

Werkcollege 8

Calculator

Calculator is een grafische rekenmachine applicatie geschreven in Java. Met de applicatie kan je twee getallen optellen, aftrekken, vermenigvuldigen of delen.

De applicatie werkt als volgt:

- De gebruiker vult een aantal cijfers in door op de cijferknoppen te drukken.
- De cijfers verschijnen in het scherm
- De gebruiker kiest een operatie. Getal 1 is nu definitief. De operatie verschijnt ook in het scherm.
- De gebruiker vult met behulp van de cijferknoppen het tweede getal in: dit verschijnt ook op het scherm
- De gebruiker drukt op de '=' knop. Het tweede getal is nu definitief. In het scherm wordt de '=' ook op het scherm getoond gevolgd door de correcte uitkomst.
- De gebruiker kan nu opnieuw beginnen door opnieuw cijfers in te toetsen

Opdracht:

Schrijf de applicatie **Calculator**.

- Foute invoer of invoer op het foute moment hoeft niet afgehandeld te worden
- Delen door 0 dient een waarschuwing op het scherm te geven.
- Alle rekenoperaties dienen uitgevoerd te worden door methoden van een aparte klasse Calculator (en dus niet rechtstreeks in de GUI klasse)
- De 4 methoden van de klasse calculator moeten worden ge-unittest.
- *Uitdaging*: maak ook een '+/-' knop waarmee het teken van een getal 'omgeklapt' kan worden voordat de operatie of '=' gekozen is.

Stappen:

1. Start een nieuw netbeans Java project
2. Maak in dit project een nieuwe Java klasse aan van type JFrame Form
 - a. Verwijder de oorspronkelijke *main* klasse (of zorg er bij 1 voor dat er geen default bestand wordt gemaakt)
 - b. Bij dit bestand kan in het editor window geswitched worden tussen de code en de grafische weergave (design) door de knoppen linksboven.
3. Sleep een heleboel JButtons op het scherm één JLabel (het scherm)
 - a. Mak een mooie layout voor je applicatie
 - b. Geef alle knoppen de juiste naam (langzaam dubbelklikken)
 - c. Geef elk schermelement een herkenbare variabele naam (rechtsklik). Denk aan de juiste CamelCase
4. De applicatie zal een aantal dingen moeten onthouden (bv de getallen en de gekozen operator). Maak hiervoor attributen aan.
 - a. Geef ze in de constructor de juiste waarde.
 - b. Geef ook het scherm een goede startwaarde
5. Maak een extra klasse aan die het echte rekenwerk gaat doen (Calculator). Zorg dat er in de GUI klasse een object van wordt aangemaakt dat we kunnen gebruiken als op '=' wordt gedrukt.

- a. Schrijf ook de methoden die het rekenwerk doen. Denk ook weer aan de juiste CamelCase
 - b. Bedenk nu alvast hoe je straks het probleem met delen door 0 gaat afhandelen
6. Maak de applicatie werkend door voor elke knop een methode te schrijven die de juiste actie onderneemt. Deze methode wordt automatisch gemaakt als je in *design* snel dubbelklikt op de betreffende knop
 - a. Bedenk dat er (wellicht) iets anders moet gebeuren als de operator al dan niet gekozen is. Gebruik methoden om code duplication te voorkomen!
 - b. Onderzoek zelf hoe je strings aan elkaar kunt plakken
 - c. Onderzoek zelf hoe je de inhoud van je JLabel verandert (op internet, javadoc of code completion)
 - d. Bedenk dat bij het drukken op '=' de klasse Calculator gebruikt moet worden. Bedenk ook dat na het drukken van '=' de 'situatie' gereset moet worden. Misschien is een methode hiervoor geschikt (die je dan ook weer in de constructor kan gebruiken)
7. Schrijf uitgebreide unittests voor de klasse Calculator
 - a. Rechtsklik → Tools → create tests
 - b. Test alle vier je methoden met 7 testcases (lage getallen, hoge getallen, 0 links, 0 rechts, links negatief, rechts negatief, beide getallen negatief). Gebruik hierbij:

```
Calculator temp = new Calculator();
assert(temp.telop(1,2)==3);
//etc...
```

Bedenk zelf hoe je de deel-methode afhandelt. Sowieso heeft delen een double als resultaat. Onderzoek zelf hoe je doubles correct vergelijkt.
 - c. Voer de test uit (rechtsklik → test file)
8. Doe de uitdaging.