

Examenproject Objectgericht programmeren

2021-2022

Deze tekst beschrijft de opgave voor het examenproject van de cursus Objectgericht Programmeren. Dit project geldt als basis voor het examen van deze cursus. Het project wordt uitgewerkt in dezelfde groepjes van 3 studenten. Indien studenten weggefallen zijn in een groep wordt de opgave ingeperkt. Meer hierover aan het einde van de opgave.

Het project handelt over het bouwen van een deel van de kern van een role-playing game (RPG) in een Middeleeuwse setting omtrent Alchemie. Het spel is slechts een weinig gebaseerd op bestaande spellen. Sommige aspecten uit de opgave komen dan ook niet overeen met bestaande spellen. Ze worden ingevoerd om de opgave zo gevarieerd mogelijk te maken. Je oplossing moet de regels volgen zoals ze in deze opgave beschreven staan. Alle grafische aspecten en alle aspecten in verband met gebruikersinteractie worden buiten beschouwing gelaten. Het project beperkt zich tot het uitwerken van die elementen, die te maken hebben met de alchemische ingrediënten die in het spel voorkomen. In de opgave worden een aantal vereisten gesteld omtrent hoe deze elementen moeten worden ondersteund in het beoogde systeem. In de eerste plaats worden een reeks functionele vereisten gesteld. Dit zijn eerder vage beschrijvingen van functionaliteiten die door het systeem moeten worden ondersteund. **We verwachten dat je zelf de nodige basisfunctionaliteiten invoert zoals getters, setters en checkers voor karakteristieken.** Verder zijn er een aantal niet-functionele vereisten, die zich richten op de kwaliteit van het beoogde systeem. Zo worden op diverse punten vereisten gesteld op het vlak van aanpasbaarheid en herbruikbaarheid.

1 Opgave

1.1 **Alchemisch Ingrediënt** [AlchemicIngredient]

Om te beginnen wordt de klasse AlchemicIngredient uitgewerkt. Een alchemisch ingrediënt is een bepaalde **hoeveelheid** van een bepaalde **substantie**, die kan worden gebruikt om samen met andere ingrediënten toverdrankjes en dergelijke te maken. Ingrediënten hebben een aantal **basissenmerken** die moeten worden uitgewerkt.

1.1.1 **Naam** [Name]

Elk alchemisch ingrediënt heeft een naam. Zo hebben we b.v. Water, Milk en Salt, maar ook Lizard's Tale, Rat's Eye Fluid en Red Mushroom Gas. De naam van een ingrediënt bestaat uit **één of meerdere woorden gescheiden door spaties**. Woorden mogen **geen cijfers, speciale tekens of leestekens** bevatten, buiten het **afkappingsteken en ronde haakjes**. Elk woord in de naam van een ingrediënt bestaat uit **minstens twee letters**, tenzij het gaat om een naam met slechts één woord; in dat geval moet dit woord uit **minstens drie letters** bestaan. Elk nieuw woord in de naam begint met een hoofdletter of toegelaten speciaal teken, en de andere letters in elk woord zijn kleine letters. Er zijn echter een

paar uitzonderingswoorden die deze hoofdletter niet krijgen: dit zijn ‘mixed’ en ‘with’. Het gebruik van deze woorden wordt duidelijk wanneer we het over het mengen van ingrediënten zullen hebben. Buiten dat gebruik zijn deze woorden niet toegelaten in de naam van een element. Op termijn is het mogelijk dat ook andere tekens worden toegelaten in de spelling van namen (bijvoorbeeld @, [en]).

De naam tot zover beschreven zullen we vanaf nu de ‘eenvoudige naam’ van het ingrediënt noemen. Die eenvoudige naam kunnen we ook uitbreiden tot de volledige naam van een ingrediënt door er bepaalde voor- en achtervoegsels aan toe te voegen, afhankelijk van hun andere eigenschappen (zoals bvb. ‘Heated’ en ‘Cooled’, zie verder). Er moeten inspectoren worden voorzien om zowel de volledige naam (met alle voor- en achtervoegsels) als de eenvoudige naam (zonder deze bijvoegsels) op te vragen. Geen enkel van de ingevoerde bijvoegsels mag worden gebruikt als woord in de eenvoudige naam van een ingrediënt.

