

## 0x03 – Übungen

---

### 'Ravencastle'

Eine Klasse 'bruch' ('fraction') entwerfen.

- Implementieren Sie einen default-ctor und einen dtor.
- Implementieren Sie einen ctor, der Zähler und Nenner übergeben bekommt. Nutzen Sie ':' zur Initialisierung.
- Implementieren Sie 'getter' und 'setter' für (ganzzahlige) Zähler bzw. Nenner.
- Schreiben Sie aussagekräftigen Testcode.

Erweiterung:

- Implementieren Sie einen copy-ctor.
- Implementieren Sie einen operator '«' zur Ausgabe.
- Verwenden Sie so oft wie möglich 'const'.

## 0x03 – Übungen

---

### 'Stone Ridge'

Eine Klasse 'punkt' mit 'x'- und 'y'-Koordinate entwerfen.

- Implementieren Sie einen default-ctor und einen dtor.
- Implementieren Sie einen weiteren ctor, der einen x- und einen y-Wert vom Typ 'double' übergeben bekommt. Benutzen Sie ':' zur Initialisierung.
- Implementieren Sie 'getter' und 'setter' für 'x' und 'y'.
- Schreiben Sie aussagekräftigen Testcode.

Erweiterung:

- Implementieren Sie einen copy-ctor.
- Implementieren Sie einen operator '«' zur Ausgabe.
- Verwenden Sie so oft wie möglich 'const'.

## 0x03 – Übungen

---

### 'Lucky Rock'

Eine Klasse 'kontakt' mit Alter und Namen entwerfen.

- Implementieren Sie einen default-ctor und einen dtor.
- Implementieren Sie einen ctor, der ein ganzzahliges Alter vom Typ 'unsigned int' und einen Namen vom Typ 'string' übergeben bekommt. Nutzen Sie ':' zur Initialisierung.
- Implementieren Sie 'getter' und 'setter'.
- Schreiben Sie aussagekräftigen Testcode.

Erweiterung:

- Implementieren Sie einen copy-ctor.
- Implementieren Sie einen operator '«' zur Ausgabe.
- Verwenden Sie so oft wie möglich 'const'.

### 'Meadow River'

- Überführen Sie die Struktur 'polynom' aus Übung 'Elkford' in eine Klasse 'polynom'.
- Überführen Sie die Funktion 'eval' in eine Memberfunktion.
- Implementieren Sie eine Memberfunktion 'at', die einen Index 'i' übergeben bekommt und den i'ten Koeffizienten zurückgibt. Werfen Sie eine Ausnahme, falls 'i' ungültig ist.

Erweiterung:

- Implementieren Sie eine globale Funktion 'add', um zwei Polynome zu addieren und geben Sie das Ergebnis zurück.
- Nutzen Sie Ihren Testcode aus 'Elkford' und testen Sie zusätzlich mit der Dimension 4.

## 0x03 – Übungen

---

### Hausübung

'work\_class\_car'. Nur ansehen.

### Selbstkontrolle

- Ich habe alle Codes und Übungsthemen verstanden.
- Ich kenne den Unterschied zwischen 'struct' und 'class'.
- Ich weiß, wie man eine Klasse definiert.
- Ich kann Konstruktoren, Destruktoren und 'getter' und 'setter' definieren.
- Ich kenne die Bedeutung von 'const' im Zusammenhang mit Memberfunktionen.
- Ich weiß, wann ein Kopierkonstruktor aufgerufen und wie er verwendet wird.
- Ich kann einen eigenen Operator '«' für meine Klasse schreiben.