

SPD MODULE 6: HERHALINGEN/LOOPS

LESSTOF

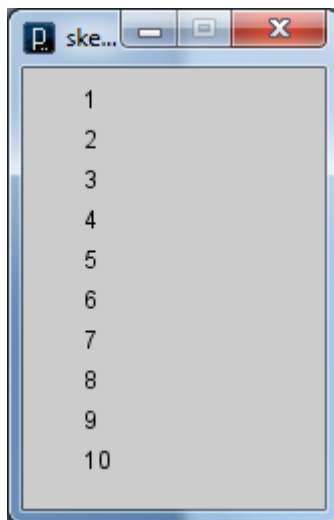
Screencasts: <https://www.youtube.com/playlist?list=PL075F0B7F35120D4E>

Reader: Hoofdstuk 6

OEFENOPGAVEN

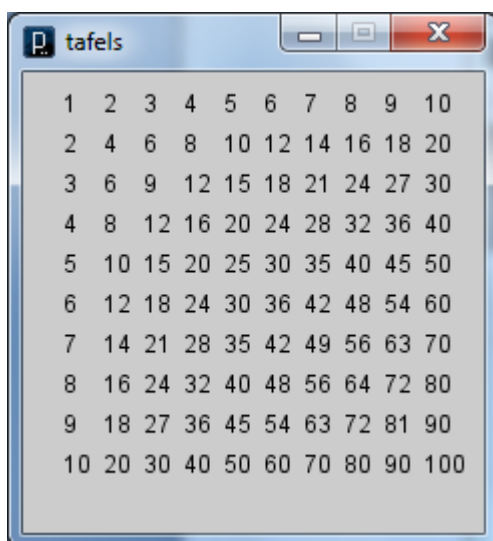
OPGAVE 6.1

Schrijf een programma dat de tafel van 1 afdruckt op het scherm.



OPGAVE 6.2

Schrijf een programma dat de tafels van 1 t/m 10 afdruckt op het scherm.



OPGAVE 6.3

Schrijf een programma dat een gecentreerd vierkant opvult met kleine vierkantjes. De opgegeven waarden zijn de breedte van de kleine vierkantjes en aantal vierkantjes per zijde. Bijvoorbeeld zo:

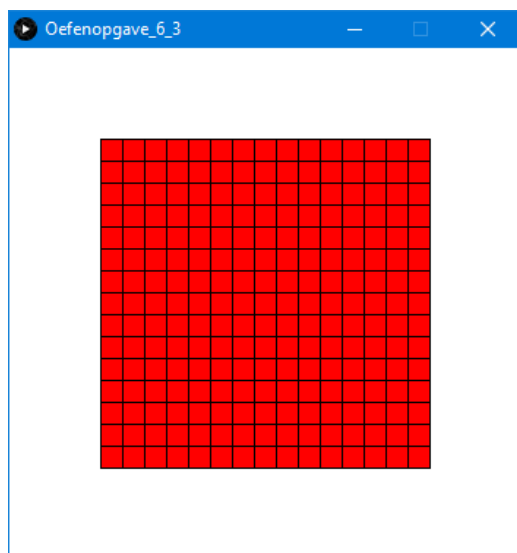
```
final int ZIJDE = 15;
final int AANTAL = 15;

void setup() {
  size(350, 350);
  background(#FFFFFF);

  // Hier jouw code om de vierkantjes te tekenen
}
```

Het woordje `final` betekent dat de variabele niet gewijzigd kan worden in code, het is eigenlijk een constante geworden. Er is een afspraak gemaakt onder programmeurs dat je constanten in hoofdletters schrijft. Je hebt er al een paar gezien bij eerdere opdrachten. Bijvoorbeeld om tekst te centreren kun je `textAlign(CENTER, CENTER);` gebruiken, waarbij `CENTER` een door Processing gegeven constante is.

Het grote vierkant staat altijd precies in het midden van het scherm. Geef een foutmelding als vierkant niet meer in het scherm past.



MODULEOPGAVE 6: PRINSESJES TELLEN

Alle koninklijke echtparen in Europa willen 1 mannelijke erfgenaam.

Als er een jongetje geboren wordt in een koninklijk huwelijk, nemen de ouders daarna geen kinderen meer. Als er een meisje geboren wordt, nemen de ouders nog een kind net zo lang totdat het volgende kind een jongetje is.

LET OP: Maak voor beide onderdelen eerst een analyse en ontwerp. Leg dit vast in een document en begin dan pas met implementeren.

ONDERDEEL 1

Schrijf een programma dat een gezin op één rij tekent en meerdere gezinnen kan tekenen.

