

SPD Oefentoets module 1 t/m 8

Versie 2.1

Bevat:

- 4 toetsvragen
- Bijbehorend nakijkmodel

Opgave 1 (35 punten)

Gegeven een array met integers. Vul onderstaand programma aan zodat bepaald wordt wat het gemiddelde is van alle getallen in een integer array en afdrukt hoeveel getallen in de array groter zijn dan dit gemiddelde. Daarnaast worden alle getallen die groter zijn dan het gemiddelde afgedrukt.

Eisen:

- Je mag de gegeven code in de setup-methode niet wijzigen of aanvullen.
- Vermijd alle dubbele code.
- Voeg nieuwe methode(n), met eventueel parameters en/of returnwaardes, toe als dat de werking en/of structuur van het programma ten goede komt. Neem deze methoden duidelijk op in je uitwerking.
- Hergebruik, als het mogelijk is, reeds eerder gemaakte methoden.
- Wanneer de array "waarden" andere getallen bevat of meer/minder elementen bevat, moet jouw oplossing blijven werken.

Gegeven is de setup-methode met aanroepen van de te implementeren methoden (deze zijn vet gedrukt):

```
void setup() {  
    int[] waarden = { 6, 5, 8, 3, 25, 4, 13, 21, 11, 9};  
    float gem = berekenGemiddelde(waarden);  
    println("Er zijn " + aantalGroterDanGemiddelde(waarden) +  
            " groter dan het gemiddelde van " + gem);  
    int[] groter = groterDanGemiddelde(waarden);  
    println(groter);  
}
```

Opdracht

Schrijf/implementeer alle vetgedrukte methoden en eventuele extra methoden zodat het programma op een efficiënte en correcte manier werkt.

Opgave 2 (15 punten)

Gegeven de volgende code:

```
int a = 4;
int b = 7;

void setup() {
    println("Setup: " + knutsel(a, b));
}

int knutsel(int b, int a) {
    println("Knutsel: " + a);
    println("Knutsel: " + b);
    return kwispel(a, 2*b) + b;
}

int kwispel(int a, int b) {
    println("Kwispel: " + a);
    println("Kwispel: " + b);
    return a - b;
}
```

Opdracht: Schrijf exact op wat er in de text console wordt afgedrukt.

Opgave 3 (20 punten)

Onderstaand programma is een implementatie van een dobbelspel, er wordt steeds met twee dobbelstenen (met 1 t/m 6 ogen) gegooid totdat er met beide dobbelstenen hetzelfde aantal ogen wordt gegooid. Daarna wordt afgedrukt hoe vaak er gegooid is met twee dobbelstenen.

Het programma, zie hieronder, bevat helaas diverse fouten.

```
1: void setup() {
2:     int aantalWorpen = 0;
3:     int eersteWorp = dobbel();
4:     int tweedeWorp = dobbel();
5:     while(eersteWorp <> tweedeWorp) {
6:         aantalWorpen++;
7:         eersteworp = dobbel();
8:         tweedeworp = dobbel();
9:     }
10:    println("Er is " + aantalWorpen + " keer gegooid.");
11: }
12:
13: int dobbel() {
14:     return round( random(0,7) );
15: }
```

Opdracht: Geef per fout aan op welke regel die staat en wat de fout is.

Opgave 4 (30 punten)

Voor een vak moeten studenten 3 toetsen doen. Het eindcijfer van het vak wordt als volgt berekend. Als een student één van de toetsen nog niet gemaakt heeft (dus geen cijfer heeft), dan is het eindcijfer een 1. Heeft een student voor één van de toetsen een cijfer lager dan een 5.5 gehaald, dan is het eindcijfer altijd een 4. Heeft een student voor alle toetsen een cijfer en elk cijfer is een 5.5 of hoger, dan is het eindcijfer het afgeronde gemiddelde.

Een deel van het programma is gegeven:

```
float[][] deelcijfers = { { 8.9, 9.0, 9.1 },
                          { 9.0 },
                          { 8.4, 2.1, 4.3 },
                          { 7.2, 6.4, 5.6 },
                          { 8.6, 7.9, 4.5 },
                          { 7.2, 2.3 }
                        };

final int AANTAL_TOETSEN = 3;
final int NIET_ALLE_CIJFERS = 1;
final int NIET_ALLES_VOLDOENDE = 4;
final float VOLDOENDE = 5.5;

void setup() {
    drukAlleEindcijfersAf();
    int aantal = aantalIncompleteCijfers();
    println(aantal + " studenten hebben nog niet alle cijfers");
}
```

Het bovenstaande programma heeft als uitvoer in de text console:

```
Alle eindcijfers zijn:
9
1
4
6
4
1
2 studenten hebben nog niet alle cijfers
```

Verdere eisen en aanvullingen:

- Je mag de gegeven code in `setup` niet wijzigen of aanvullen.
- Vermijd alle dubbele code.
- Je mag wel nieuwe methoden (met eventueel parameters en/of returnwaardes), variabelen etc. toevoegen als dat de werking en/of structuur van het programma ten goede komt. Neem deze methoden duidelijk op in je uitwerking.
- Houd er in je code rekening mee dat de array van deelcijfers een willekeurige lengte kan hebben.

Opdracht: Implementeer de methoden `drukAlleEindcijfersAf` en `aantalIncompleteCijfers`

Reference:

Bij de toets krijg je een deel van de Processing reference. Bij deze oefentoets laten we dat achterwege.

Correctiemodel

Op de volgende pagina vind je het correctiemodel. Het maken van een oefentoets heeft weinig effect als je te snel naar de voorbeelduitwerkingen kijkt. Maak eerst deze toets in zijn geheel op papier, kijk daarna je werk pas na.