Wanneer ingrediënten gemengd worden met elkaar, moet het ook mogelijk zijn om aan dat mengsel een speciale naam te geven. Zo kunnen we b.v. ‘Beer mixed with Coke’ vanaf nu ‘Mazout’ noemen. Deze speciale naam moet aan alle voorwaarden voldoen waaraan de naam van een niet-gemengd ingrediënt moet voldoen. (Merk op dat namen van niet-gemengde ingrediënten op geen enkele manier de woorden ‘mixed’ of ‘with’ mogen bevatten.) De volledige naam van een mengsel is nu de speciale naam gevolgd door, tussen haakjes, de eenvoudige naam aangevuld met eventuele bijvoegsels. Wanneer echter geen speciale naam werd toegekend aan een mengsel, dan wordt de volledige naam op dezelfde manier gevormd als bij niet-gemengde ingrediënten. Bij mengsels moeten dus drie namen opvraagbaar zijn: de eenvoudige, de volledige en de speciale naam.

De naam van een ingrediënt wordt defensief geprogrammeerd.

1.1.2 Soorten Ingrediënten [IngredientType]

De substantie van een ingrediënt wordt ook wel de soort genoemd. Het gaat om het element op zich, en alle vaste eigenschappen van dergelijk element. Er kunnen dus twee alchemische ingrediënten bestaan van dezelfde soort, die bijvoorbeeld verschillen in hoeveelheid of temperatuur. Elke hoeveelheid alchemisch ingrediënt bestaat uit exact één soort ingrediënt. Wanneer de opgave over een ingrediënttype spreekt, dan gaat het over de soort, wanneer de term ingrediënt wordt gebruikt, dan gaat het om een bepaalde hoeveelheid van een bepaalde soort. Een aantal eigenschappen van ingrediënten behoort tot de klasse IngredientType. Alles wat voor een ingrediënt vast staat, wordt bepaald door de soort waartoe het behoort. De eigenschappen van een ingrediënt die kunnen wijzigen, zoals b.v. de hoeveelheid of de temperatuur, behoren tot de klasse van ingrediënten. Verderop in de opgave volgen nog meer aanwijzingen voor wat in welke klasse kan/moet komen. De soort van een ingrediënt moet totaal worden uitgewerkt.

1.1.3 Toestand [State]

Ingrediënten en Ingrediënttypes krijgen een bepaalde toestand waarin ze zullen verkeren. De toestand van het ingrediënttype beschrijft de toestand in dewelke het ingrediënt onder normale omstandigheden voorkomt. Ingrediënten kunnen

enkel op **magische wijze van toestand veranderen**. Dit kan in een laboratorium (zie verder). Er zijn twee toestanden waarin ingrediënten kunnen voorkomen:

- **Liquid**. Dit zijn vloeibare ingrediënten, die kunnen worden bewaard in flessen en dergelijke.
- **Powder**. Dit zijn poedervormige ingrediënten, die kunnen worden bewaard in dozen, kisten, enz.

1.1.4 Hoeveelheid [Quantity]

De hoeveelheid van het **ingrediënt** wordt uitgedrukt met een **geheel getal**. Alchemisten in de Middeleeuwen hadden nog geen notie van het SI-stelsel, vandaar dat we niet met gram en kilogram werken, maar met een hoeveelheid lepels (spoons). De aanduidingen voor hoeveelheden zijn: drop, spoon, (8 drops), vial (5 spoons), bottle (3 vials), jug (7 bottles), barrel (12 jugs) en storeroom (5 barrels). De hoeveelheid van een ingrediënt kan uiteraard nooit negatief zijn. De hoeveelheid van een ingrediënt kan niet meer veranderen na het ontstaan ervan. **Indien er dus iets aan een ingrediënt verandert dat invloed heeft op de hoeveelheid, dan beschouwen we het resultaat als een nieuw ingrediënt.**

