Programmieren IIB: 2. Hausaufgabenblatt

Winter 2013/2014
Prof. Dr. Marzena Fügenschuh

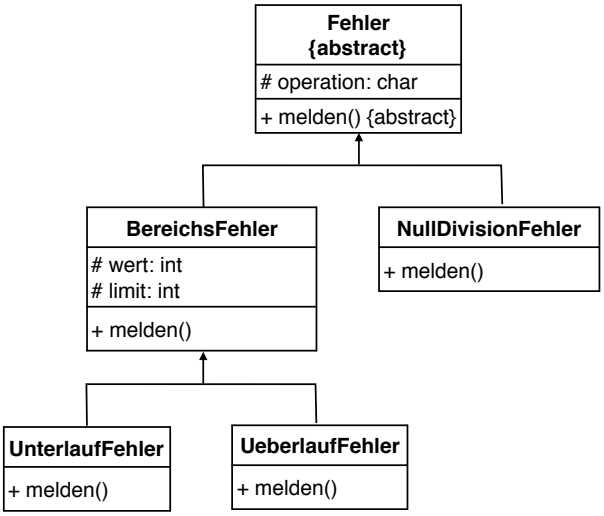
BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

Ausnahmebehandlung, STL

Ausgabe: 19. Juni 2019
Abgabe: 7. Juli 2019

1. Aufgabe (5 Punkte): Ausnahmebehandlung

Die Klasse `Winzig` erlaubt das Rechnen ausschließlich in einem vorgegebenen Bereich der ganzen Zahlen. Um dies zu gewährleisten implementieren Sie die folgende Ausnahmebehandlung[1] und verwenden entsprechend in der Klasse `Winzig`.



1. Ein Unterlauf- bzw. Überlauf-Fehler wird ausgelöst, wenn bei einer Rechenoperation die untere bzw. obere Bereichsgrenze unterschritten wird.
2. Der zu kleine bzw. zu große Wert wird im Attribut `wert` der Klasse `Bereichsfehler` gespeichert. Das Attribut `limit` bezeichnet die überschrittene Bereichsgrenze.
3. Die Methode `melden` gibt ausreichend detailierte Informationen an, was für eine Ausnahme aufgetreten ist.

In *moodle* finden Sie den Header der Klasse `Winzig` als Vorlage für die Implementierung wie auch ein Beispiel der Ausgabe des Test-Programms `Winzig-Main.cpp` zur Orientierung, wie detailliert die Ausnahmen beschrieben werden sollen. Sie können gerne das Test-Programm umgestalten.

[1] S. Vorlesungsskript, falls Sie diese UML-Darstellung der Klassenhierarchie nicht nachvollziehen können.

2. Aufgabe (3 Punkte): STL

Kopieren Sie in ein neues Projekt die letzte Version der Klasse `Datum` samt der Ausnahmebehandlungsklassen, s. die Lösung zu 3. Übungsblatt/2. Übung.

1. Implementieren Sie die folgende Methode `setHeute` für die Klasse `Datum`. (Binden Sie hierzu die Bibliothek `ctime` ein!)

```
void Datum::setHeute() {
  time_t zeit = time(NULL);
  tm heute = *localtime(&zeit);
  tag   = heute.tm_mday;
  monat = heute.tm_mon + 1;   // Zaehlung beginnt bei 0
  jahr  = heute.tm_year + 1900;   // Zaehlung beginnt bei 1900
}
```

2. In der Datei `Datum-Main.cpp` legen Sie einen leeren `vector`-Container, der `Datum`-Objekte speichern soll.

   (a) Füllen Sie ihn mit 10 Zufällig erstellten Terminen, oder mit allen Klausur- und Test-Terminen für Programmieren 2a und 2b.

   (b) Fügen Sie den Termin mit dem heutigen Datum an der 3. und 7. Position ein.

   (c) Löschen Sie die letzten 3 Termine.

   (d) Geben Sie nach jeder Operation den Inhalt des Containers aus.

3. Überladen Sie die Operatoren `<, >, ==` in der Klasse `Datum`.

4. Binden Sie die Bibliothek `algorithm` im Kopf der Datei `Datum-Main.cpp` ein.

   (a) Sortieren Sie alle Termine im Container.

   (b) Löschen Sie alle Termine mit dem heutigem Datum aus dem Container.

   (c) Geben Sie alle Termine aus, die in der Zukunft liegen;

   (d) Verschieben Sie alle Termine um einen Tag.

Beispiele zur Nutzung von Standardklassen `vector` und `algorithm` finden Sie im Skriptum.

Abgabe 7. Juli 2019

Alle Dateien zu einer Aufgabe in einem separaten Ordner speichern, beide Ordner zippen und hochladen.