

Opgave 3: UNO

Leerdoelen

Na afloop van deze opdracht kun je:

- Lijsten maken in Java;
- Elementen op iedere plek aan een lijst toevoegen;
- Efficient elementen aan het eind van een lijst toevoegen;
- Elementen uit een lijst verwijderen;
- Een Iterator maken voor een linked list.

Uno tegen de computer

In deze opgave vragen we je een variant van het spel *Uno* te maken waarin één speler tegen de computer kan spelen. Uno is een kaartspel dat erg op pesten lijkt. In Uno worden de volgende kaarten gebruikt:

- gewone kaarten in de kleuren *Geel*, *Groen*, *Blauw* en *Rood* met waarde 0 tot en met 9, van iedere kaart zijn er twee stuks in het spel;
- kaarten met de betekenis *neem-er-2*, van alle vier de kleuren zijn er twee kaarten;
- kaarten met de betekenis *keer-speelrichting-om*, ook hiervan zijn er twee van alle vier de kleuren, als je maar met twee spelers bent (zoals in dit geval) heeft dit dus geen betekenis;
- kaarten met de betekenis *sla-beurt-over* ook hiervan zijn er twee van elke kleur;
- 4 *keuze* kaarten;
- 4 *keuze-en-neem-er-4* kaarten. Zie verderop voor uitleg over deze kaarten.

In het begin krijgt iedere speler 7 kaarten. De rest van de kaarten ligt met het beeld naar beneden op een stapel. De speler die het eerst alle kaarten weg gespeeld heeft wint het spel. Een speler mag een kaart wegleggen als deze dezelfde kleur of de zelfde waarde heeft als de laatste kaart in de stapel. Iedere gespeelde kaart wordt achteraan de stapel toegevoegd. Als een speler niet kan of wil spelen pakt zij een kaart van de stapel, als deze kaart past mag deze in de zelfde beurt nog weggelegd worden. Als de kaart niet past wordt deze op de hand genomen.

In deze versie van het spel mag de mens altijd beginnen. De computer toont steeds de laatste kaart van de stapel en alle kaarten in de hand van de speler. De speler tikt het nummer van de te spelen kaart in, als deze kaart bestaat en past wordt deze kaart gespeeld. Anders geeft de computer de bovenste kaart van de stapel aan de speler. Als deze kaart direct gespeeld kan worden, doet het programma dat automatisch ook. Natuurlijk mag je inbouwen dat het programma aan de gebruiker vraagt of deze kaart gespeeld moet worden, of op de hand genomen moet worden.

De speciale kaarten

De speciale kaarten hebben de volgende betekenis:

- Een *neem-er-2* kaart kan gespeeld worden als de kleur gelijk is aan kleur van de laatst gespeelde kaart. De tegenstander neemt 2 extra kaarten van de stapel. Als maker van het spel mag je beslissen wie er aan de beurt is.
- Een *keer-speelrichting-om* kaart mag ook alleen gespeeld worden als de kleur past. Deze kaart heeft verder geen betekenis in deze versie van het spel omdat er maar twee spelers zijn.

- Na het spelen van een *sla-beurt-over* kaart mag de zelfde speler direct nog een kaart leggen.
- Een *keuze* kaart past altijd. Op zo'n keuze kaart past elke andere kaart. Het is aan de makers van het spel wie er mag beslissen met welke kleur er verder gespeeld mag worden.
- Een *keuze-en-neem-er-4* kaart past ook altijd. De volgende speler krijgt 4 extra kaarten wanneer deze kaart gespeeld wordt.

In de officiële versie van spel mogen de kaarten voor 2 en 4 kaarten pakken gestapeld worden. In plaats van kaarten te pakken speelt de speler een kaart om net zo'n aantal kaarten te spelen. De volgende speler moet dan de som van het aantal kaarten pakken (of weer zo'n kaart spelen natuurlijk). Voor meer informatie kun je kijken in de officiële spelregels, op een van de vele pagina's op internet over Uno (zie bijvoorbeeld [http://nl.wikipedia.org/wiki/UNO_\(kaartspel\)](http://nl.wikipedia.org/wiki/UNO_(kaartspel))), of de uitgebreidere engelse verie). Je mag deze stapel regel natuurlijk implementeren, maar dat hoeft niet.

Opdracht

In deze opdracht gaan jullie dit spel implementeren. Om je op weg te helpen vindt je op BlackBoard een klasse `Kaart`. Implementeer het spel bijvoorbeeld in de volgende stappen.

1 Gelinkte lijst in Java

In dit spel moeten er steeds kaarten van de stapel naar de handen en van de handen weer naar de stapel. We gaan zowel de handen als de stapel implementeren met een gelinkte lijst. Het leerdoel van deze opgave is snappen hoe die gelinkte lijsten geïmplementeerd moeten worden. Op Bb vindt je het raamwerk voor een eigen klasse `Lijst`. Alle benodigde methoden van `Lijst` bestaan. De implementatie van die methode is echter leeg, of bevat juist genoeg code om de compiler tevreden te stellen. De bedoeling is dat je alle methoden zelf implementeert. Op Bb vindt je ook een klasse `Main` die wat simpele testjes met die lijsten van kaarten doet.

2 De stapel

De stapel bevat initieel alle genoemde kaarten. Iedere speler krijgt aan het begin van het spel 7 kaarten van deze stapel. Zowel de stapel als de handen kunnen direct gemaakt worden met de boven genoemde klasse `Lijst`.

3 De computer als speler

Als de computer aan de beurt is legt deze een passende kaart op de stapel als dat kan. Zo niet dan pakt de computer een kaart van de stapel. Dit horen simpele lijst manipulaties te zijn. Het is goed genoeg als de computer de eerste kaart uit zijn hand die past speelt, een slimme strategie is niet nodig.

4 De mens als speler

Definieer een klasse om de mens te laten spelen. Als de mens aan de beurt is worden alle kaarten getoond en mag er een passende kaart gespeeld worden, of een kaart gepakt worden.

5 Het spel

Combineer de boven gemaakte ingrediënten tot een compleet spel. De mens en de computer om de beurt spelen. Het spel stopt zodra een van de spelers geen kaarten meer in de hand heeft. Bedenk wat de handigste manier is om een van de spelers een beurt over te laten slaan of om een speler extra kaarten te geven.

Extra: *dit onderdeel hoef je niet te maken*

Voor mensen die behoefte hebben aan extra uitdaging kunnen het spel implementeren volgens de officiële spelregels die je ook op BlackBoard kunt vinden. Een andere mogelijkheid is om de computer slimmer te laten spelen dan de eerste kaart die past.

Inleveren van je producten

Lever vóór maandag 21 februari 8.30 uur via Blackboard jullie Java code in. Vergeet niet om **in** je uitwerking duidelijk jullie namen te vermelden.