

WISKUNDE

Probleemoplossend denken en vraagstukken

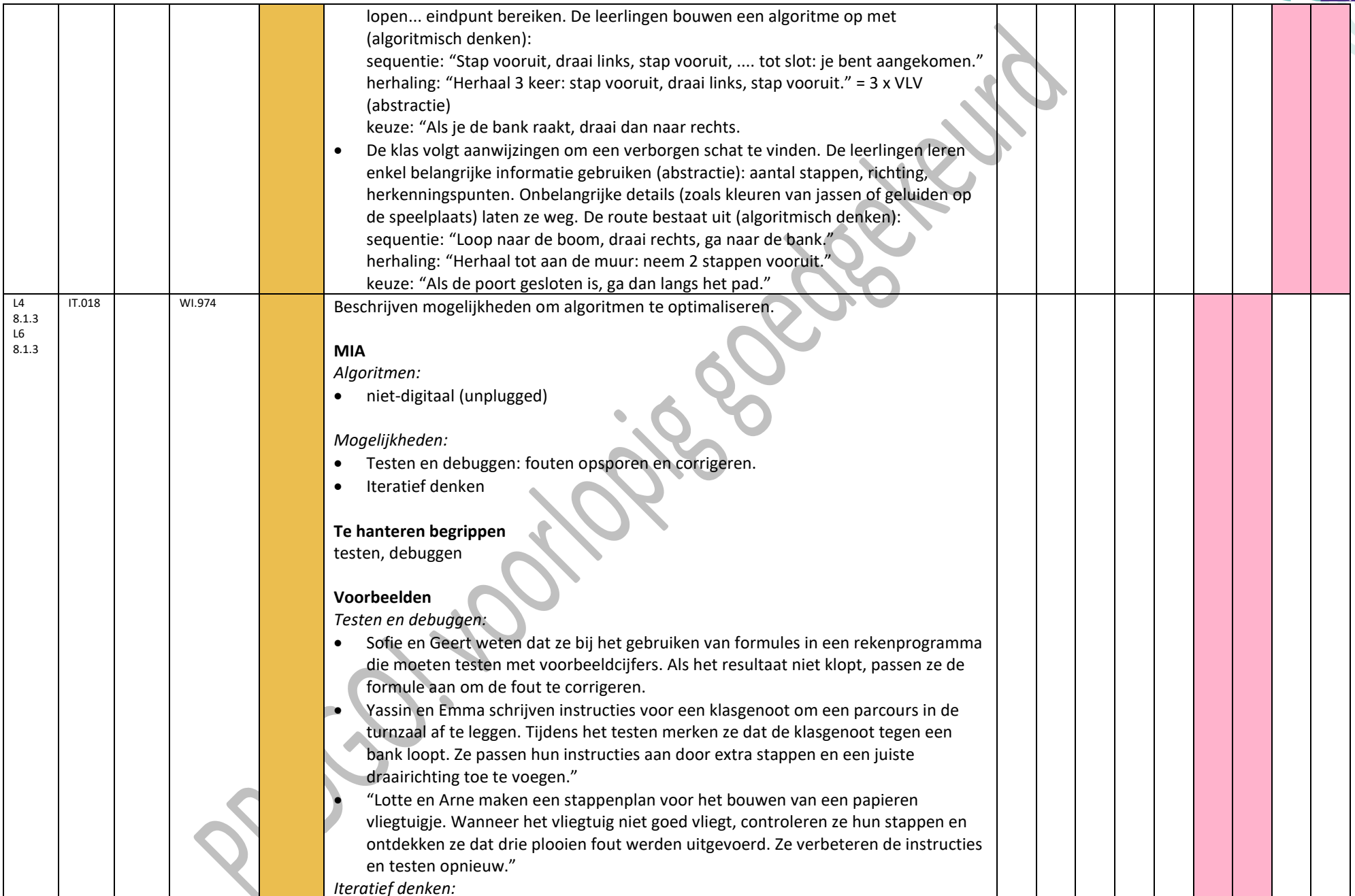
KO De kleuters kunnen 2.6.1 problemen in spel- en leersituaties oplossen gebruikmakend van wiskundige elementen, door: • op zoek te gaan naar manieren om een probleem op te lossen, ook als er meerdere oplossingen zijn; • concreet materiaal te gebruiken; • hun ideeën te delen met anderen.	L4 De leerlingen kennen 8.1.2 het volgende begrip: het algoritme. De leerlingen kunnen 2.2.11 betekenisvolle situaties omzetten naar bewerkingen met natuurlijke getallen en decimale getallen. 2.6.1 wiskundige problemen oplossen met minstens 1 bewerking/handeling, door: • denkstappen bij wiskundige problemen toe te passen; < bv: * het probleem in eigen woorden vertellen; * het gegeven en het gevraagde onderscheiden; * herkennen of een routineuze methode (zie vraagstukken) of heuristische methode nodig is; * overbodige gegevens negeren; * fouten aangrijpen als aanknopng voor wat nodig is voor een correcte oplossing; * controleren of antwoorden mogelijk zijn. > • heuristieken toe te passen. < bv: * concreet materiaal gebruiken; * een tekening of schets maken; * een schema of tabel maken; * een systematiek