0x09 – Übungen

’Eastbourne’

Template Spezialisierung

- Nehmen Sie Ihre Punkt-Klasse, spezialisieren Sie sie für bool und machen dort den ctor privat, so dass Sie keine Instanz davon erzeugen können.

0x09 – Übungen

’Openshaw’

unique- und shared-ptr

- Entwerfen Sie eine beliebige Klasse und füllen Sie (jeweils) einen std::vector
  - mit Objekten Ihrer Klasse,
  - mit (raw) Zeigern auf dynamisch angelegte Objekte Ihrer Klasse, bzw.
  - mit unique- und shared-ptr auf diese.

Erweiterung:

- Werden alle Elemente ordnungsgemäß zerstört oder gibt es Memory Leaks?
- Können Sie alle Elemente in einer Schleife ausgeben?

0x09 – Übungen

’Wintervale’

Metaprogramming

- Schreiben Sie eine Template-Klasse, die zur Compile-Zeit B^N (B hoch N) für zwei natürliche Zahlen B und N berechnet.
- Schreiben Sie aussagefähigen Testcode.

Erweiterung:

- Spezialisieren Sie Ihre Klasse für den Fall B=1.

0x09 – Übungen

’Banrockburn’

Anwendung shared-ptr ggf. weak-ptr

- Implementieren Sie eine generische doppelt verkettet Liste mit smart-Zeigern anstelle von raw-Zeigern.
- Entwerfen Sie eine Template-Klasse node für die Daten und einen Zeiger auf das nächste Element, und einen auf das vorhergehende Element, sowie eine Template-Klasse list für die Liste mit einem root Zeiger auf das erste, und einem tail-Zeiger auf das letzte Element.
- Implementieren Sie einen += Operator zum Anhängen neuer Elemente.
- Schreiben Sie aussagefähigen Testcode und iterieren Sie über Ihre Liste (vorwärts und rückwärts).

0x09 – Übungen

’Clarcton’

advanced!

- Entwerfen Sie eine eigene smart-Ptr Klasse, die sich wie unique- bzw. shared-ptr verhält.