Gebruik de ingebouwde methode `random` om te bepalen of er een jongetje, of een meisje getekend wordt en gebruik externe afbeeldingen voor de jongetjes en meisjes.

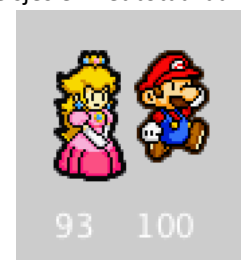
Hieronder is een voorbeeld te zien van 8 gezinnen.



ONDERDEEL 2

Maak een nieuw programma dat de gezinnen niet tekent, maar het totaal aantal meisjes en het totaal aantal jongens bijhoudt. Genereer vervolgens 100 gezinnen, of meer, en controleer of het aantal meisjes ongeveer even groot is als het aantal jongens. Zie screenshot hiernaast.

Je mag de waarden ook afdrukken in de tekst console.



EXTRA OPGAVEN: OPDELEN VAN CODE IN FUNCTIES

EXTRA OPGAVE 1: CASUS PRIEMGETALLEN

Gegeven is de onderstaande code waarmee voor de getallen 2 tot 1000 wordt bepaald of het om een priemgetal gaat. Indien dit het geval is, wordt daarvan een melding op het console geschreven.

```
void setup() {
    for (int huidigGetal = 2; huidigGetal < 1000; huidigGetal++) {
        boolean priemgetal = true;

        for (int deler = 2; deler < huidigGetal; deler++) {
            if (huidigGetal % deler == 0) {
                priemgetal = false;
            }
        }

        if (priemgetal) {
            println(huidigGetal + " is een priemgetal!");
        }
    }
}
```

OPDRACHT

Herschrijf de code, zodat zo veel mogelijk gebruik gemaakt wordt van korte functies die elk een duidelijke taak hebben. Let op de namen van de functies, en denk goed na over parameters en return-waarden. Het is uiteraard niet de bedoeling om globale variabelen te introduceren.

EXTRA OPGAVE 2: CASUS GOKSPELLETJE

Gegeven is onderstaande code waarmee een gokspelletje wordt gespeeld. Het idee is dat iemand met een bepaald saldo start, een inzet doet en er een muntje wordt gegooid. In het geval van kop wordt het saldo met de inzet verhoogd, in het geval van munt, wordt het saldo met de inzet verlaagd. De speler gaat net zo lang door tot hij ofwel geen geld meer heeft, ofwel een bepaald maximumbedrag behaald heeft.

```
void setup() {
    while ((saldo > 0) && (saldo < maximumbedrag)) {
        gooiMuntje();
        nieuwSaldo();
        System.out.println(saldo);
    }
}

void gooiMuntje() {
    if (random(1) > 0.5) {
        gewonnen = true;
    } else {
        gewonnen = false;
    }
}

void nieuwSaldo() {
    if (gewonnen) {
        saldo += inzet;
    } else {
        saldo -= inzet;
    }
}
```

```
}
```

OPDRACHT A

Herschrijf de code, zodat zo veel mogelijk gebruik gemaakt wordt van korte functies die elk een duidelijke taak hebben. Let op de namen van de functies, en denk goed na over parameters en return-waarden. Probeer om globale variabelen zo veel mogelijk uit het programma te verwijderen. Introduceer echter geen “magic numbers”. Dat zijn getallen die zomaar ineens in je code opduiken zonder dat het aan de lezer van je code duidelijk is wat ze betekenen (in deze opdracht zou je bijvoorbeeld de variabele `maximumbedrag` kunnen verwijderen en er gewoon het getal 5000 neerzetten, maar daardoor wordt je code minder inzichtelijk).

OPDRACHT B

Herschrijf de code die je in opdracht A gemaakt hebt, zodat deze uitgevoerd wordt in een methode met de naam `speelSpelletje` en de parameters `saldo`, `inzet` en `maximumbedrag`, en als returnwaarde het eindbedrag (dat is dus altijd 0 of de waarde van `maximumbedrag`) die door onderstaande code kan worden aangeroepen om het hele spelletje vanaf begin tot einde meerdere keren te kunnen spelen.

```
public void setup() {  
    int aantalRondes = 10;  
    int totaal = 0;  
    int saldo = 1000;  
    int inzet = 10;  
    int maximumbedrag = 5000;  
  
    for (int i=0; i < aantalRondes; i++) {  
        totaal += speelSpelletje(saldo, inzet, maximumbedrag);  
    }  
    System.out.println("Totaal: " + totaal);  
}
```