

Inleiding Programmeren - Opgave 3: Complexe Breuken

Criterion	Expert (10)	Bedreven (8-9)	Competent (6-7)	Gevorderd (4-5)	Beginner (1-3)
<i>Opbouw</i>	De code is op een logische manier en goed objectgeoriënteerd opgebouwd: de gemaakte methoden zijn logisch en DRY. Er is gebruik gemaakt van constructor chaining, de GGD wordt slechts in 1 constructor aangeroepen en de deel methode maakt gebruik van het vermenigvuldigen met de omgekeerde. 3.5 pt	De code is op een redelijk logische manier en redelijk objectgeoriënteerd opgebouwd, maar de code is niet helemaal DRY. Bijvoorbeeld: Constructor chaining is niet aanwezig, of de GGD wordt op meerdere plekken aangeroepen, of de deel functie is niet DRY geïmplementeerd, of de toString in ComplexeGetallen maakt geen gebruik van de toString in Breuk. 3.0 pt	De code is enigszins logisch en enigszins objectgeoriënteerd opgebouwd maar er zijn meerdere fouten gemaakt in de opbouw. Op maximaal 2 plekken is de code niet logisch opgebouwd (zie voorbeelden bij Bedreven). 2.0 pt	De benodigde methoden zijn wel aanwezig maar deze zijn zeer onlogisch opgebouwd. Op minstens 3 plekken is de code niet logisch opgebouwd (zie voorbeelden bij Bedreven). 1.3 pt	Alle eerdergenoemde fouten zijn gemaakt. De tests staan binnen de Breuk/ComplexGetal klassen. 0.0 pt
<i>Uitvoering</i>	Alle functionaliteit werkt volledig naar behoren zoals de voorbeelduitvoer en is zo eenvoudig mogelijk geïmplementeerd. Daarnaast is een goed testscript geschreven dat modulaair inzetbaar is en alle gemaakte code test. Compilatie verloopt zonder meldingen. 3.0 pt	Beide klassen werken in principe correct, echter zitten er kleine foutjes in de uitvoer doordat of de breuk niet goed vereenvoudigd wordt, of doordat de toString/equals niet juist werkt. Het geschreven testscript test niet de gehele code grondig genoeg. 2.4 pt	Beide klassen werken in principe correct, al zijn er een aantal implementatiefouten die invloed hebben op de uitvoer van het programma. Er treden compilatiewaarschuwingen op. 1.8 pt	Er zitten flinke fouten in de implementatie of het programma compileert niet direct. Een klasse werkt niet of het testscript ontbreekt. 1.2 pt	De code compileert niet zonder uitgebreide aanpassingen, of werkt grotendeels niet. 0.0 pt
<i>Commentaar</i>	Commentaar is geschreven in grammaticaal correcte volzinnen, beginnend met een hoofdletter en eindigend met een punt, en bevindt zich bovenaan ieder bestand (en dus iedere klasse), iedere methode en code dat niet in een oogopslag te begrijpen valt. De kern van de functionaliteit wordt kort en bondig beschreven. Variabelnamen zijn kort en bondig, en beschrijven het doel van de variabele. 2.0 pt	Bovenaan ieder bestand (en dus klasse) en iedere methode staat commentaar in grammaticaal correcte volzinnen. Het commentaar raakt niet altijd de kern van wat het beschrijft, maar is verder wel volledig. 1.6 pt	Bovenaan bijna ieder bestand (en dus klasse), bijna iedere methode en bijna elk lastig stuk code in een methode staat commentaar. Het commentaar raakt niet de kern, van wat het probeert te beschrijven. Het aantal taalfouten is op een hand te tellen. Variabelnamen beschrijven vaak niet het doel van de variabelen. 1.2 pt	Commentaar ontbreekt op essentiële plekken, bevat taalfouten, en beschrijft niet datgene dat het moet beschrijven. Hier en daar staat commentaar met // aangegeven. 0.8 pt	Commentaar is niet of nauwelijks aanwezig. 0.0 pt
<i>Stijl</i>	De gebruikte stijl is netjes en volledig consistent: er zijn geen tabs/spaties gemixt, geen trailing-spaces, accolades staan altijd op dezelfde plek, spaties staan op de juiste plek. Verticale ruimte is goed, regels zijn niet langer dan 120 karakters en namen en commentaar zijn consistent in één taal, Nederlands of Engels. 1.5 pt	De gebruikte stijl is zo goed als consistent. Het aantal keer dat de stijl inconsistent is, is op twee handen te tellen. Meerdere regels zijn langer dan 120 karakters. 1.3 pt	Tabs en spaties staan gemixt, en sommige regels bevatten trailing-spaces. De code is juist geïndenteerd, en de accolades staan, meestal, consistent op dezelfde plek. Sommige regels bevatten meer dan 120 karakters. 1.0 pt	De code is geïndenteerd, maar hier zijn fouten in gemaakt. Spaties zijn niet of nauwelijks aanwezig in en rond expressies en accolades zijn niet consistent geplaatst. 0.5 pt	De indentatie klopt niet, tabs en spaties staan door elkaar, (onderdelen van) expressies zijn niet gescheiden met spaties, regels eindigen met trailing spaces en het gebruik van accolades is inconsistent. 0.0 pt
<i>Compleetheid</i>	De opgave is volledig geïmplementeerd: zowel de Breuk, als ComplexGetal klasse met Breuken. De gespecificeerde naamgeving is aangehouden. 0.0 pt	De opgave is bijna volledig geïmplementeerd: er missen 1 of 2 methoden maar verder zijn zowel de Breuk klasse als de ComplexGetal klasse met Breuken aanwezig. Of de gespecificeerde naamgeving is (deels) genegeerd. -1.5 pt	De Breuk klasse is (bijna) volledig aanwezig, maar de ComplexGetal is met doubles geïmplementeerd. -3.0 pt	De Breuk klasse of de ComplexGetal klasse met doubles is aanwezig, maar niet beiden. -6.0 pt	Er is geen noemenswaardige implementatie van één van beide klassen. -9.0 pt