



Van Algoritme naar PSD

Inleiding

We kunnen in principe alle dagelijkse handelingen omzetten in PSD's. In deze les zullen we naar een aantal van deze alledaagse voorbeelden kijken.

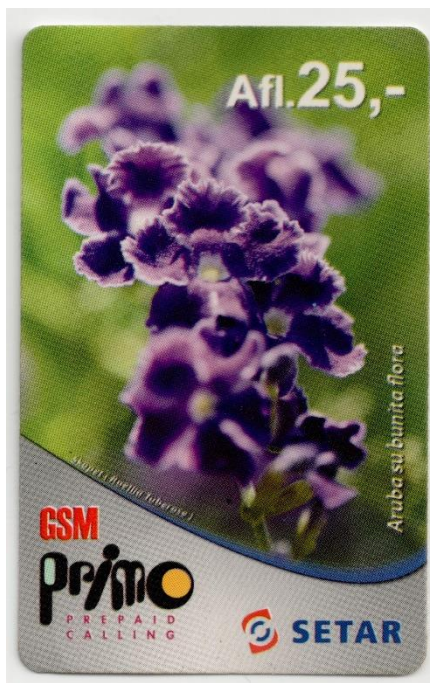
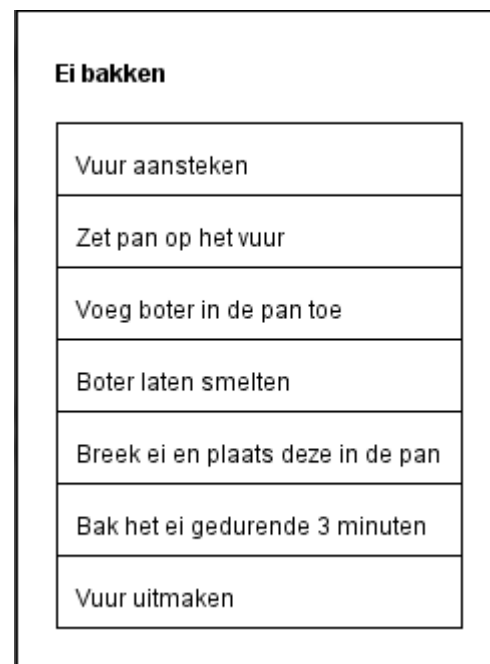
Ei bakken

In de vorige les hebben we gekeken naar het bakken van een ei. Het algoritme was toen weergegeven als geschreven tekst (recept). De opdracht is om nu een PSD hiervan te maken.

Recept Ei bakken

1. Vuur aansteken
2. Zet pan op het vuur
3. Voeg boter in de pan toe
4. Boter laten smelten
5. Breek ei en plaats deze in de pan
6. Bak het ei gedurende 3 minuten
7. Vuur uitmaken

PSD



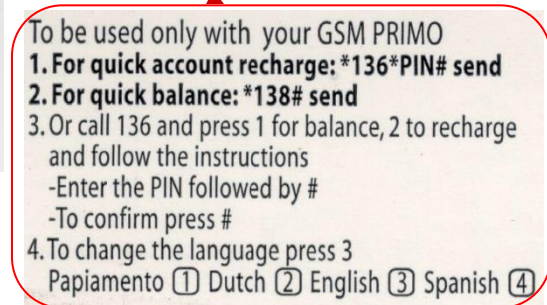
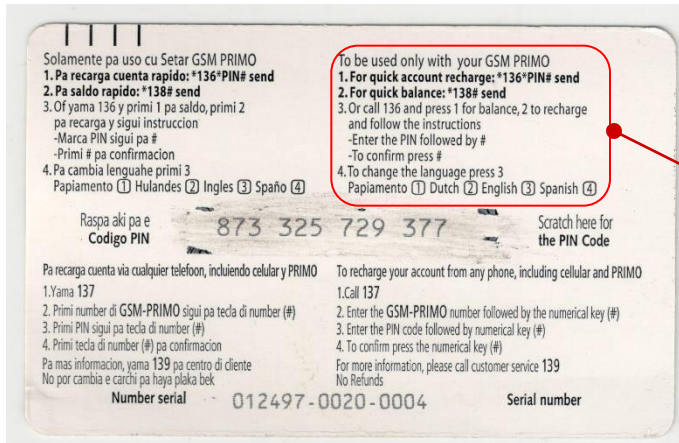
Een ander veel gebruikt algoritme tegenwoordig is geld toevoegen aan een mobiele telefoon doormiddel van een telefoonkaart waar op de achterkant een code staat die moet worden ingevoerd in de telefoon.

