

TI1200: Extra opdracht voor ervaren programmeurs

Je kan programmeren? Je hebt ervaring met Java? Je hebt nood aan een uitdaging? En, je wil een punt extra kunnen verdienen?

Context: we weten allemaal dat computers heel goed zijn in het maken van berekeningen, het zijn uiteindelijk heel grote rekenmachines. Toch is dat eigenlijk niet zo evident, want als je gaat kijken wat er binnenin een computer allemaal gebeurt voor het maken van die wiskundige berekeningen dan lopen er eigenlijk wel een paar dingetjes fout.

Laten we even kijken... Wij stellen getallen voor d.m.v. het decimale getalstelsel, bijvoorbeeld: 10, 22 of 0.33. Computers stellen die getallen voor in het binaire getalstelsel, wat voor de voorgaande getallen betekent dat ze er zo uitzien:

10 → 1010

22 → 10110

0.33 → 0.0001100110011001100110011001100110011001100110011...

Wat is hier nu het geval? 0.33 is niet exact voorstelbaar in het zogenaamde floating-point formaat dat computers hanteren. In programmeertalen heb je een zogenaamde *float* (ook wel *single* genoemd) of *double* om komma getallen (floating points) voor te stellen. Een float wordt voorgesteld door 32 bits en een double door 64 bits. Het precieze formaat is als volgt:

Float: 32 bits: 24 bits voor de significand, 7 bits voor de exponent en 1 bit voor het teken

Double: 64 bit: 53 bits voor de significand, 10 bits voor de exponent en 1 bit voor het teken

De waarde van een getal wordt dan $s \times b^e$, met de s significand, b de getalbasis en e de exponent.

De manier waarop getallen worden voorgesteld houdt dus in dat sommige getallen niet precies kunnen worden voorgesteld in de computer, maar dat er afrondingsfouten in de voorstelling van het getal zitten. Voor een float kunnen er ongeveer 7 decimale cijfers “exact” worden voorgesteld, voor een double zijn dit er 16.

Wat is nu jullie opdracht? Maak in Java een programma dat exact rekt (en dus geen gebruik maakt van floats en doubles). Maak ook geen gebruik van BigDecimal of gelijkaardige klassen, maar maak enkel gebruik van primitieve types (int, boolean) en arrays (geen String, Integer, Vector of andere collectieklassen). Zorg ervoor dat je minstens de optelling, aftrekking en vermenigvuldiging implementeert en eventueel ook de deling.

Voor deze opdracht heb je de tijd tot **11 oktober 2012 23:59**. Je kan indienen via Blackboard.

Dien je project in als een zip via Blackboard. Indien je beschikt over een volledig werkende implementatie (zonder fouten) dan krijg je 1 punt extra voor je tentamen van het vak TI1200 (het maximum voor het vak is en blijft uiteraard 10). Let erop, dit is individueel werk. Enige vorm van plagiaat wordt bestraft!

Specificatie:

Maak een klasse "CalculateExactly" met onderstaande specificatie:

```
class CalculateExactly {  
    char[] add(char[] a, char[] b);  
  
    char[] subtract(char[] a, char[] b);  
  
    char[] multiply(char[] a, char[] b);  
  
    char[] divide(char[] a, char[] b);  
}
```

Lees door middel van System.in gebruikersinput in:

- laat de gebruiker eerst beslissen of hij/zij een optelling, aftrekking, vermenigvuldiging of deling wil uitvoeren.
- daarna komen de 2 getallen
- Lees deze komma-getallen (met of zonder teken) in, zonder precisie-verlies. Je mag hiervoor een String gebruiken (en dit is ook de enige plek in je programma waar je een niet-primitief type mag gebruiken)