Het uitdrukken van de hoeveelheid van een ingrediënt hangt af van de toestand waarin het ingrediënt verkeert. Ook de containers waarin ingrediënten kunnen worden bewaard hangen hier vanaf. In een container met een inhoud van één kist kan je b.v. geen lepels water opslaan, aangezien water vloeibaar is. De mogelijke hoeveelheden beschreven hierboven zijn enkel geldig voor Liquids. Voor Powders voeren we nog een aantal andere eenheden van hoeveelheid in, van klein naar groot: pinch, spoon (6 pinches), sachet (7 spoons), box (6 sachets), sack (3 boxes), chest (10 sacks), storeroom (5 chests). De spoons en storerooms stellen voor beide toestanden dezelfde hoeveelheid voor. De andere hoeveelheden zijn voor beide toestanden verschillend.

Hoeveelheden worden **nominaal** geïmplementeerd.

1.1.5 Temperatuur [Temperature]

Ingrediënten kunnen op een bepaalde temperatuur gebracht worden door ze op te warmen of af te koelen. Aangezien onze Middeleeuwse alchemisten nog niets wisten over negatieve getallen (Dat zou een absurditeit zijn die niet in de natuur voorkomt!), wordt de temperatuur voorgesteld door twee waarden: **de warmte (hotness) en de koelte (coldness)**. Beide waarden zijn begrensd van 0 tot 10000. Wanneer een ingrediënt wordt afgekoeld, **verlaagt eerst de warmte tot 0, waarna de koelte begint te stijgen**. Bij verwarming gebeurt het omgekeerde. Het is mogelijk dat op termijn een **andere waarde gebruikt wordt voor de bovengrens** van koelte en warmte. De bovengrens zal wel voor beide karakteristieken steeds **dezelfde** zijn, en zal ook dezelfde zijn voor alle temperaturen. De bovengrens zal daarenboven nooit de waarde `Long.MAX_VALUE` overschrijden. Zowel de koelte als de warmte van een ingrediënt moeten apart kunnen worden opgevraagd, deze mogen echter niet rechtstreeks worden aangepast, hiertoe zijn de methodes **heat** en **cool** voorzien, die beide een positief getal als parameter aanvaarden en de temperatuur op logische wijze aanpassen. Verder moet er een methode **getTemperature** worden voorzien die een array teruggeeft met beide waarden van de temperatuur. We geven daarbij steeds **eerst de koelte weer en op de**

tweede plaats de warmte. Alle alchemische ingrediënttypes hebben verder nog een standaardtemperatuur, waarop ze in de alchemie worden gebruikt. Voor water, dat bevriest op $[0, 0]$, is dit bijvoorbeeld $[0, 20]$. Wanneer een ingrediënt warmer is dan zijn standaardtemperatuur, zetten we het woordje 'Heated' voor de naam. Wanneer het kouder is, gebruiken we 'Cooled'. Op deze manier krijgen ingrediënten dus een voorvoegsel in hun volledige naam. Bij hun ontstaan hebben alle ingrediënten hun standaardtemperatuur. Temperaturen worden *to-taal* geïmplementeerd. Standaardtemperaturen zullen altijd strikt hoger zijn dan $[0, 0]$. Merk op dat een temperatuur die zowel een zekere warmte ($\neq 0$) als een zekere koelte ($\neq 0$) heeft, betekenisloos is.

1.1.6 Constructoren

Er moeten constructoren worden voorzien die ingrediënttypes maken met een gegeven naam en standaardtemperatuur. Er moeten constructoren worden voorzien die toelaten om een bepaalde hoeveelheid ingrediënt aan te maken van een bepaald type bij standaardtemperatuur, en om een bepaalde hoeveelheid standaardingrediënt te maken zonder verdere argumenten, hetgeen 'Water' zal zijn, met standaardtemperatuur $[0, 20]$.

1.2 Ingrediëntencontainer [IngredientContainer]

Een ingrediëntencontainer is, zoals de naam al zegt, een container voor ingrediënten. Ingrediëntencontainers zijn ofwel leeg, ofwel bevatten ze een bepaalde hoeveelheid ingrediënt.