Hiernaast zie je de afbeelding van de voorkant van zo'n kaart.



Van Algoritme naar PSD

Op de achterkant staan de instructies hoe je het bedrag dient toe te voegen aan de mobiele telefoon.



Deze stappen vormen het algoritme om het geld te kunnen toevoegen. Het PSD ziet er dan als volgt uit:

PSD

Quick Account Recharge
Scratch for replacement pin
Enter *136*PIN number#
Press send

De uitleg op de achterkant bestaat namelijk uit vier verschillende handelingen, ofwel vier verschillende algoritmen.

1. Geld toevoegen
2. Saldo opvragen
3. Geld toevoegen of opvragen door te bellen.
4. De taal veranderen van de automaat.

Opdracht: Maak het PSD voor het opvragen van het saldo van een mobiele telefoon.



Van Algoritme naar PSD

Iedereen kent het spelletje "Raad het getal" waar één persoon een getal kiest tussen bijvoorbeeld 0 en 100 en de andere persoon(en) het moeten proberen te raden. Het enige dat gezegd wordt is of het gerade getal hoger of lager is dan het getal dat de persoon in gedachte heeft.

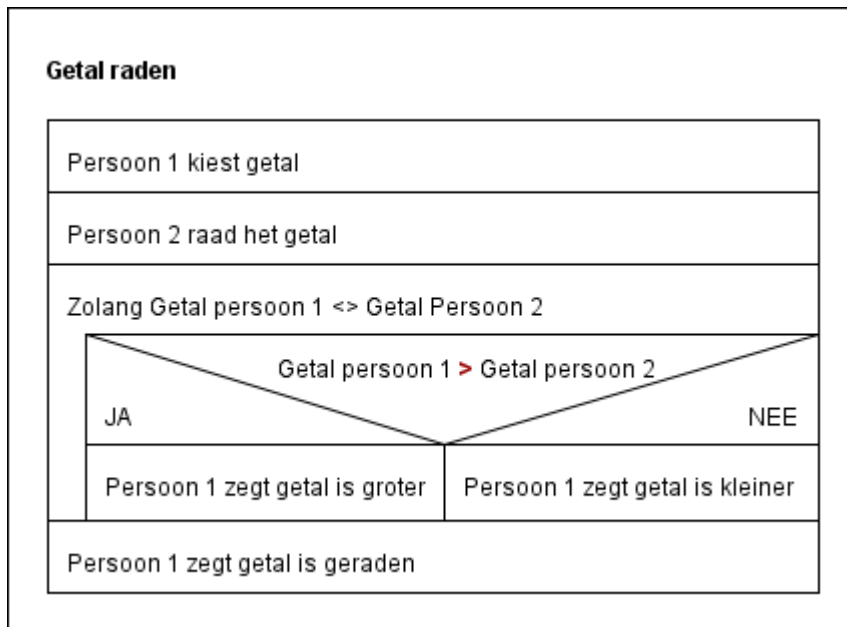
Opdracht: Schrijf het algoritme in stappen op die het verloop van dit spel speelt.

Uitwerking:

1. Persoon 1 kiest een getal tussen 0 en 100
2. Persoon 2 raad het getal
3. Persoon 1 geeft aan of het geraden getal hoger, lager of geraden is.
4. Is het getal hoger of lager dan raad persoon 2 opnieuw.
5. Is het getal geraden dan eindigt het spel.

Opdracht: Maak de PSD voor dit spel.

Uitwerking PSD:



Stel nu dat persoon 1 de computer is en een persoon speelt het spel met de computer. De computer moet nu dus een willekeurig (random) getal verzinnen tussen 0 en 100.

Daarna moet de computer vragen en de persoon om het getal te raden. Waarna de computer gaat controleren of het getal van de persoon groter of



Van Algoritme naar PSD

kleiner is dan die van de computer of dat de persoon het getal van de computer geraden heeft.

Om de PSD te maken voor dit spel hebben we variabelen nodig. Eentje die de waarde van het door de computer gekozen getal bewaard en eentje die het getal van de persoon die raad vast houdt.

Gebruik, "TeRadenGetal" als naam voor de variabele die het getal dat de computer kiest vast houdt en "GeradenGetal" als variabele van het getal van de persoon die aan het raden is.

