Stack Builders

# Forget about classes, welcome objects

Alexandre de Oliveira


Greenland
CANADA
RUSSIA
NORTH ATLANTIC OCEAN
ITED STATES
CHINA
ALGERIA
LIBYA
EGYPT
IRAN
SAUDI ARABIA
INDIA
MALI
NIGER
CHAD
SUDAN
NIGERIA
ETHIOPIA
SOMALIA
BRAZIL
PERU
BOLIVIA
PARAGUAY
ARGENTINA
SOUTH ATLANTIC OCEAN
DEMOCRATIC REPUBLIC OF THE CONGO
TANZANIA
ANGOLA
ZAMBIA
MOZAMBIQUE
SOUTH AFRICA
INDIAN OCEAN
French Southern and Antarctic Lands
reference-guides.com

# Agenda

## Perspective

## Objects

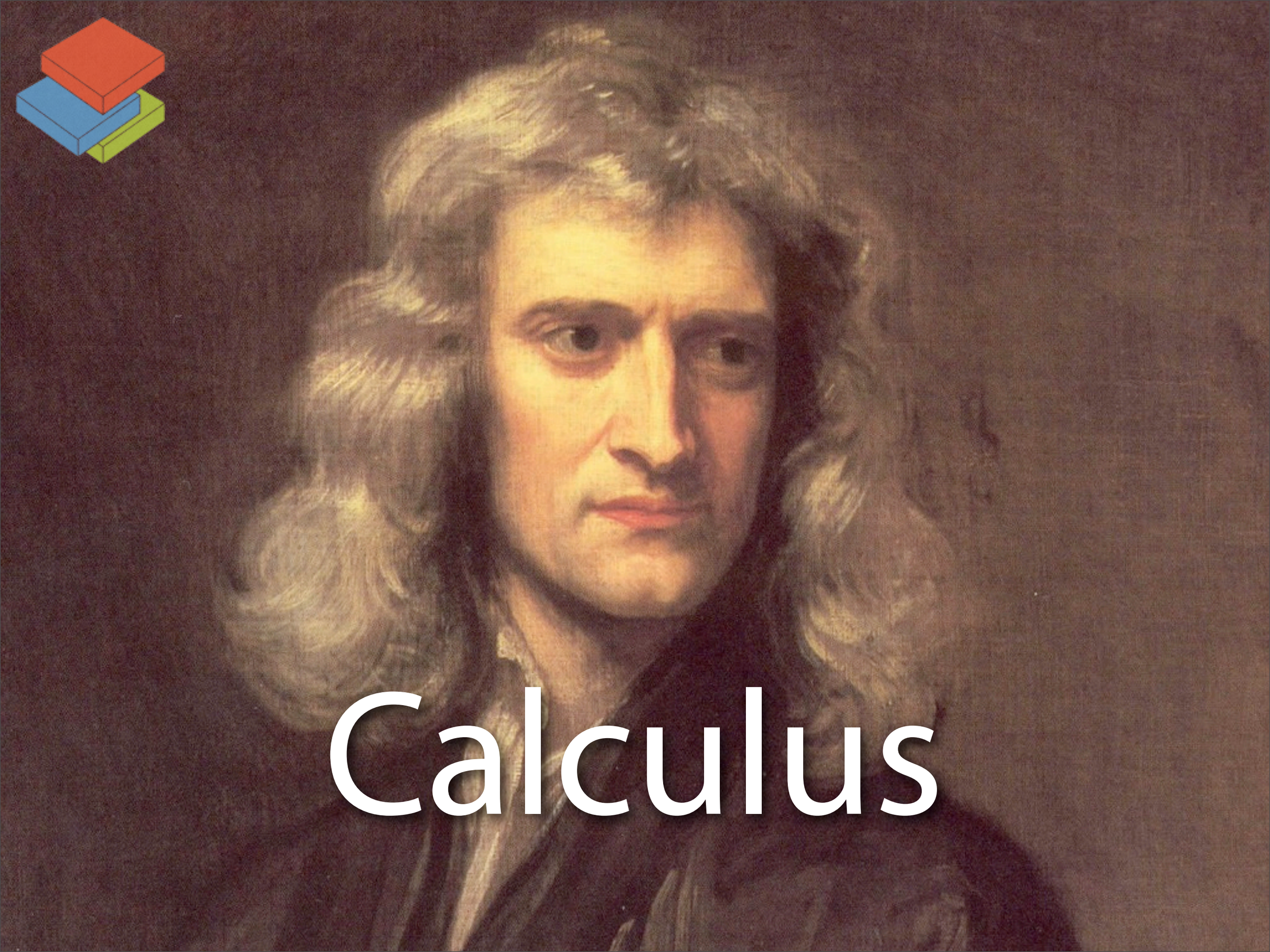
## Classes

# Perspective

# “Point of view is worth 80 IQ **points**”, Alan Kay

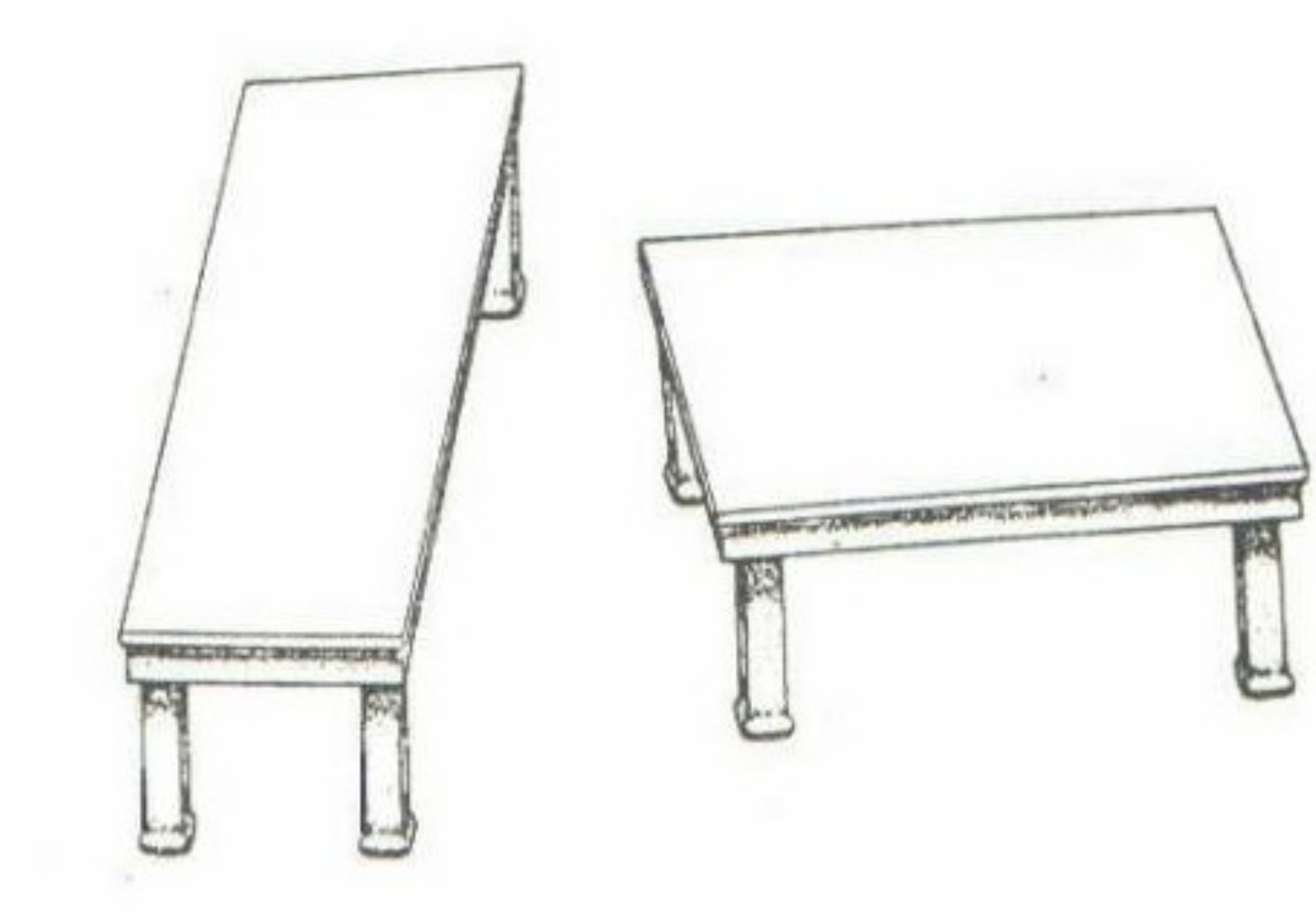
# VI

# 6

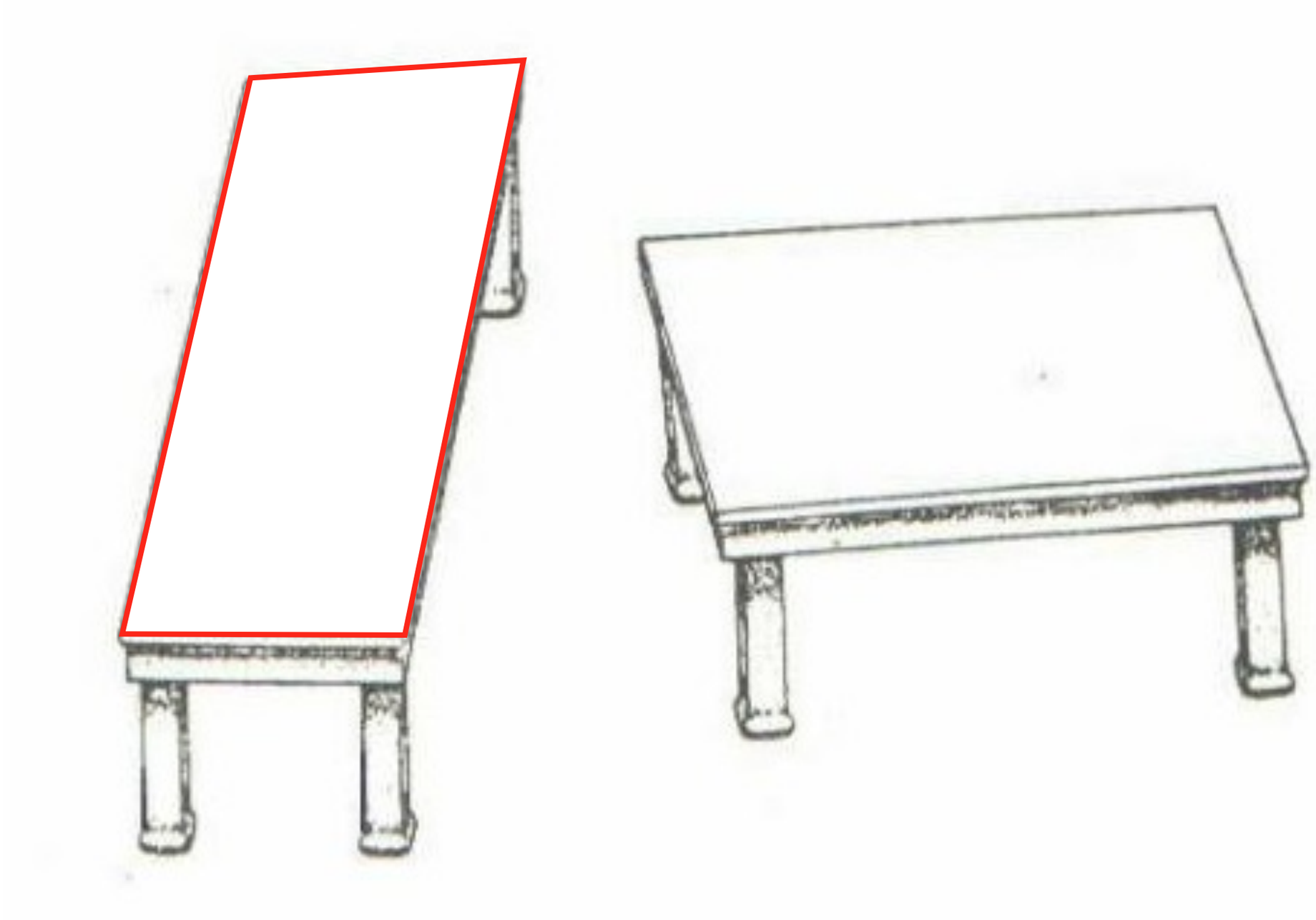


# Calculus

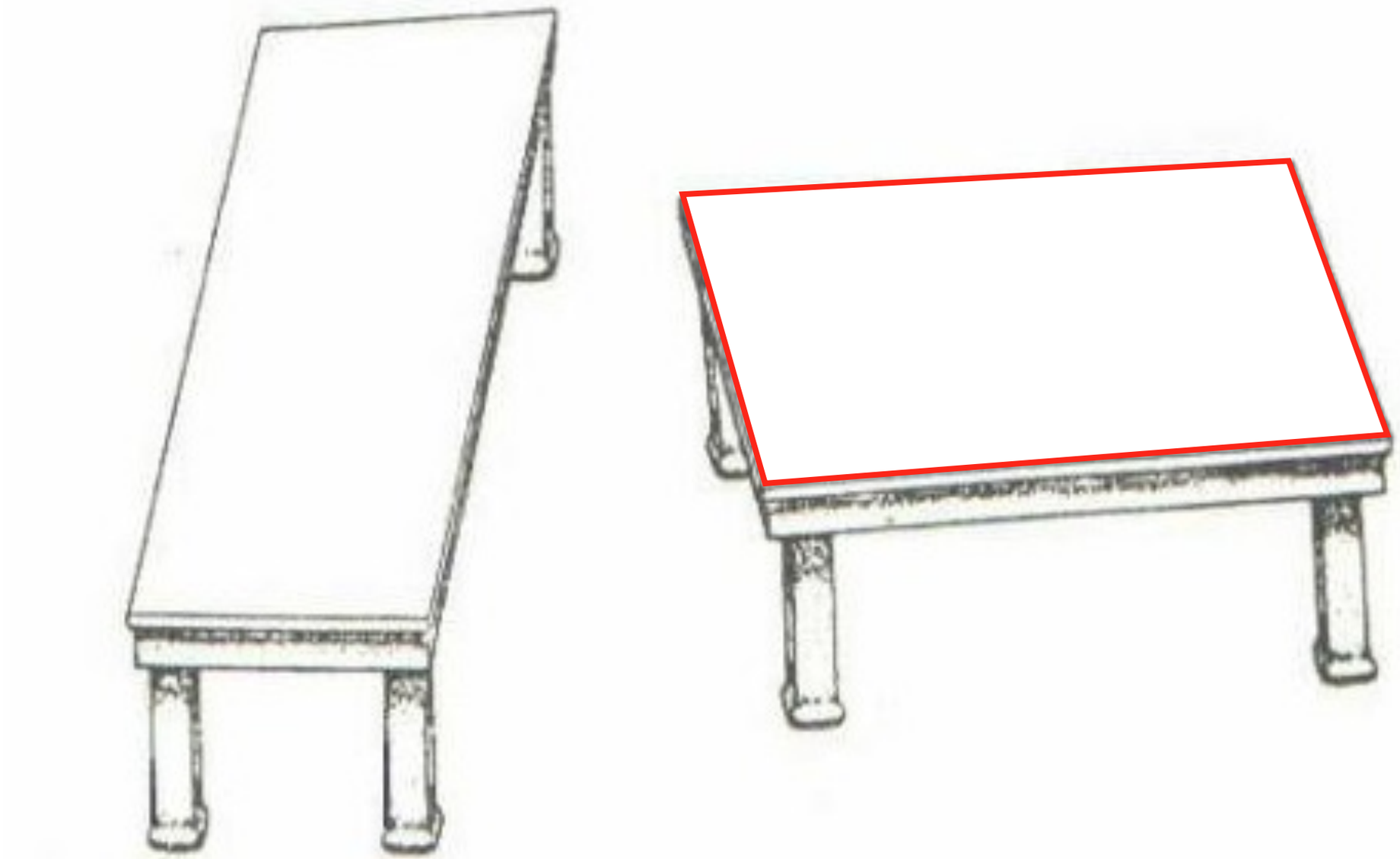
# The reality we create



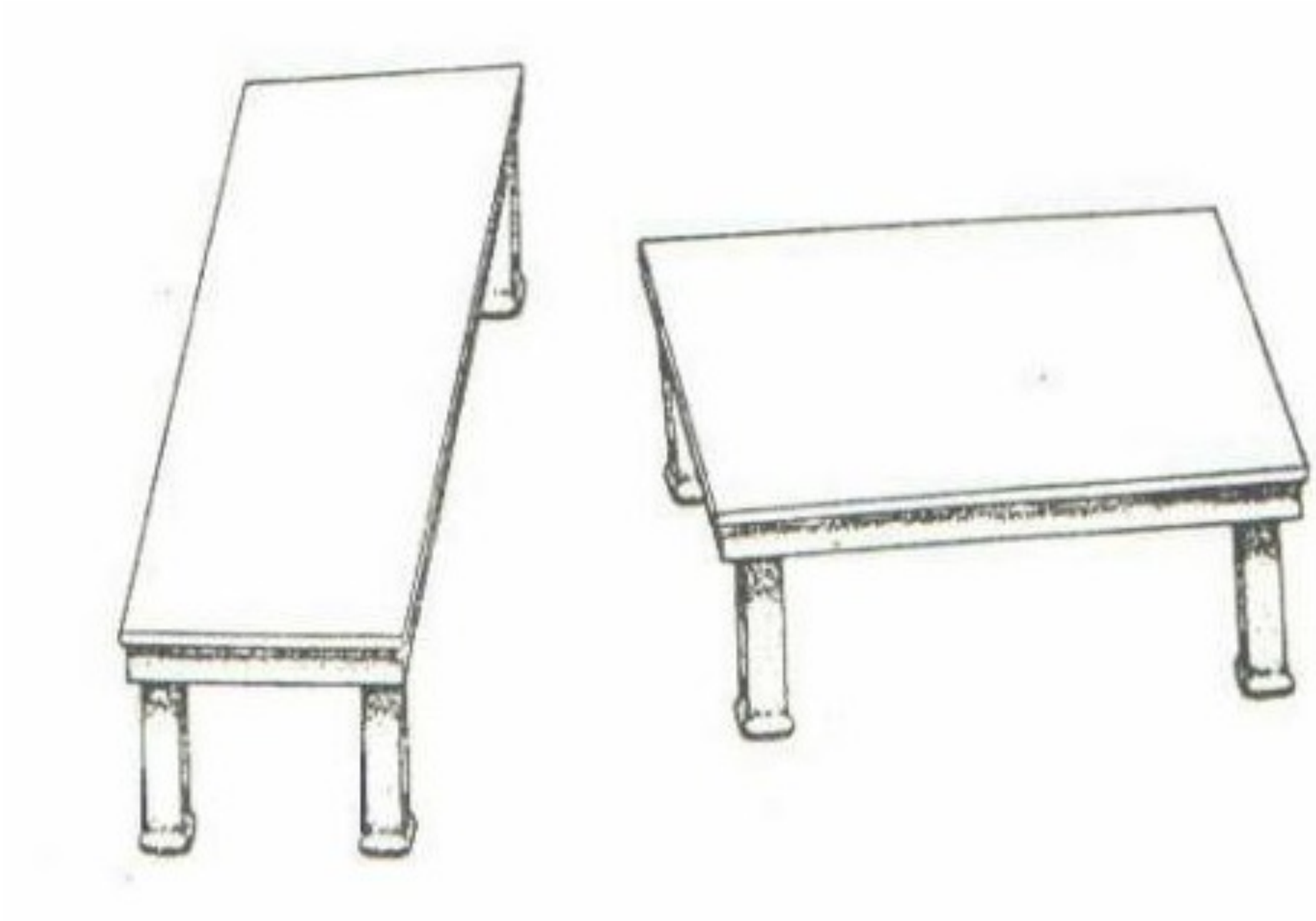
# The reality we create



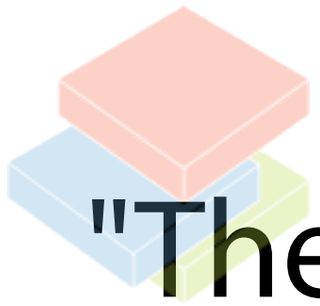
