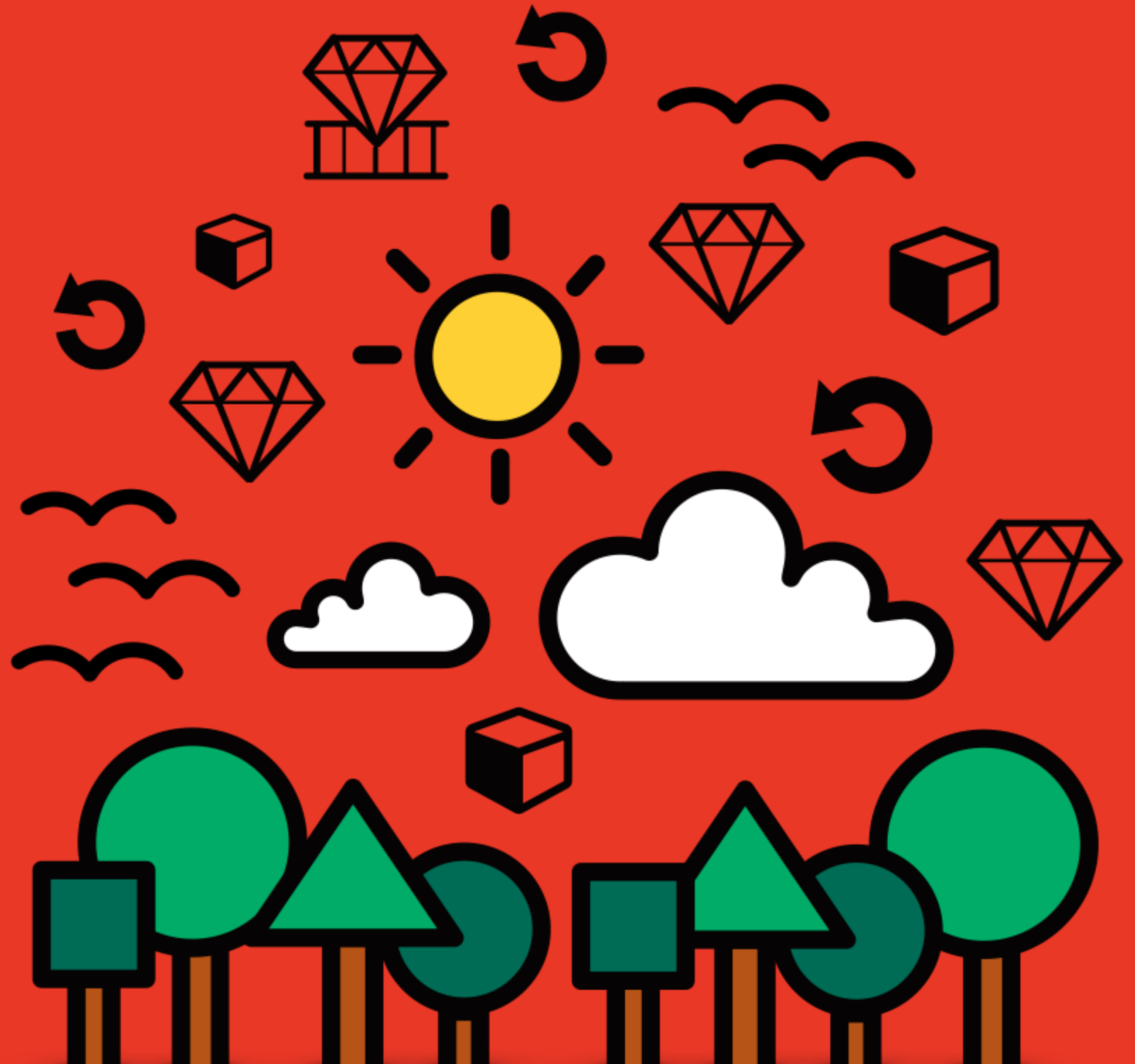


Object Oriented Programming



by Benjamin Udink ten Cate



- **Objects in daily life**
- **Applying objects to programming**
- **Classes**
- **More classes**
- **Objects**
- **Inheritance**
- **Multiple files**



Snoes

Applying this knowledge





```
cat = {name: 'Snoes', fur_color: :red, eyes: 2}
```


Symbols

- **:symbol**
- **symbol: value (:symbol => 'value')**



```
cat = {name: 'Snoes', fur_color: :red, eyes: 2, last_meal: 8_hours_ago}
```

```
def feed(cat)
  if(cat[:last_meal] < 6_hours_ago) do
    puts('feed the cat')
  end
end
```



```
class Cat  
end
```


Class

- A blueprint
- All members are private
 - characteristics
 - functions

Instance variables

- Always start with @
- not available outside object
- are given a value with constructor or set methods



Koekie



```
class Parrot  
end
```


Create instances

- Create objects from classes
- Constructor method
- **pluk** = Cat.new(2, 'Pluk', 'male', 'grey')



Questions?



Workshop

Objectify your shop

Split up your project into files

Create a Product superclass

Create classes for your Products

Create a Shopping Cart

...

16:45 DEMO