

Het Strategy Pattern

Objectgeoriënteerde analyse en ontwerp

Strategy Pattern

Doel

Het Strategy Pattern definieert **een familie van algoritmen**, isoleert ze en maakt ze **uitwisselbaar**.



Gedrag

The diagram consists of two blue callout boxes. One box, labeled 'Gedrag', points to the word 'uitwisselbaar' in the text above. The other box, containing the text 'dat veranderd kan worden', points to the word 'algoritmen' in the same text.

dat veranderd kan
worden

Strategy maakt het mogelijk om het algoritme, los van de client die het gebruikt, te veranderen.

Strategy Pattern



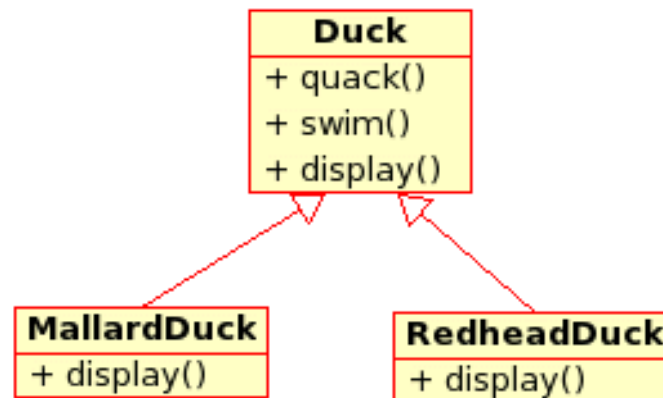
Een machine met
'uitwisselbare
algoritmen'

Strategy Pattern

Hoe het beter niet gedaan wordt: overerving

Voor elk combinatie van object en algoritme/gedrag een unieke klasse.

Gemeenschappelijk gedrag in een superklasse.



Strategy Pattern

Problemen bij de aanpak met overerving

Dit systeem is slecht onderhoudbaar.

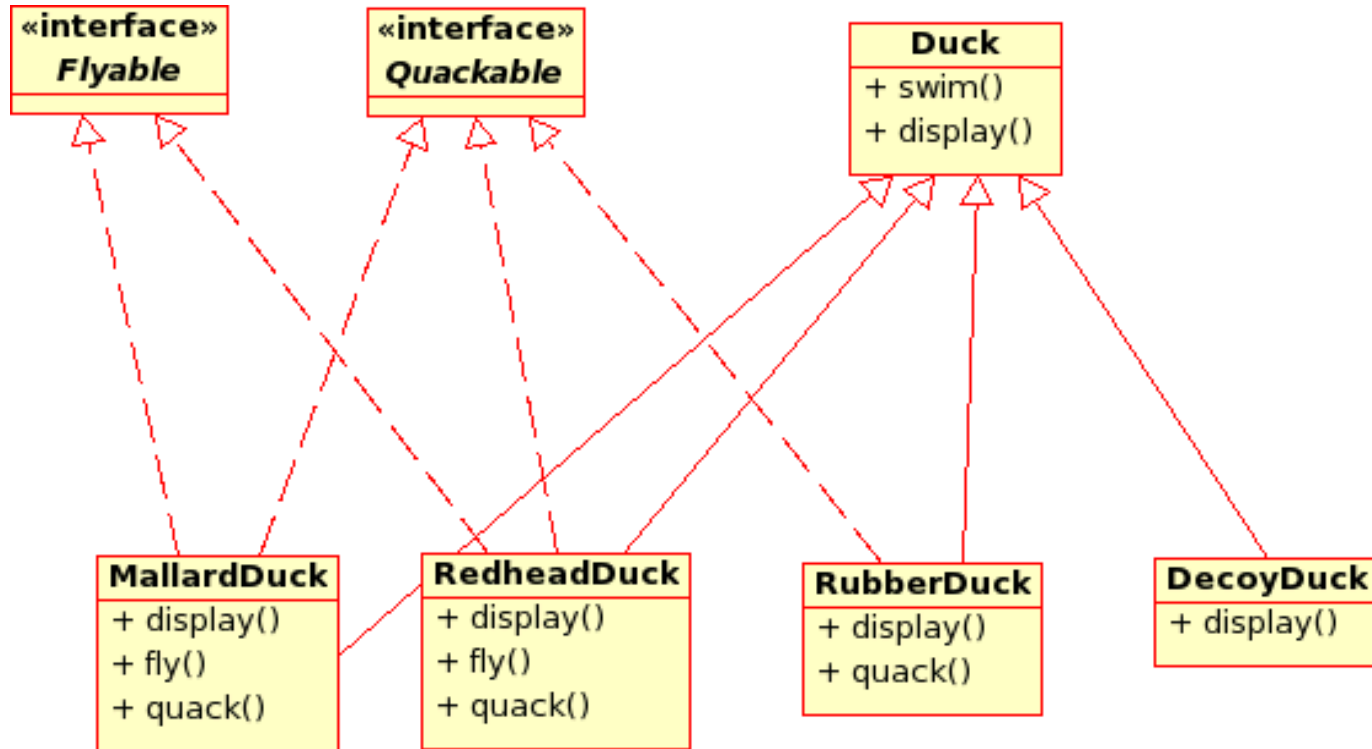
Veranderingen in de basisklasse kunnen problemen veroorzaken in de afgeleide klassen.