# The reality we create



# The reality we create

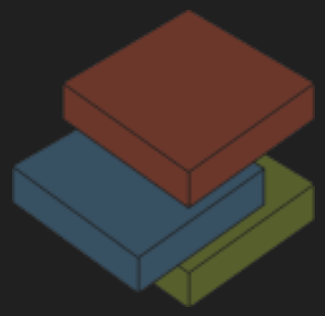



UNIVERSITY OF
GEORGIA



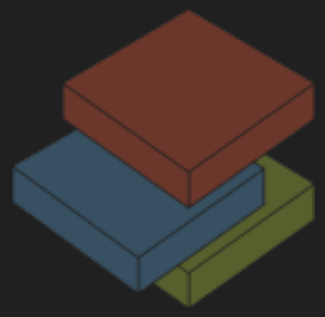
"The programmer, like the poet, works only slightly removed from pure thought-stuff. He builds his castles in the air, from air, creating by exertion of the imagination. Few media of creation are so flexible, so easy to polish and rework, so readily capable of realizing grand conceptual structures."

*The Mythical Man-Month*

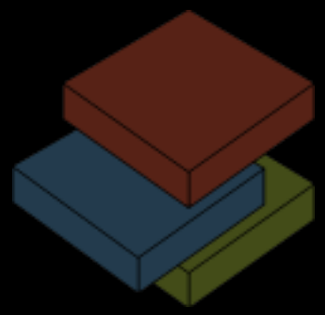


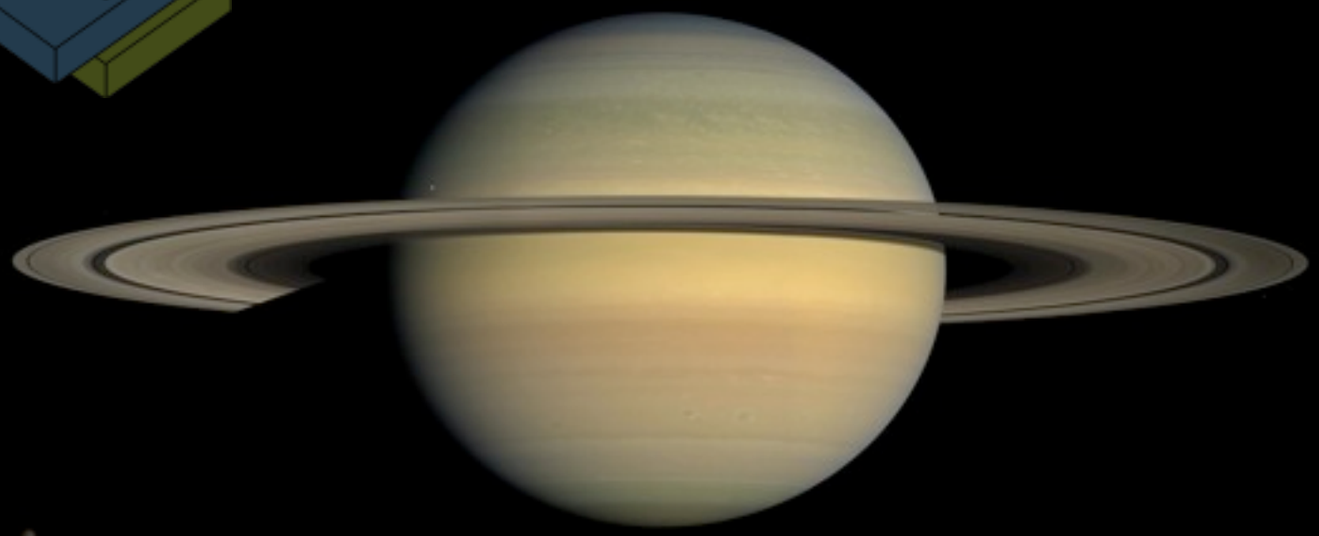
# OOP?

