Les 1 lesprogramma

# Opgave Delen

*Individueel, programmeervaardigheid*

## Onderdeel A

Wanneer je twee gehele getallen door elkaar deelt, krijg je niet automatisch een kommagetal, terwijl dat soms wel wenselijk is.

Bijvoorbeeld: println(5 / 2); levert 2, terwijl het handig kan zijn dat er 2.5 uit zou komen.

Maak een functie, floatDelen, die twee gehele getallen heeft als invoer. De functie deelt het eerste getal door het tweede getal en retourneert het resultaat als een kommagetal.

## Onderdeel B

Test het programma door in het hoofdprogramma de functie een paar keer aan te roepen, het resultaat naar het scherm te schrijven en te kijken of het resultaat op het scherm overeen komt met het resultaat dat je verwacht.

Tags: datatypes, testen, methodeaanroep <> methodedefinitie, parameters, argumenten, returnwaarden, functie, methode

# Opgave Globaal delen

Klassikaal, discussie, begrip

Gegeven onderstaande oplossing:

int getal1, getal2;

float resultaat;

void setup() {

getal1 = 5;

getal2 = 2;

floatDelen();

println(resultaat);

}

void floatDelen() {

float f1 = (float)getal1;

float f2 = (float)getal2;

resultaat = f1 / f2;

}

Bovenstaande oplossing werkt wel, maar is niet ideaal. Geef een nadeel van deze uitwerking.

Tags: globale variabelen <> lokale variabelen, returnwaarden, parameters, argumenten

# Opgave Delen door nul

Klassikaal, discussie, begrip

Onderstaande prorgamma’s veroorzaken drie verschillende foutenmeldingen:

## A

void setup() {

println(floatDelen(5, 2);

}

## B

void setup() {

println(floatDelen(5, 2.0));

}

## C

void setup() {

println(1 / 0);

}

Hoewel er drie verschillende fouten worden veroorzaakt, kun je deze drie fouten in twee soorten onderverdelen. Welke twee fouten horen bij dezelfde soort en welke fout is van de andere soort. Waarom kies je voor deze indeling?

Verzin een passende naam voor beide soorten foutmelding

Tags: runtime, , delen door 0, syntax, compile-time, datatypes, statische types

# Opgave Zoeken in een array

Individueel, programmeervaardigheid

De functie komtGetalVoorIn(int getal, int[] lijst) retourneert true als getal voorkomt in lijst. Als getal niet voorkomt in lijst, dan wordt er false geretourneerd.

## Onderdeel A

Maak eerst een programma waarmee je de functie komtGetalVoorIn kunt testen:

1. Verzin waarden voor de twee parameters van de functie waarvan je de uitkomst van de functieaanroep uit het hoofd kunt bepalen.
2. Roep de functie aan in setup van processing (zonder de functie te definiëren).
3. Schrijf het resultaat van deze functieaanroep naar de console en controleer of je de foutmelding krijgt die aangeeft dat de functie nog niet bestaat.

## Onderdeel B

Implementeer de functie komtGetalVoorIn en controleer of het testresultaat overeenkomt met wat je verwacht.

## Onderdeel C

Klassikaal, testen

Wanneer is een test goed?

Tags: testen (eerst), array, testen, methodeaanroep <> methodedefinitie, return

# Opgave doeFunctie?

Klassikaal, codebegrip

Gegeven onderstaande code waarin een functie voorkomt met een onbruikbare functienaam.

int[][] hetVeld = {

{1, 6, 3},

{3, 2, 9},

};

void setup() {

println(doeFunctie(hetVeld, 1));

}

int doeFunctie(int[][] a, int b) {

int c = 0;

int[] d = a[b];

for (int i = 0; i < d.length; i++) {

c += d[i];

}

return c;

}

## Onderdeel A

Loop de functie, doeFunctie, regel voor regel door en houd bij welke variabelen er zijn en wat de waarde van elke variabele is (in feite speel je nu zelf de runtime omgeving na). Probeer er op deze manier te achter te komen wat de functie doet.

## Onderdeel B

Verander in de aanroep van doeFunctie het tweede argument in 2 (ipv 1). Welk soort foutmelding krijg je nu? Wat betekent de foutmelding?

## Onderdeel C

Als je de variabele a, b, c of d zou aanroepen in setup, dan gaat dit mis. Andersom (hetVeld aanroepen in doeFunctie) gaat wel goed. Hoe komt dit? Geef in de runtime administratie die je bij opgave A hebt gemaakt aan welke variabelen bij welke functie horen.

Tags: runtime <> compile-time, stap voor stap doorlopen, runtime administratie, runtime omgeving, ArrayIndexOutOfBoundsException, Exception, locale variabele

# Opgave doeKeerTwee

Tweetallen, begrip

int testGetal = 5;

int[] testGetallen = {5, 5};

void setup() {

doeKeerTwee(testGetal);

doeKeerTwee(testGetallen);

println(testGetal);

println(testGetallen);

}

void doeKeerTwee(int getal) {

getal = 2 \* getal;

}

void doeKeerTwee(int[] getallen) {

for (int i = 0; i < getallen.length; i++) {

getallen[i] = 2 \* getallen[i];

}

}

Voer dit programma uit en bekijk de uitvoer. Vergelijk de waarde van het testGetal met de waarde in testGetallen. Wat is het belangrijkste verschil?

In de eerstvolgende les besteden we aandacht aan dit verschil.

Tags: referentietypes, primitieve types, runtime administratie, overloading

Oefeningen

# Opgave Arrays bouwen

*Programmeervaardigheid*

## Opgave A getallen uit twee arrays bij elkaar optellen

Maak de methode telElementenOp die twee integer arrays als invoer heeft. De methode telt de getallen van elk element uit beide arrays bij elkaar op en slaat de resultaten op in een nieuwe array. Vervolgens retourneert de methode de nieuwe array.

Test deze methode in de setup-methode van processing.

*Hint: ga ervan uit dat beide arrays dezelfde lengte hebben*

## Opgave B maximum bepalen van twee arrays

Maak de methode maakMaxArray. Deze methode krijgt twee arrays als invoer, bepaalt de methode per element uit beide arrays het getal met de hoogste waarde en retourneert een nieuwe array met deze getallen.

Als één van beide arrays langer is dan de ander, dan moet het resultaat aangevuld worden met de elementen uit de langste array.

Test deze methode in de setup-methode van processing.