De persoon krijgt 10 kansen om het getal te raden. Als het niet geraden is dan geeft het programma het juiste antwoordt en eindigt het spel.

Opdracht: Maak de PSD voor dit spel op de computer. Voor deze opdracht zijn alle drie de structuren (sequentie, selectie en iteratie) noodzakelijk.

In de afbeelding hieronder zie je hoe dit programma er uit zou komen te zien:

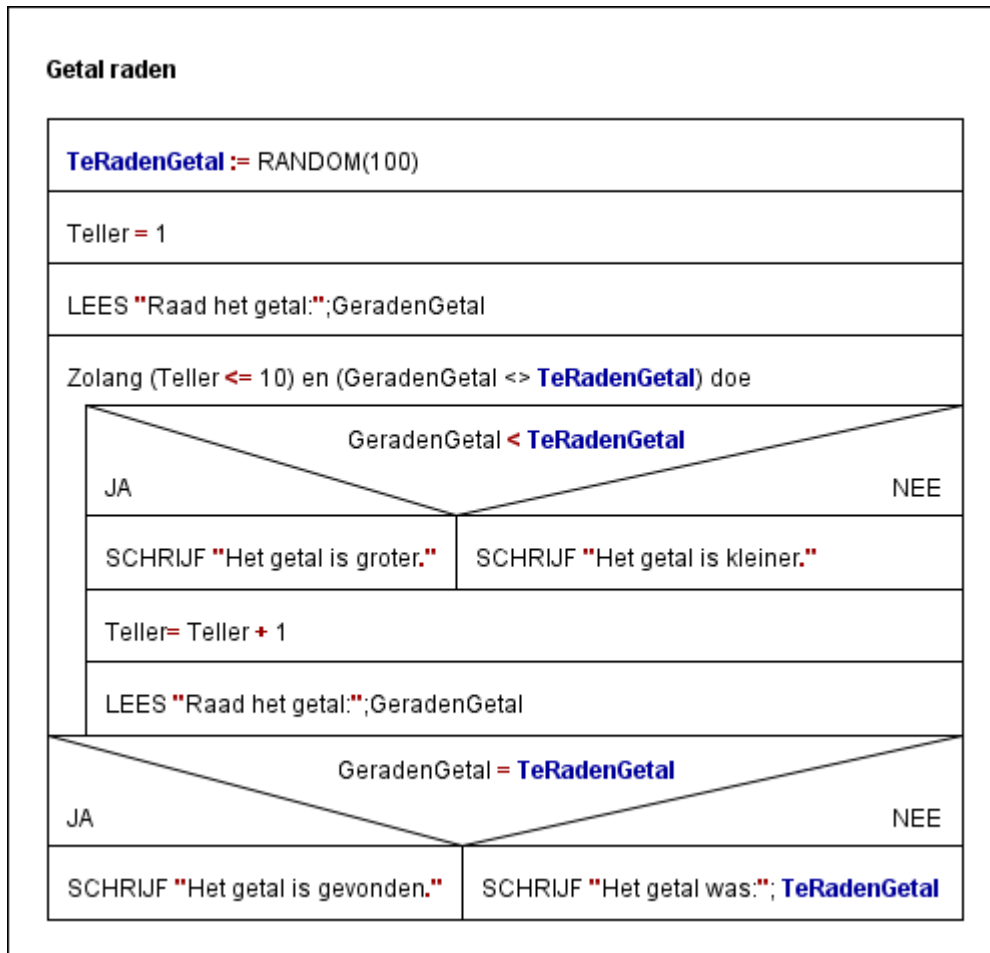
```
Voer een getal in waar tussen geraden moet worden: 10
Raad het getal: 5
Het getal is groter.
Raad het getal: 7
Getal is geraden!
Press any key to continue...
-
```

In het voorbeeld is het algoritme iets uitgebreider daar er eerst door het programma gevraagd wordt tussen nul en wel getal maximaal een getal gekozen moet worden door het programma.



Van Algoritme naar PSD

Uitwerking PSD:



Voor het genereren van een willekeurig getal gebruik je in de PSD structuur de opdracht **RANDOM(WAARDE)**. Deze functie genereert een getal tussen 0 en de waarde die is ingevuld. Elke programmeertaal heeft er zo zijn eigen methode voor. In een latere les zullen we zien hoe dat gebeurt in de programmeertaal Small Basic waar mee verder gewerkt gaat worden.

De opdracht `Teller = 1` wordt gebruikt om een variabele Teller te maken in het werkgeheugen van de computer en deze van de waarde één te voorzien.