1.2.1 Capaciteit [Capacity]

Een ingrediëntencontainer heeft een bepaalde capaciteit, uitgedrukt als een eenheid van hoeveelheid. Dit is steeds exact één van die hoeveelheden, zo kan er b.v. slechts één fles (als hoeveelheid) in een glazen fles (als container) en slechts één vat [Barrel] in een blauw vat. De containers stellen dus fysieke objecten voor waar meestal ingrediënten in zitten, en hun capaciteit wordt bijgevolg door alchemisten gebruikt om hoeveelheden uit te drukken. Er bestaat b.v. geen fles, met een capaciteit van 2 flessen. Voor de allerkleinste en allergrootste eenheden van hoeveelheid bestaan er overigens ook geen containers, enkel voor de tussenliggende eenheden. Verder is het perfect mogelijk dat een container slechts gedeeltelijk gevuld is (4 'pinches' Salt in een 'Sack'); het kan echter niet dat er meer inzit dan de capaciteit toelaat.

1.3 Alchemische Toestellen [Devices]

Alchemische Toestellen kunnen worden gebruikt om alchemische bewerkingen uit te voeren op de ingrediënten. Ze vormen meteen ook de enige manier om veranderingen aan te brengen aan alchemische ingrediënten. Alle toestellen zullen om die reden de volgende drie methodes aanbieden:

- Een methode om er hoeveelheden ingrediënt in te stoppen. Deze zal een ingrediëntencontainer als argument krijgen, die na uitvoering leeg zal zijn. Omdat alchemisten niet graag hun containers schoonmaken worden deze voorgoed verwijderd na hun gebruik.

- Een methode om het resultaat van de alchemische bewerking terug uit het toestel te halen. Deze zal een nieuwe ingrediëntencontainer aan de gebruiker teruggeven, gevuld met de resulterende hoeveelheid ingrediënt.
- Een methode om de alchemische bewerking van het toestel uit te voeren.

Toestellen kunnen een onbeperkte hoeveelheid ingrediënt verwerken, maar ze kunnen nooit meer dan één container resulterend ingrediënt aan de gebruiker teruggeven. De eventuele rest gaat verloren. Er zijn een aantal verschillende soorten toestellen:

- **De Koelkast [CoolingBox].** De koelkast is een kist waarin ingrediënten kunnen worden afgekoeld. Hoe koel de ingrediënten worden hangt af van de temperatuur van de koelkast, deze kunnen we dan ook instellen alvorens het afkoelen uit te voeren. De koelkast kan slechts 1 ingrediënt bevatten. Het toevoegen van een extra ingrediënt moet resulteren in een Exception. Ingrediënten die reeds koeler zijn dan de ingestelde temperatuur worden niet opgewarmd, maar blijven op hun oorspronkelijke temperatuur na uitvoering van de alchemische bewerking.
- **De Oven [Oven].** In een oven kunnen ingrediënten worden verwarmd. Hoe warm de ingrediënten worden hangt af van de warmte van de oven. De warmte van de oven moeten we dus kunnen instellen alvorens we de verwarming uitvoeren. Een oven werkt echter niet zo precies als een koelkast. De temperatuur die de ingrediënten uiteindelijk zullen krijgen kan namelijk tot 5% afwijken van de ingestelde temperatuur. Ook in de oven kan slechts 1 ingrediënt tegelijk opgewarmd worden. Ingrediënten die reeds warmer zijn dan de ingestelde temperatuur worden niet afgekoeld, maar blijven op hun oorspronkelijke temperatuur na uitvoering van de alchemische bewerking.
- **De Ketel [Kettle].** In een ketel kunnen verschillende ingrediënten gemixt worden. Er kunnen dus meerdere ingrediënten toegevoegd worden. Er wordt geen limiet vooropgesteld. Bij het mixen wordt een nieuw ingrediënt geproduceerd volgens de volgende regels:
 - **Naam.** De eenvoudige naam wordt aangepast om duidelijk te maken dat het om een mengsel gaat. We krijgen daardoor eerst de eenvoudige naam van het alfabetisch eerste ingrediënt, gevolgd door ‘mixed with’, en tenslotte de eenvoudige naam van de andere elementen. Merk op dat ‘mixed with’ enkel voorkomt na de naam van het eerste ingrediënt. De volgende ingrediënttypes worden dan eventueel gescheiden door komma’s of het woordje *and* indien er nog slechts één ingrediënttype volgt. Een voorbeeld van de eenvoudige soortnaam voor zo’n mengsel is dus: ‘Garlic mixed with Imp Gas, Mercurial Acid and Water’. Een uitzondering geldt wanneer meerdere ingrediënten van dezelfde soort (dezelfde eenvoudige naam) worden gemengd: de eenvoudige naam blijft dan ongewijzigd. (Merk op dat we hier dan ook niet over een mengsel spreken.) Zodra er twee verschillende soorten worden gemixt spreken we van een mengsel. Eventuele voor- en achtervoegsels komen zoals gewoonlijk rond deze eenvoudige naam te staan om zo de volledige naam te vormen. Standaard wordt geen