vb: een gedrag toevoegen dat sommige afgeleide klassen wel nodig hebben, maar andere niet.

Het gedrag is ook niet aanpasbaar at runtime.

Strategy Pattern

Ook niet goed: overerving met interfaces



Strategy Pattern

Probleem bij overerving met interfaces

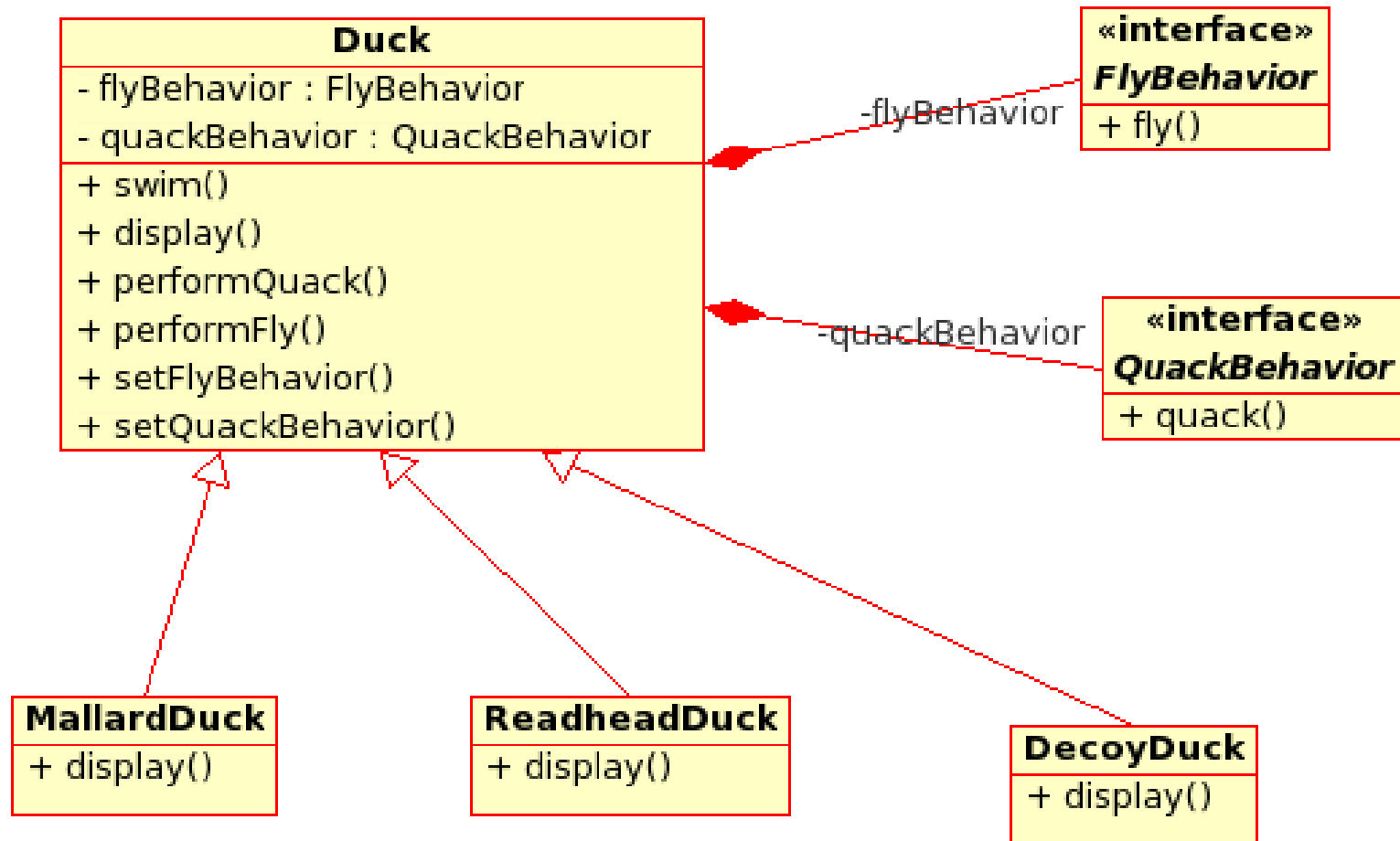
Te veel **codeduplicatie!**

In de interface zit geen implementatie

-> elke afgeleide klasse moet een eigen implementatie voorzien, ook al is die misschien voor veel afgeleide klassen hetzelfde.

Strategy Pattern

De oplossing: interfaces en compositie



Strategy Pattern

De oplossing: interfaces en compositie

- Het gemeenschappelijk gedrag in de basisklasse
- ONVERANDERLIJK gespecialiseerd gedrag in de afgeleide klassen
- VERANDERLIJK gespecialiseerd gedrag toevoegen met compositie via een interface (de interface zorgt voor uitwisselbaarheid)