herkennen en voortzetten (patroonherkenning); * gissen en missen. > 8.1.3 onder begeleiding de principes van computationeel denken toepassen om een lineair algoritme te ontwerpen,	L6 De leerling kan 2.2.4 betekenisvolle situaties omzetten naar bewerkingen met breuken en procenten. 2.6.1 wiskundige problemen oplossen met meerdere bewerkingen/handelingen, door: • denkstappen bij wiskundige problemen toe te passen; < bv: * de denkstappen van het 4^{de} jaar toepassen; * een model om problemen op te lossen hanteren zoals het model van Polya dat de volgende stappen onderscheidt: (1) een probleem begrijpen, (2) een plan maken, (3) een plan uitvoeren, (4) reflectie. > • heuristieken toe te passen. < bv: * de heuristieken van het 4^{de} jaar toepassen; * naar analogie te werken met soortgelijke, eerder opgeloste problemen; * een probleem opsplitsen in deelproblemen (decompositie); * een gegeven voorlopig buiten beschouwing laten; * moeilijke gegevens tijdelijk vervangen door eenvoudige; * een algoritme ontwerpen om het probleem op te lossen; * abstraheren. > De leerlingen kunnen 8.1.3 onder begeleiding de principes van computationeel denken toepassen om een algoritme te ontwerpen, te
---	---	---