speciale naam aangemaakt, maar die kan later wel ingevoerd worden door de alchemist.

- **Toestand.** Het nieuw ontstane ingrediënt neemt de toestand aan van het basisingrediënt waarvan de standaardtemperatuur het dichtst bij $[0,20]$ ligt. Indien er meerdere basisingrediënten zijn met een standaardtemperatuur even dicht bij $[0,20]$ dan geldt volgende regel: ‘Liquid gaat voor op Powder’.
- **Hoeveelheid.** De hoeveelheid van het resultaat is gelijk aan de totale hoeveelheid van alle gemengde hoeveelheden ingrediënt. Er kan nu echter een afrondingsfout optreden wanneer de gemengde soort een andere toestand heeft dan één of meerdere van de basisingrediënten, én de hoeveelheden uit kleinere eenheden dan spoons bestaan. In dat geval tellen we de fracties spoon van de overgebleven hoeveelheden die een andere toestand hadden op en ronden we het resultaat af naar beneden tot op een geheel aantal spoons. B.v. $4 \text{ pinches} + 5 \text{ pinches} = 4/6 \text{ spoon} + 5/6 \text{ spoon} = 10/6 \text{ spoon}$, wat afgerond wordt tot 1 spoon. De kleine hoeveelheden ingrediënt van soorten die dezelfde toestand hebben als het resulterend ingrediënt worden echter wel apart bij de eindhoeveelheid opgeteld.
De resulterende hoeveelheid is dus niet groter dan de som van de delen, zelfs in de Middeleeuwen konden alchemisten geen materie uit het niets tevoorschijn toveren ...
- **Standaardtemperatuur.** Als nieuwe standaardtemperatuur voor het mengsel wordt de standaardtemperatuur van het ingrediënttype genomen waarvan die het dichtste bij $[0,20]$ ligt. Indien er meerdere kandidaten zijn, dan wordt de de warmste gekozen.
- **Temperatuur.** Het mengsel heeft als temperatuur het gewogen gemiddelde van de temperaturen van beide gemengde ingrediënten. Het gewicht wordt bepaald door de hoeveelheid van beide elementen. Wanneer we b.v. 25 lepels water van $[0,100]$ mengen met 75 lepels water van $[100,0]$, krijgen we $100 \text{ lepels water met een temperatuur van } (25 * [0,100] + 75 * [100,0]) / (25 + 75) = [0,25] + [75,0] = [50,0]$.
- **De Transmogrifeeerder [Transmogrifier].** Met de Transmogrifeeerder kunnen ingrediënten van toestand worden veranderd. Wanneer ingrediënten van toestand veranderen, zal er zoveel mogelijk van de hoeveelheid bewaard blijven, indien niet mogelijk zal er naar beneden worden afgerond. Dit betekent dat er een geringe hoeveelheid ingrediënt (kleiner dan een drop of pinch) kan verloren gaan bij het omzetten van de toestand.