“OOP to me means only messaging, local retention and protection, hiding of state”, *Alan Kay*

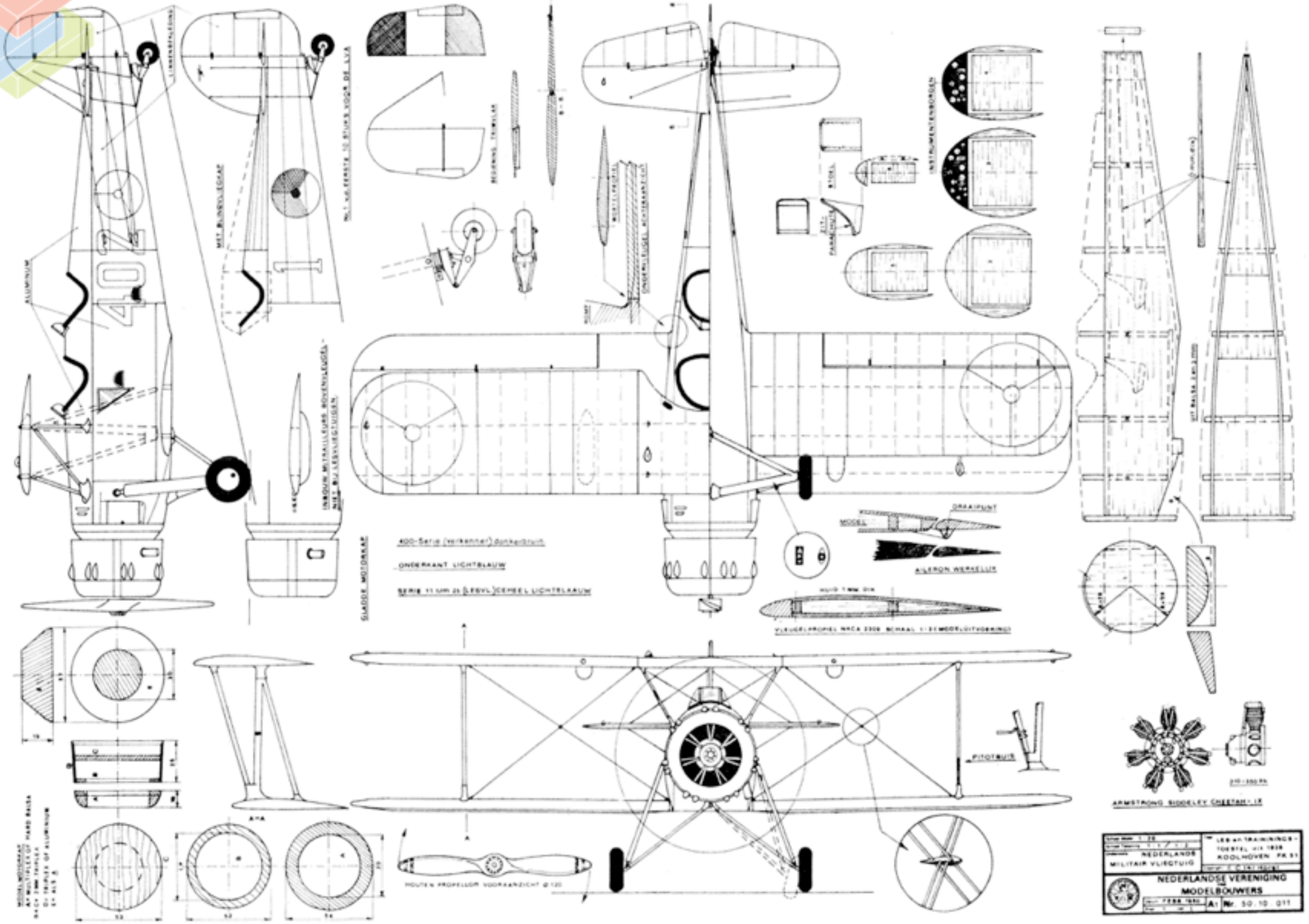


# Classes & Objects



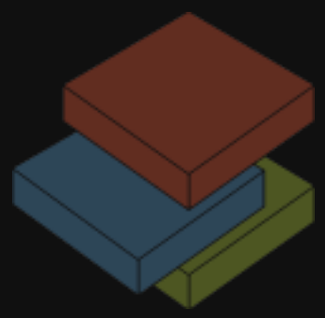


Source: https://commons.wikimedia.org/wiki/File:Saturn_during_Equinox.jpg

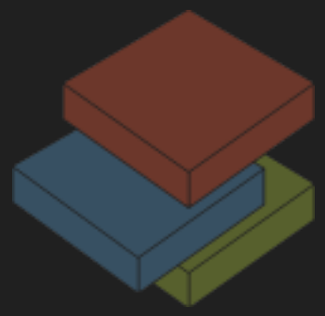



Source: http://www.smcars.net/forums/planes/8135-aircraft-koolhoven-fk-51-a.html

Source: http://www.flickr.com/photos/flyforfun/2694158656/

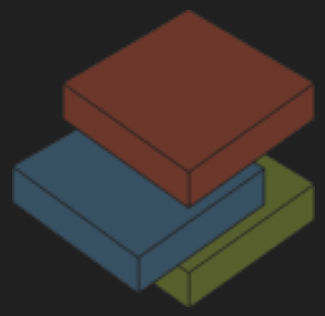


```
class Importer
  def self.process!
    # ...
  end
end
```