De opdracht `LEES "Raad het getal.";GeradenGetal` wordt gebruikt om de tekst "Raad het getal:" op het scherm te tonen en de gebruiker van het programma te vragen een getal in te voeren waarvan de waarde dan in de variabele GeradenGetal wordt opgeslagen.

De opdracht `Zolang (Teller <= 10) en (GeradenGetal <> TeRadenGetal) Doe` geeft het begin aan van de iteratie-lus waarin de criteria (voorwaarden) staat om



Van Algoritme naar PSD

de opdrachten in de lus te blijven uitvoeren.

De criteria bestaat uit twee delen. In het eerste deel wordt er gecontroleerd of de waarde van de variabelen Teller kleiner of gelijk (\leq) aan 10 is. In het tweede deel wordt gecontroleerd of de waarden in de variabelen **GeradenGetal** en **TeRadenGetal** ongelijk zijn (\neq).

Wanneer beide voorwaarden waar zijn dan wordt er verder gegaan met de opdrachten in de lus. Echter als één van de voorwaarden niet voldoet dan worden alle opdrachten in de lus overgeslagen en gaat het programma verder na de lus. In dit geval zou dat de tweede selectie structuur zijn in de PSD.

De selectie opdracht `GeradenGetal < TeRadenGetal` bepaald of het te raden getal groter of kleiner is dan het geraden getal. Afhankelijk van deze voorwaarde wordt of de JA- of de NEE-actie uitgevoerd.

De opdrachten `SCHRIJF "Het getal is groter."` en `SCHRIJF "Het getal is kleiner."` tonen de teksten op het beeldscherm zodat de gebruiker van het programma deze kan lezen.

De opdracht `Teller = Teller + 1` zorgt ervoor dat de waarde in de variabelen Teller steeds met 1 opgehoogd wordt. Zo wordt het aantal beurten van de speler bijgehouden.

Daar het getal niet geraden is wordt er op nieuw aan de speler gevraagd om een getal in te voeren.

Dit is een herhaling van de opdracht `LEES "Raad het getal: ";GeradenGetal` als die voor de lus in zijn gegaan wordt.

Wanneer de lus verlaten wordt volgt er weer een selectie-structuur. Deze wordt gebruikt om te controleren of de speler het getal geraden heeft of niet en de juiste tekst aan de speler te laten zien.

Fouten zoeken

Een PSD moet vertaald kunnen worden naar een goed werkend programma in elke programmeertaal. Om te begrijpen hoe een PSD werkt of om deze te controleren is het noodzakelijk om de opdrachten één voor één na te lopen.

De opdrachten worden stap voor stap op papier uitgevoerd. Dit heet een **ooggetuigenverslag**.



Van Algoritme naar PSD

Een ooggetuigenverslag is een manier om eventuele fouten in een programma op te sporen. Dit wordt ook wel **debuggen** genoemd. In moderne professionele programmeeromgevingen zijn er speciale hulpmiddelen die helpen bij het opsporen van programmafouten. Een programmafout wordt een bug genoemd.

Hieronder volgt een voorbeeld van een ooggetuigen:

Voorbeeld PSD

Ooggetuigenverslag

Ooggetuigenverslag		regel	x	y	$x > -3y$	Commentaar
{1} X := 5		1	5			5 toekennen aan x
{2} y := -1		2	5	-1		1 toekennen aan y
{3} ZOLANG $x > -3y$		3	5	-1	Waar	5 is groter dan $(-3 * -1 =) 3$
{4} $x := x + 2y$		4	3	-1		x krijgt waarde $x + 2y$, dus 3
{5} $y := -2y$		5	3	2		y krijgt waarde $-2y$, dus 2. Terug naar regel 3
{6} SCHRIJF "x="; x		3	3	2	Waar	3 is groter dan $(-3 * 2 =) -6$
{7} SCHRIJF "y="; y		4	7	2		x wordt $x + 2y$, dus 7
		6	7	-4		y wordt $-2y$, dus -4. Terug naar regel 3
		3	7	-4	Niet waar	7 is niet groter dan $(-3 * -4 =) 12$, dus naar regel 6
		6	7	-4		Afgedrukt wordt: x=7
		7	7	-4		Afgedrukt wordt: y=-4

Doormiddel van een ooggetuigenverslag kan precies gevolgd worden welke waarden de variabelen hebben tijdens de uitvoering van het programma door de computer.