1.4 Het Laboratorium [Laboratory]

Enkel in een laboratorium kan er iets met alchemische ingrediënten gebeuren. Toestellen zullen zich dus in het labo moeten bevinden alvorens ze gebruikt kunnen worden.

1.4.1 Capaciteit [Capacity]

Een laboratorium heeft een bepaalde hoeveelheid opslagruimte voor ingrediënten, uitgedrukt in een hoeveelheid opslagplaatsen (storerooms). Er moeten dan ook

methodes worden aangeboden om hoeveelheden ingrediënt in het laboratorium op te slaan en er terug uit te halen. Concreet betekent dit dat men aan het labo een **container met ingrediënt kan aanbieden**, waarbij het ingrediënt in het labo wordt opgeslagen en de container wordt vernietigd. Omgekeerd kan men ook een **hoeveelheid ingrediënt (op basis van naam) opvragen uit het labo**, waarbij een nieuwe container gecreëerd wordt waarin de gevraagde hoeveelheid wordt opgeslagen. Deze nieuwe container wordt teruggegeven.

We gaan er dus vanuit dat het laboratorium een eindeloze hoeveelheid containers (flessen, vaten, enz.) in voorraad heeft. Deze methode kan dus steeds worden uitgevoerd zolang de gevraagde hoeveelheid maar aanwezig is. **Ingrediënten worden door een laboratorium bij hun opslag wel steeds naar hun standaardtoestand gebracht, wat betekent dat ze hun standaardtemperatuur aannemen (door gebruik van een Oven en/of Koelkast).**

Verder worden de ingrediënten van dezelfde soort samen bewaard. Nieuwe ingrediënten worden (na het op temperatuur brengen) gemixt met het reeds aanwezige ingrediënt met dezelfde naam. (De eigenschappen van de opgeslagen ingrediënten worden dus door het mixen bepaald.)

Een voorbeeld: Men voegt 10 flessen verhitte vloeibare knoflook (die in één onvolledig gevuld vat (barrel) zitten) toe aan een labo waar reeds 1 vat knoflook in aanwezig is. Een vat is evenveel als 84 flessen, dus het eindresultaat zal zijn dat er 94 flessen knoflook in het labo aanwezig zullen zijn.¹ De nieuwe knoflook zal afgekoeld worden tot de standaardtemperatuur en dan bij de rest van de knoflook worden gevoegd.

De methode om iets uit een labo te halen zal de (eenvoudige of speciale) naam van een ingrediënttype en een bepaalde hoeveelheid als argumenten meekrijgen. Het resultaat van die methode is een ingrediëntencontainer gevuld met ingrediënt van dat type en die hoeveelheid, indien dat aanwezig was in het labo. Verder willen we deze methode ook oproepen met één parameter waarbij de volledige hoeveelheid van dat ingrediënt uit het labo zal teruggegeven worden. Indien dit niet in de grootste container past (een barrel of een chest), gaat daarbij de overschot verloren. **Tenslotte moet het ook mogelijk zijn om een overzicht op te vragen van alle in een laboratorium aanwezige ingrediënten (met minstens hun naam en de beschikbare hoeveelheid).**

Het aanwezig zijn van ingrediënten in een laboratorium wordt **defensief** uitgevoerd.

1.4.2 Alchemische Toestellen

In een laboratorium is er in principe **onbeperkte plaats voor alchemische toestellen**. Er kan echter slechts **één toestel van elke soort aanwezig zijn**. Alle alchemische bewerkingen die in een laboratorium worden uitgevoerd, maken gebruik van de aanwezige toestellen. **De link tussen een laboratorium en zijn toestellen dient bi-directioneel uitgewerkt te worden**, zodat hard gemaakt kan worden dat alchemische toestellen enkel gebruikt kunnen worden wanneer ze zich in een labo bevinden.

¹Je mag zelf beslissen of je deze hoeveelheid in 94 flessen, of in 1 vat en 10 flessen bijhoudt.

