



Programming by Contract

WS 2003/2004 – Übungsblatt 4

4. Dezember 2003

Ausgabe: 24. November 2003

Aufgabe 1. *Integer Under-/Overflows*

Schreiben Sie das Pascal-Beispiel `function fakultaet(n : integer) : integer` aus Abschnitt 3.2 der Vorlesung nach C um und testen Sie. Benutzen Sie dabei die Version, die einen Overflow abfängt.

Gehen Sie analog vor beim Beispiel zur Berechnung einer nichtnegativen ganzzahligen Potenz einer ganzen Zahl nach dem Algorithmus von Seite 29 der Materialsammlung. Erzeugen Sie dabei im Falle der Berechnung 0^0 eine Exception.

Aufgabe 2. *Overflow-Abfangen*

Wie sehen die Fallunterscheidungen zur Vermeidung von INTEGER-Overflows und -Underflows bei den vier Grundrechenarten, bei `sqrt()` und bei `sqr()` aus?

(Dabei sei `sqr()` eine selbstdefinierte Funktion zur Berechnung des Quadrats eines ganzzahligen Arguments.)

Welche Konstanten aus welcher Headerdatei benötigen Sie?

Aufgabe 3. *assert*

Ergänzen Sie die Klasse `Day` von Aufgabe 3 des letzten Übungsblatts um geeignete `assert`-Anweisungen, um Vor- und Nachbedingungen sowie Invarianten bei jeder Methode (inklusive den Konstruktoren) zu überprüfen.

Ergänzen Sie zwei geeignete Template-Funktionen Ihres letzten C++-Kurses um eine `CT_ASSERT`-Anweisung und testen Sie.

Aufgabe 4. *unordered arguments*

Wie können Sie bei C++-Programmen den Problemfall von zwei “unordered” Argumenten beim Operator `>=` richtig in die Hand bekommen?