Correctiemodel oefentoets module 1 t/m 8 SPD**Opgave 1 (max 35 pt)**

Mogelijke uitwerking:

```

void setup() {
    int[] waarden = { 6, 5, 8, 3, 25, 4, 13, 21, 11, 9};
    float gem = berekenGemiddelde(waarden);
    println("Er zijn " + aantalGroterDanGemiddelde(waarden) +
        " groter dan het gemiddelde van " + gem);
    int[] groter = groterDanGemiddelde(waarden);
    println(groter);
}

float berekenGemiddelde(int[] waarden)                                3pt
    float som = 0.0;
    for (int i = 0; i < waarden.length; i++) {                      1pt
        som += waarden[i];                                           1pt
    }
    return som/waarden.length;                                       2pt
}

int aantalGroterDanGemiddelde(int[] waarden) {                      3pt
    float gemiddelde = berekenGemiddelde(waarden);                  3pt
    int aantal = 0;
    for (int i = 0; i < waarden.length; i++) {                      2pt
        if (gemiddelde < waarden[i]) {                              2pt
            aantal++;                                                2pt
        }
    }
    return aantal;                                                  1pt
}

int[] groterDanGemiddelde(int[] waarden) {                          3pt
    float gemiddelde = berekenGemiddelde(waarden);                  2pt
    int aantal = aantalGroterDanGemiddelde(waarden);                4pt
    int[] resultaat = new int[aantal];                              3pt
    int index = 0;                                                  2pt
    for(int i=0; i<waarden.length; i++){                            1pt
        if (gemiddelde < waarden[i]) {                              1pt
            resultaat[index] = waarden[i];                          2pt
            index ++;                                                2pt
        }
    }
    return resultaat;                                                2pt
}

```

- Achter elke regel staat het aantal punten dat in mindering moet worden gebracht als deze niet correct is.
- **-10pt** als het berekenen van het gemiddelde niet hergebruikt wordt
- **-10pt** als het berekenen van het aantal niet hergebruikt wordt
- Punten per code-regel alleen toekennen als deze volledig goed is.
- Natuurlijk nooit minder dan 0 punten voor deze vraag in zijn geheel geven.

Opgave 2 (max 15 pt)

Volgende komt in de console te staan

```
Knutsel: 7
Knutsel: 4
Kwispel: 7
Kwispel: 8
Setup: 3
```

- **7pt** voor volgorde goed (alles of niets)
- **4pt** als alle getallen correct zijn
- **4pt** als de getallen bij de correcte println staan

Opgave 3 (max 20pt)

```
1: void setup() {
2:   int aantalWorpen = 0;           // moet 1 zijn           4pt
3:   int eersteWorp = dobbel();
4:   int tweedeWorp = dobbel();
5:   while(eersteWorp <> tweedeWorp) { // <> moet != zijn      4pt
6:     aantalWorpen++;
7:     eersteworp = dobbel();         // hoofdletter W       3pt
8:     tweedeworp = dobbel();         // idem                 1pt
9:   }
10:  println("Er is " + aantalWorpen + " keer gegoooid.");
11: }
12:
13: int dobbel() {
14:   return round( random(0,7) ); //round moet floor zijn en 0 moet 1 zijn
                                   floor: 4pt
                                   random(1,7): 4pt
15: }
```

- Elke onterecht aangemerkte fout **-2pt**
- Natuurlijk nooit minder dan 0 punten voor deze vraag in zijn geheel geven.

Opgave 4 (30 punten)

```
float[][] deelcijfers = { { 8.9, 9.0, 9.1 },
                          { 9.0 },
                          { 8.4, 2.1, 4.3 },
                          { 7.2, 6.4, 5.6 },
                          { 8.6, 7.9, 4.5 },
                          { 7.2, 2.3 }
                        };

final int AANTAL_TOETSEN = 3;
final int NIET_ALLE_CIJFERS = 1;
final int NIET_ALLES_VOLDOENDE = 4;
final float VOLDOENDE = 5.5;

void setup() {
    drukAlleEindcijfersAf();
    int aantal = aantalIncompleteCijfers();
    println(aantal + " studenten hebben nog niet alle cijfers");
}

void drukAlleEindcijfersAf() {
    println("Alle eindcijfers zijn: ");
    for(int i=0; i<deelcijfers.length; i++) {
        println(eindcijfer(deelcijfers[i]));
    }
}

int eindcijfer(float[] toetscijfers) {
    if(toetscijfers.length < AANTAL_TOETSEN) {
        return NIET_ALLE_CIJFERS;
    }
    else {
        float totaal = 0;
        for(int i=0; i<toetscijfers.length; i++) {
            // of i < AANTAL_TOETSEN
            if(toetscijfers[i] < VOLDOENDE) {
                return NIET_ALLES_VOLDOENDE;
            }
            else {
                totaal += toetscijfers[i];
            }
        }
        return round(totaal / toetscijfers.length);
        // of round(totaal / AANTAL_TOETSEN)
    }
}

int aantalIncompleteCijfers() {
    int aantal = 0;
    for(int i=0; i<deelcijfers.length; i++) {
        if(eindcijfer(deelcijfers[i]) == NIET_ALLE_CIJFERS) {
            aantal++;
        }
    }
    return aantal;
}
```

1pt
1pt
2pt
2pt
5pt
2pt
2pt
1pt
2pt
2pt
3pt
2pt
1pt
2pt
2pt
3pt
1pt
1pt

Natuurlijk nooit minder dan 0 punten voor deze vraag in zijn geheel geven.