	testen en debuggen om tot een werkende oplossing te komen: - de principes van computationeel denken: abstractie, decompositie, patroonherkenning, algoritme; < * abstractie: het focussen op wat belangrijk is en het weglaten van overbodige details * decompositie: het opdelen van een groot probleem in kleinere, hanteerbare delen * patroonherkenning: het zien van gelijkenissen en herhalingen in gegevens of opdrachten, bv. in reeksen, ritmes, bewegingen en programmeeropdrachten * algoritme: een expliciete reeks eenduidige instructies die stapsgewijs moeten worden uitgevoerd, waarbij zowel de instructies als hun volgorde essentieel zijn om het gewenste resultaat te bereiken om zowel een niet-digitaal als digitaal probleem op te lossen > - opbouw van algoritme: sequentie;	testen en debuggen om tot een werkende oplossing te komen: opbouw van algoritme: combinatie van sequentie, herhaling, keuze;
--	--	---

Probleemoplossend denken

Computationeel denken

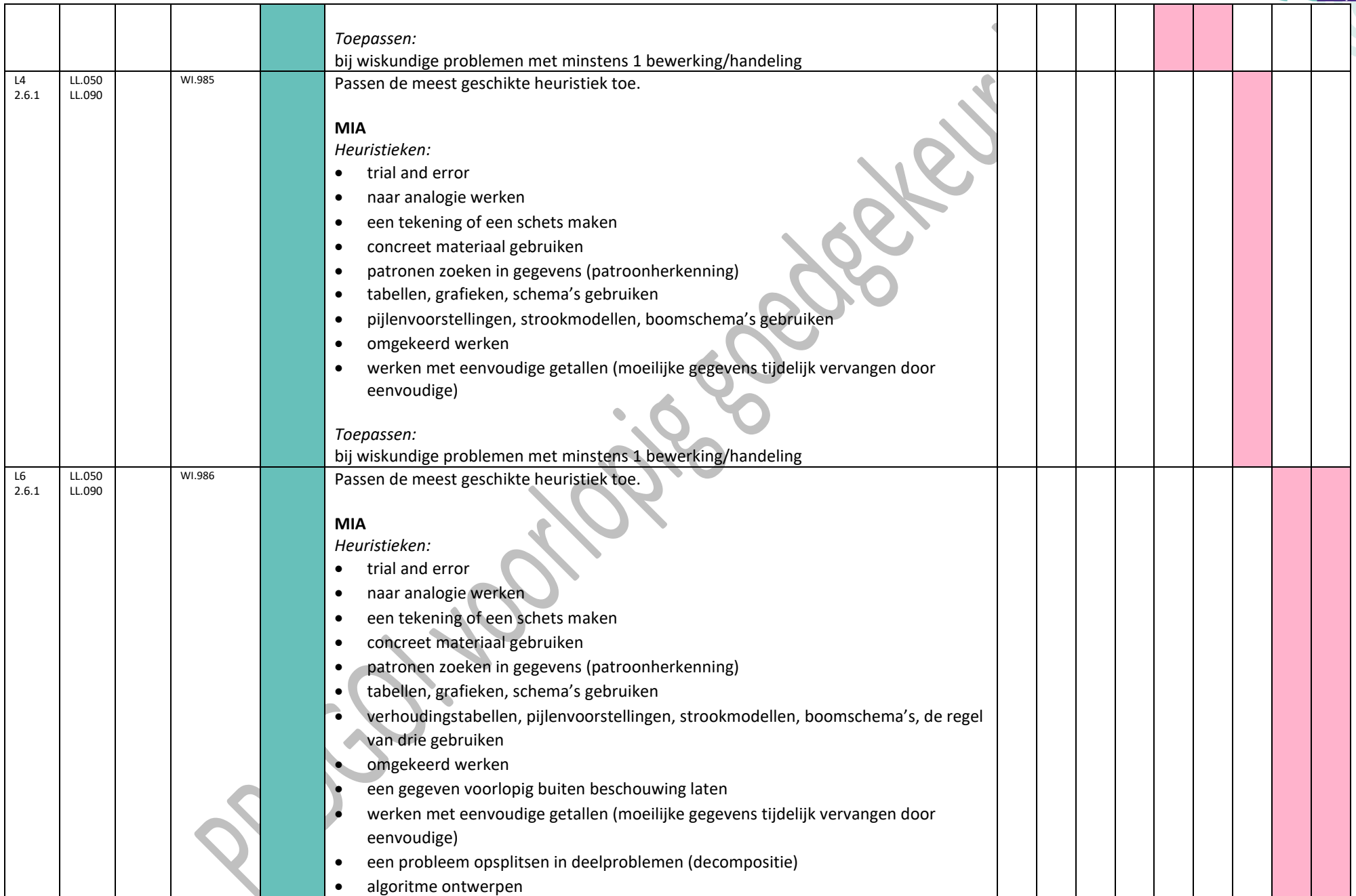
MD/GO!			Nr.	E/B/G	Leerlijn	2,5-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12
L4 8.1.3	IT.015		WI.971		Herkennen hoe een eenvoudig lineair algoritme opgebouwd is volgens de principes van computationeel denken. MIA <i>Algoritme:</i> - niet-digitaal (unplugged) <i>Principes van computationeel denken:</i> - abstractie: Het focussen op wat echt belangrijk is en het weglaten van details die er niet toe doen. - decompositie: Het verdelen van een eenvoudige en concrete taak of probleem in kleinere, haalbare stukjes. - patroonherkenning: Het zien van herhalingen of gelijkenissen. - algoritmisch denken: Stap voor stap een oplossing opbouwen Voorbeelden <i>Niet-digitaal:</i>									



