar solche Konstruktionen.