Strategy Pattern

Ontwerpprincipe: isoleer wat veranderlijk is

Zoek uit wat veranderlijk is. Scheid dat van wat hetzelfde blijft.



Strategy Pattern

Ontwerpprincipe: liever compositie dan overerving

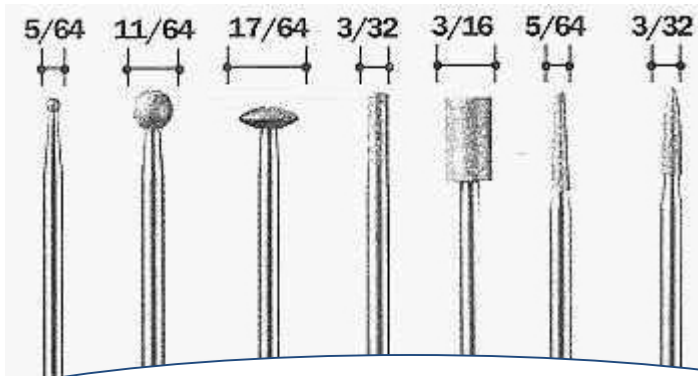
Verkiez compositie boven overerving.



Strategy Pattern

Ontwerpprincipe: programmeer naar interfaces

*Programmeer naar een interface,
niet naar een implementatie.*

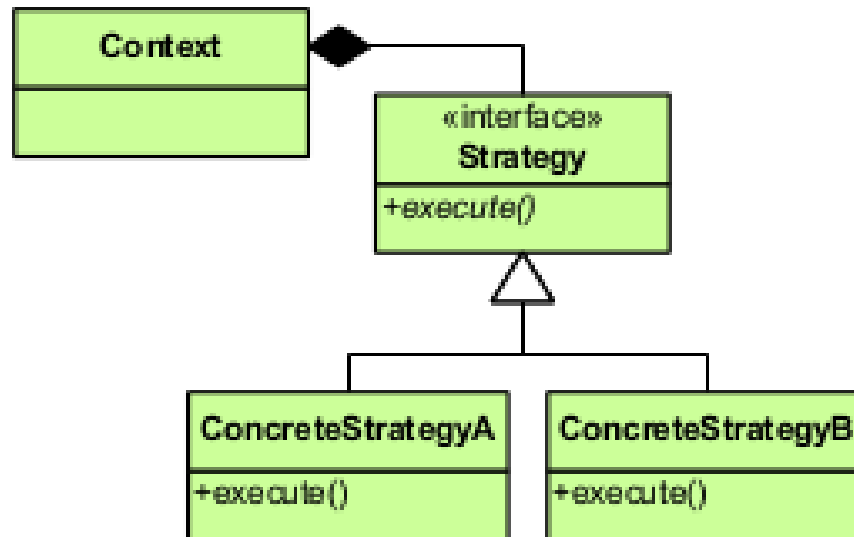


Alle stelen zijn gelijk!

Strategy Pattern

Structuur en terminologie

- Andere naam: **policy pattern**
- Onderdelen: context, strategy, concrete strategy



Strategy Pattern voorbeeld in de jdk

In de jdk kun je collecties sorteren met het strategy pattern:

```
java.util.Collections#sort(List list, Comparator < ? super T > c)  
java.util.Arrays#sort(T[], Comparator < ? super T > c)
```



Strategy!