					lets steeds opnieuw proberen en verbeteren, in kleine stappen naar een oplossing toewerken, een proces meerdere keren doorlopen om het te verbeteren, een gestructureerde aanpak volgen waarbij je steeds aanpassingen maakt												
L4 8.1.3	IT.019		WI.975		Ontwerpen een eenvoudig lineair algoritme volgens de principes van computationeel denken. MIA <i>Ontwerpen:</i> • onder begeleiding • niet-digitaal (unplugged) <i>Principes van computationeel denken:</i> • abstractie • decompositie: eenvoudig en concreet • patroonherkenning • algoritmisch denken Voorbeeld Glen maakt een stappenplan om een geblinddoekte leerling door een parcours te leiden. Hij bepaalt eerst de verschillende stappen (decompositie), focust daarbij enkel op de essentie, zorgt voor een weergave o.a. aan de hand van kaartjes met pijlen (abstractie) en schrijft alle stappen op in de juiste volgorde (algoritmisch denken).												
L4 8.1.2 L4 8.1.3	IT.020		WI.976		Ontwerpen een lineair algoritme volgens de principes van computationeel denken. MIA <i>Ontwerpen:</i> • onder begeleiding • niet-digitaal (unplugged) <i>Principes van computationeel denken:</i> • abstractie • decompositie: groter en meer complex • patroonherkenning • algoritmisch denken: met aandacht voor sequentie Voorbeeld Sandra maakt een instructieblad om een andere leerling een origami-dier te laten vouwen. Ze verdeelt de opdracht in kleine, haalbare stapjes (decompositie), zoals: papier dubbelvouwen, een hoek omplooien en de oren vormen. Bepaalde stappen keren terug: papier dubbelvouwen en punt naar boven plooiën (patroonherkenning). Ze												

					noteert elke stap met een symbool en de terugkerende stappen met bv. 3 x ... (abstractie). Daarna zet ze alle stappen in de juiste volgorde op haar instructieblad (algoritmisch denken: sequentie)												
L6 8.1.3	IT.021		WI.977		Ontwerpen een algoritme volgens de principes van computationeel denken. MIA <i>Ontwerpen:</i> • onder begeleiding • niet-digitaal (unplugged) <i>Principes van computationeel denken:</i> • abstractie • decompositie: groter en meer complex • patroonherkenning • algoritmisch denken: met aandacht voor de combinatie van sequentie, herhaling en keuze Voorbeeld Dilan bedenkt een korte dans die andere kinderen exact moeten kunnen uitvoeren. Hij splitst eerst de dans op in kleine bewegingen: klappen, draaien, springen en stappen (decompositie). Hij merkt dat sommige bewegingen telkens opnieuw voorkomen, zoals twee keer klappen of drie stappen naar voren (patroonherkenning). Hij kiest eenvoudige tekens en woorden, bijvoorbeeld pijlen voor richtingen en symbolen voor bewegingen (abstractie). Daarna maakt hij een stappenplan (algoritmisch denken): "Zet 2 stappen vooruit. Klap in de handen. Herhaal 3 keer: draai een halve cirkel en spring. Als de muziek stopt, blijf dan stilstaan..."												
L4 8.1.3 L6 8.1.3	IT.022		WI.978		Sturen een algoritme bij. MIA <i>Bijsturen:</i> • testen en debuggen • niet-digitaal (unplugged) <i>Algoritme:</i> • gekregen • Zelfgemaakt Voorbeelden												

8



[illegible]

KO	L4 De leerlingen kunnen 2.6.2 met natuurlijke getallen tot en met 1 000 (of grotere getallen met eindnullen), decimale getallen (tot op 1 honderdste) en breuken (noemer tot en met 20 of breuken die te vereenvoudigen zijn tot deze breuken): vraagstukken oplossen over: • de bewerkingen met (enkelvoudige vraagstukken): ○ natuurlijke getallen voor optellen, aftrekken, vermenigvuldigen en delen; ○ decimale getallen en breuken voor optellen en aftrekken. • verhoudingen met behulp van een verhoudingstabel waarbij: ○ vaste verhoudingen worden vastgesteld; < bv. verhouding blauwe en rode kralen in kralenkettingen > ○ gelijkwaardige verhoudingen bepaald worden; ○ het ontbrekende verhoudingsgetal berekend wordt;	L6 De leerling kan 2.6.2 met natuurlijke getallen tot en met 10 000 (of grotere getallen met eindnullen), decimale getallen (tot op 1 duizendste) en breuken (noemer tot en met 100): (samengestelde) vraagstukken oplossen over: • breuken en procenten met: ○ een breuk als operator of deel/geheel; ○ groeipercentages; < bv. een bevolkingstoename > ○ absolute en relatieve vergelijkingen (met verhoudingstabel, een breuk of een procent). < bv. in de Vlinderschool (100 lln.) fietst 48%, in de Parkschool (2000 lln.) een vijfde. Relatief meer fietsers in de Vlinderschool, absoluut meer in de Parkschool. > • delers, veelvouden en restbepaling; • de bewerkingen (+, -, x, :) met (samengestelde vraagstukken): ○ natuurlijke getallen; ○ decimale getallen, breuken, procenten.
------------------	---	--