1.4.3 Recepten Uitvoeren

De alchemist zal in een laboratorium recepten uitvoeren. Hiertoe biedt het labo een methode `execute(Recipe,int)` aan. Het laboratorium voert dan stap voor stap elke bewerking van het recept uit om uiteindelijk aan het laatste mengsel te komen. De tweede parameter bepaalt met welke (gehele) factor de in het recept aangegeven hoeveelheden vermenigvuldigd moeten worden zodoende meer eindproduct te bekomen. Indien er niet genoeg ingrediënt aanwezig is, dan zal de uitvoering van het recept niet voltooiën. Dit betekent echter niet dat helemaal gefaald is. Alchemisten hebben geen tijd om op voorhand de nodige hoeveelheden te gaan controleren. Het recept wordt dus stap voor stap uitgevoerd tot het klaar is, of tot er een bepaalde hoeveelheid ingrediënt niet (of niet voldoende) aanwezig is. Het tot dan toe bekomen mengsel wordt opgeslagen in het labo, net als alle andere niet-gebruikte ingrediënten. Het mengsel en de eventuele reeds verwarmde of gekoelde ingrediënten worden hierbij opnieuw op hun standaardtemperatuur gebracht. Indien het uitvoeren van het recept wel slaagt, dan wordt het resultaat ook op standaardtemperatuur gebracht en eveneens in de opslagruimte opgeslagen. De gebruikte ingrediënten zijn uiteraard opgebruikt. De methode om een recept uit te voeren hoeft niet te worden gespecificeerd.

1.5 Recepten en Receptenboeken

1.5.1 Recepten [Recipe]

Een recept is in principe de combinatie van twee lijsten. Het bevat een lijst van bewerkingen die achtereenvolgens moeten worden uitgevoerd op een lijst van ingrediënten, om tot een bepaald product te komen. Er wordt hierbij geen rekening gehouden met karakteristieken van de ingrediënten in de lijst zoals temperatuur, enz. Er zijn 4 bewerkingen mogelijk:

- **add.** Haal de gevraagde hoeveelheid ingrediënt uit de voorraad en voeg deze toe aan het recept, voer verder voorlopig geen bewerkingen uit.
- **heat.** Verwarm het laatst gebruikte ingrediënt met 50 eenheden. Er is geen overeenkomstig ingrediënt in de tweede lijst. Het laatst gebruikte ingrediënt is het net toegevoegde, verwarmde (of afgekoelde) ingrediënt, tenzij de vorige bewerking een mix was. In dat geval is het laatst gebruikte ingrediënt het bekomen mengsel.
- **cool.** Analooq aan heat, maar koelt het laatst gebruikt ingrediënt af met 50 eenheden.
- **mix.** Mix alle reeds in het recept aanwezige ingrediënten. Er is geen overeenkomstig ingrediënt in de andere lijst.

Een recept bestaat dus b.v. uit twee lijsten van de volgende vorm:

- add, cool, add, heat, mix, add, cool, mix, add, heat, mix, heat, add, heat add, mix
- 3 drops Mercurial Acid, 1 vial Water, 2 spoons Imp Gas, 3 spoons Garlic, 4 drops Black Lotus, 1 spoon Mustard

Dit recept betekent: Voeg 3 drops gekoeld Mercurial Acid toe, voeg 1 verwarmd vial Water toe, mix alles, koel 2 spoons Imp Gas af en voeg toe, mix alles, verwarm 3 spoons Garlic en voeg toe, mix en verwarm het mengsel, verwarm 4 drops Black Lotus en voeg toe, voeg 1 spoon Mustard toe en mix alles. Elke mix bewerking betekent dat alles wat we tot dan toe behandeld werd, inclusief eventuele vorige mengsels, vermengd worden tot 1 nieuw mengsel. In principe moet de laatste bewerking mix zijn. Ingeval die niet voorkomt in het recept wordt ze sowieso toch uitgevoerd als laatste bewerking.

