# Opgaven bij de game Solitaire

1. **Representatie van de speelkaarten**

In het spel wordt een principe gebruikt om speelkaarten aan te geven met een getal.

Zoek uit wat het principe is en met welke classes en methodes dat principe wordt ingevuld.

Maak daar een korte beschrijving van

En: laat zien dat je het principe doorhebt door een “Nederlandse vertaling’ van het spel te maken.

*Hulpvraag:*

Hoe kun je bijvoorbeeld bepalen of een kaart de ‘waarde’ 3 heeft of dat het een kaart is van de ‘kleur’ klaveren?

1. **Groeperen van de kaarten**

De werking van het spel is gebaseerd op het werken met een aantal verschillende soorten groepjes/stapels kaarten.

Zoek uit hoe die groepjes heten en welke programmacode het gedrag van die groepjes definieert.

1. **Uitvoeren van ‘unit tests’**

Met elke *class* kun je een afgebakend stuk van je programmacode maken. Doordat het zo is afgebakend, kun je het ook apart testen om te kijken of de code inderdaad precies doet wat de opzet is.

Bij de eerste twee opdrachten heb je uitgezocht welke classes je kunt gebruiken om te werken met speelkaarten en met stapels kaarten. Bij deze opdracht ga je uitzoeken hoe je die classes kunt gebruiken om de basisfuncties uit te proberen die je nodig hebt voor het spel.

Schrijf een testprogramma waarmee je vier stapeltjes kaarten maakt: de kaarten van elk stapeltje hebben dezelfde kleur en ze hebben en waarde die oploopt van aas tot heer. Druk vervolgens de kaarten van de stapeltjes af (elke kaart op een nieuwe regel, tussen twee stapeltjes in een lege regel).

Maak nu een programma voor dertien stapeltjes: binnen elk stapeltje hebben de kaarten dezelfde waarde en ze hebben een kleur die is gesorteerd in de vaste volgorde harten, klaveren, ruiten, schoppen (en NIET in een andere volgorde!).

Tot nu toe heb je functies getest die goed werken. Je hebt je immers (waarschijnlijk onbewust) gehouden aan de uitgangspunten waarbinnen de code is ontworpen. Maar wat zijn die uitgangspunten dan precies. En wat gebeurt er als je iets wilt doen dat niet aan die uitgangspunten voldoet? Stel bijvoorbeeld dat je een lege stapel kaarten hebt. Of een kaart van een ‘vijfde’ kleur?

Bedenk eerst een paar van dit soort tests zonder naar de code te kijken en schrijf op wat je wilt testen. Kijk dan naar de code en bedenk dan tests die je kunt gebruiken om dingen te testen waarvan je verwacht dat ze ‘net wel’ of ‘net niet’ goed zullen gaan. Selecteer dan vervolgens voor ieder lid in je groepje een test en werk die individueel uit.

1. **Opstarten van het spel**

Bij het begin van het spel worden de kaarten geschud en in stapeltjes uitgelegd.

Het schudden zorgt ervoor dat je steeds weer met een nieuwe spelsituatie begint. Dat maakt het spel ‘echt’, maar het maakt het ook lastig om specifieke situaties goed uit te testen.

Speel het spel een paar keer om te ontdekken hoe het werkt en wat het effect is van het ‘schudden’ van de kaarten: wanneer zijn de kaarten gunstig geschud en bij welke situaties kom je vast te zitten?

Zoek uit met welke code de kaarten worden geschud en hoe ze daarna over stapeltjes worden verdeeld. Bedenk dan vervolgens hoe je de kaarten ‘geschud’ wilt hebben zodat je zeker weet dat je het spel uit kunt spelen.

En dan komt de ‘cheat’: maak een nieuwe method aan (bv. SchuddenSpeciaal) die de kaarten in de door jou gewenste volgorde klaarzet.

*Tip*

Zoek uit wat je kunt met *.mapToObj*, en *IntStream.of*

1. **Andere spelstructuur**

In het spel leg je de kaarten in zeven kolommen uit (A t/m G).

Zoek uit wat je moet veranderen om er zes kolommen van te maken.

1. **‘Spieken’ en ‘Stiekem ruilen’**

Als je de patience speelt, dan is het steeds een verrassing wat je aantreft als je een ‘dichte kaart’ mag omdraaien. Soms heb je liever niet dat er een heer ligt en soms komt dat juists goed uit.

Bedenk twee nieuwe handelingen die je als speler kunt doen: ‘spieken’ en ‘stiekem ruilen’.

Bij ‘ spieken’ kijk je wat de waarde en de kleur van dichte kaarten is.

Bij ‘stiekem ruilen’ wissel je twee dichte kaarten om.

Probeer het ‘fool proof’ te maken, zodat je alleen twee kaarten kunt omruilen die allebei dicht liggen!

Gebruik je cheat uit opdracht 4 en pas hem zo aan dat je de twee nieuwe handelingen kunt demonstreren.

1. **Andere spelregels**

Misschien krijg je door het spelen van het spel de behoefte om de spelregels zo aan te passen dat je de patience makkelijker kunt uitspelen.

Bij de versie die je hebt gekregen mag je in een open kolom alleen een heer bovenaan leggen. Zoek uit wat je moet doen om die regel uit te schakelen en kijk of je het spel dan inderdaad vaker uit kunt spelen.

Een andere variant is dat je niet op de kleur let als je kaarten onderaan een kolom legt. Wat moet je hiervoor doen?

1. **Grafische user interface**

Het maken van een grafische user interface is een behoorlijke uitdaging.

Bedenk hoe je dit zou willen aanpakken. Kun je een eerste aanzet maken waaruit blijkt wat je aanpak is?