		○ recht evenredige grootheden een rol spelen. < bv. aantal-prijs, massa-prijs, een recept van 2 personen omzetten naar 6 personen > • één grootheid: lengte, oppervlakte, inhoud, massa, tijd, geld, temperatuur.	• ongelijke verdeling waarbij: ○ de som en het verschil gegeven zijn; ○ de som en de verhouding van de delen gegeven zijn. • mengsels; • verhoudingen met behulp van een verhoudingstabel (waaronder de regel-van-drie) waarbij: ○ gelijkwaardige verhoudingen bepaald worden; ○ het ontbrekende verhoudingsgetal berekend wordt; ○ recht, omgekeerd en niet-evenredige grootheden een rol spelen. < bv. aantal-prijs, afstand-prijs, afstand-tijd, massa-prijs, aantal dieren-hoeveelheid voedsel, debiet-tijd, tijd-snelheid > • één grootheid: lengte, oppervlakte, inhoud, volume, massa, tijd, geld, temperatuur, hoekgrootte; • relaties tussen grootheden (zonder en met procenten): ○ prijsberekeningen: winst/verlies, korting, prijsstijging, enkelvoudige interest; ○ bruto/tarra/netto ; ○ afstand, snelheid en tijd; ○ schaal, lengte op schaal en werkelijke lengte. • meetkundige figuren waarbij de eigenschappen gebruikt worden; • statistiek met tabellen, grafieken, diagrammen, gemiddelde en mediaan.
--	--	--	--

Wiskundige vraagstukken

MD/GO!			Nr.	E/B/G	Leerlijn	2,5-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12
	LL.090		WI.988		Zetten een rekenverhaal en een rekenzin naar elkaar om. Voorbeeld <i>Rekenverhaal:</i> Nel wandelt in het park en ziet in de ene vijver 2 eenden en in de andere 3 eenden. Hoeveel eenden ziet Nel? <i>Rekenzin:</i>									

					2 en 3 zijn samen 5.										
L4 2.6.2 L6 2.6.2	LL.090		WI.989		(De)mathematiseren een wiskundig vraagstuk. Voorbeeld <i>Wiskundig vraagstuk:</i> Ik koop in de winkel twee flessen melk (prijs = €2,15 per fles) en een halve kilo peren (prijs = €3,2 per kilo). Ik heb maar €5 op zak. Heb ik voldoende? <i>Mathematiseren:</i> 2 x 2,15 = 4,3 1/2 x 3,2 = 1,6 4,3 + 1,6 = 5,9 5,9 > 5 <i>Demathematiseren:</i> Ik moet €5,9 betalen. Dat is meer dan €5. Ik kom dus 90 eurocent te kort om mijn aankopen te kunnen betalen										
L4 2.6.2	LL.050 LL.090		WI.990		Lossen enkelvoudige vraagstukken over gekende leerinhouden op. MIA <i>Vraagstukken:</i> getalbereik ≤20 • met optellen en aftrekken • over één grootheid (natuurlijke maten lengte, inhoud, massa, oppervlakte, tijd en geld) types: • vraagstukken op basis van parate kennis • deel-geheelvraagstukken binnen de natuurlijke getallen (additief) • veranderingsvraagstukken binnen de natuurlijke getallen (additief, het resultaat is onbekend) • vergelijkingsvraagstukken binnen de natuurlijke getallen (vergeleken hoeveelheid is onbekend) Voorbeelden <i>Veranderingsvraagstukken binnen de natuurlijke getallen (additief, het resultaat is onbekend):</i> Hugo heeft 4 knikkers. Hij krijgt er 2 bij. Hoeveel heeft hij er nu? <i>Vergelijkingsvraagstukken binnen de natuurlijke getallen (additief, vergeleken hoeveelheid is onbekend):</i> Caro heeft 4 snoepjes. Jan heeft 2 snoepjes meer dan Caro. Hoeveel snoepjes heeft Jan?										
L4 2.6.2	LL.050 LL.090		WI.991		Lossen enkelvoudige vraagstukken over gekende leerinhouden op.										

- over één grootte (lengte, inhoud, massa, oppervlakte, volume, tijd, geld, temperatuur en hoekgrootte)
- met breuken en procenten
 - een breuk als operator/deel-geheel