1.5.2 Receptenboek [RecipeBook]

Een receptenboek is een verzameling recepten. De volgorde waarin recepten in een boek zitten, staat vast; wanneer men een nieuw recept in een boek aanmaakt, komt het achteraan in het boek terecht. Wanneer een recept uit een boek verwijderd wordt, dan wordt de pagina gewoon uit het boek gescheurd. (Veronderstel dat elk recept exact 1 pagina inneemt.) Recepten kunnen in verschillende boeken voorkomen. Wanneer een recept verwijderd wordt uit één boek heeft dat geen impact op de andere boeken.

De relatie tussen een receptenboek en zijn recepten dient uni-directioneel uitgewerkt te worden.

2 Constructoren en Testen

De constructoren van alle gevraagde klassen zijn in principe vrij te kiezen, tenzij anders aangegeven, al moet er natuurlijk wel rekening worden gehouden met de nodige constraints en klasse-invarianten.

Wat de verificatie betreft, verwachten we *niet* dat je elke methode uitgebreid gaat testen. Beperk je tot de toestellen en de methodes die van daaruit gebruikt worden. Er wordt uiteraard nog steeds gebruik gemaakt van het `jUnit` framework.

3 Ontwerptekening

Naast de specificatie en implementatie van de gevraagde klassen, moet ook een schets van het ontwerp worden gemaakt. Deze bevat alle ontwikkelde klassen, behalve zelfgedefinieerde uitzonderingsklassen en testklassen. Het ontwerp moet uitgetekend worden in UML. Neem voor alle klassen de naam, de velden en de publieke methoden op.

4 ~~Beperkte opgave voor kleinere teams~~

~~Teams die niet uit langer uit 3 studenten bestaan dienen dit zo snel mogelijk te melden via email aan tommy.messelis@kuleuven.be en jan.goedgebeur@kuleuven.be, met vermelding van de reden. Studenten die dit vak voor de tweede keer opnamen maken het project ook individueel. Alle teams die uit slechts 1 of 2 studenten bestaan mogen alle zaken in verband met recepten en receptenboeken negeren.~~

~~Studenten die individueel werken mogen daarnaast ook het opslaan van ingrediënten in een labo negeren. Voor hen dient een labo enkel om toestellen te huisvesten.~~

5 Praktische afspraken

Het examenproject OGP is verplicht uit te werken: ieder team moet een oplossing voor de vooropgestelde datum inleveren! **Het tijdig en correct indienen van het project is een voorwaarde voor deelname aan het examen!** Het project moet in dezelfde groepjes gemaakt worden als de practica. Er mag altijd overlegd worden met andere studenten over mogelijke oplossingen. Uiteindelijk moet ieder groepje voor zichzelf beslissen hoe bepaalde aspecten best worden uitgewerkt.

5.1 Indienen van het project

Het project wordt ingediend via de opdracht op Toledo, **ten laatste op vrijdag 20 mei 2022, 23:59**.¹ 1 indiening per team volstaat. Zorg ervoor dat de namen van de teamleden vermeld worden in de hoofding van de klasse(n) (als `@author`). Je oplossing bestaat uit een .zip of .jar bestand dat alle broncode bevat, en een pdf met de ontwerptekening.² Let erop dat je al je ontwikkelde klassen (ook de JUnit klassen) selecteert (Select the resources to export), en neem zeker de .java bestanden op! (**Eclipse neemt deze default niet op in het JAR-bestand!**)

5.2 Examen

Het examenproject wordt op 5 van de 20 punten van het vak gequoteerd en dient als basis voor de vraagstelling op het examen zelf. De overige 15 punten worden bepaald **door je antwoorden op het examen zelf**. Een goed project leidt dus niet automatisch tot een goede score. Omgekeerd maakt een project van mindere kwaliteit het uiteraard wel moeilijker om een goed examen af te leggen.

5.3 Vragen

Vragen omtrent het project kan je steeds per email stellen aan tommy.messelis@kuleuven.be en jan.goedgebeur@kuleuven.be. Stuur je email steeds naar beide adressen.

Veel succes!

²Je kan met de Eclipse omgeving eenvoudig een JAR-bestand genereren met de export functie die je o.a. in het File-menu vindt. De tutorial op Toledo beschrijft ook dit proces.