# Developer's point of view


**Person calls Pencil#write**



# Object's point of view

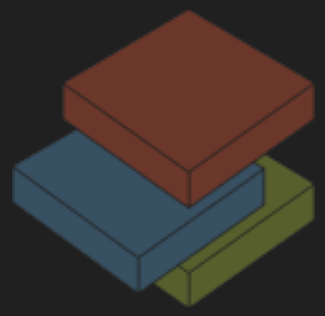
someone else

pay()

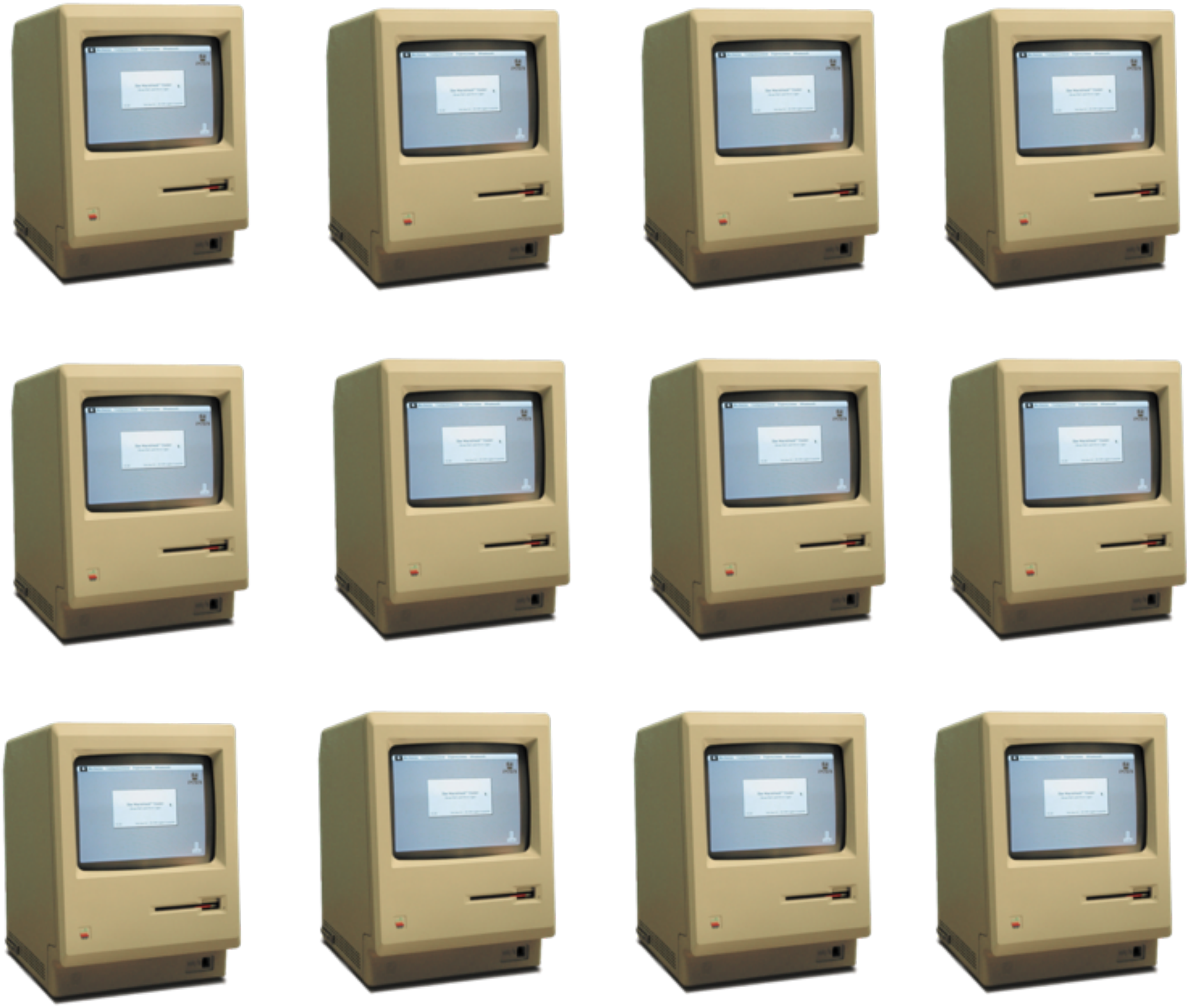
someone

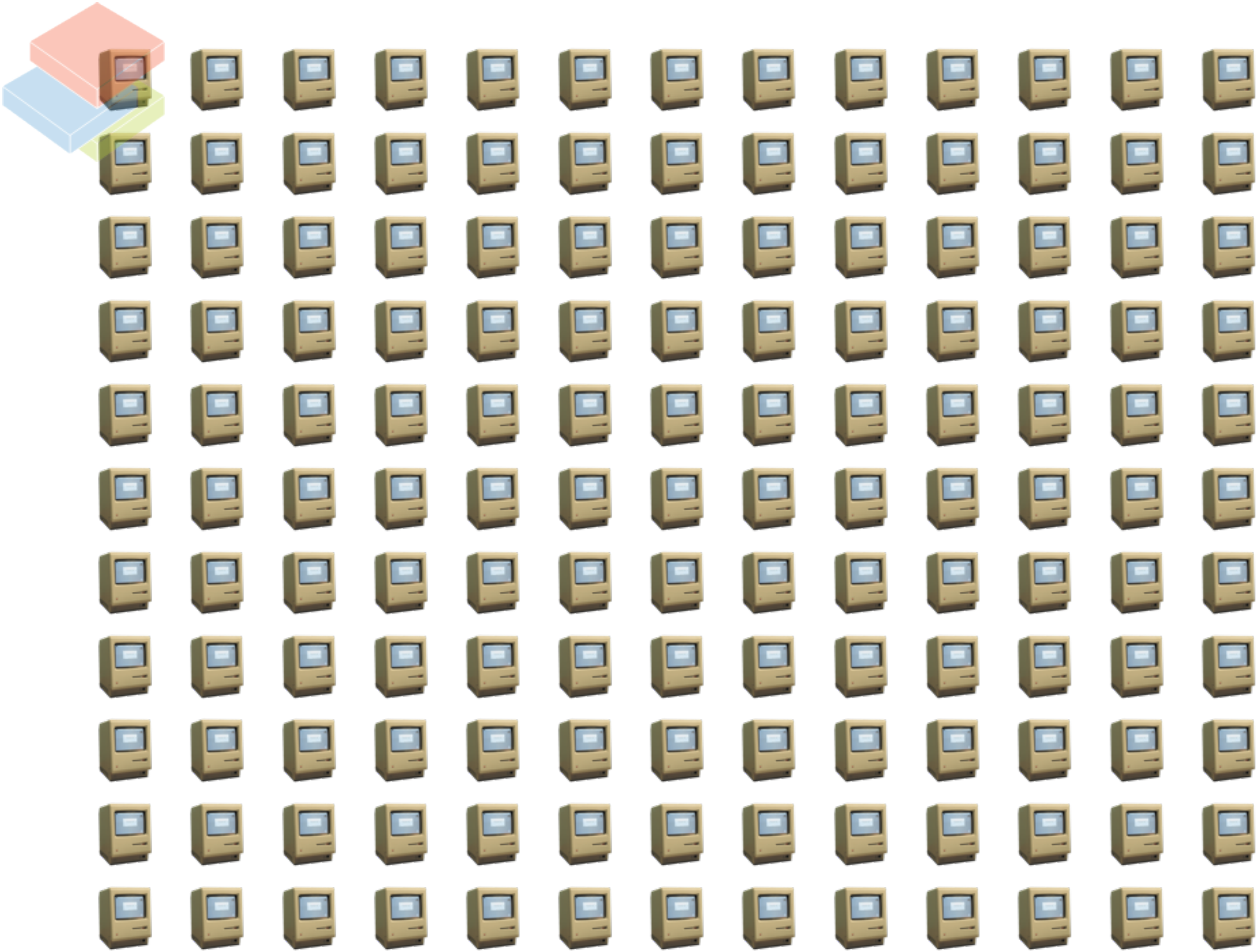
pay()

yet another person

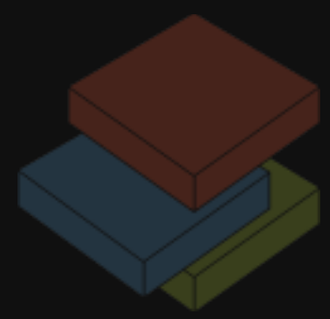


# Objects





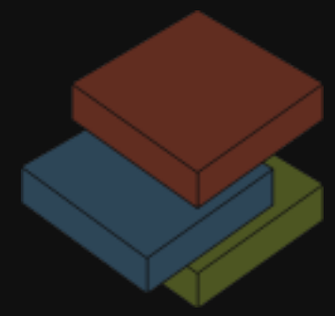
```
class Order
  # ...

  def pay
    gateway = Gateway.new
    gateway.could_u_please_charge(5)
  end
end
```



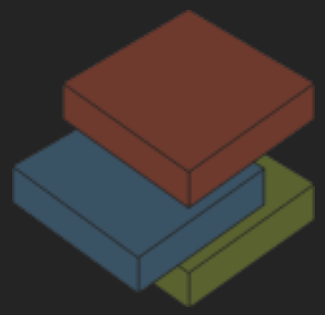
**This is a message!**

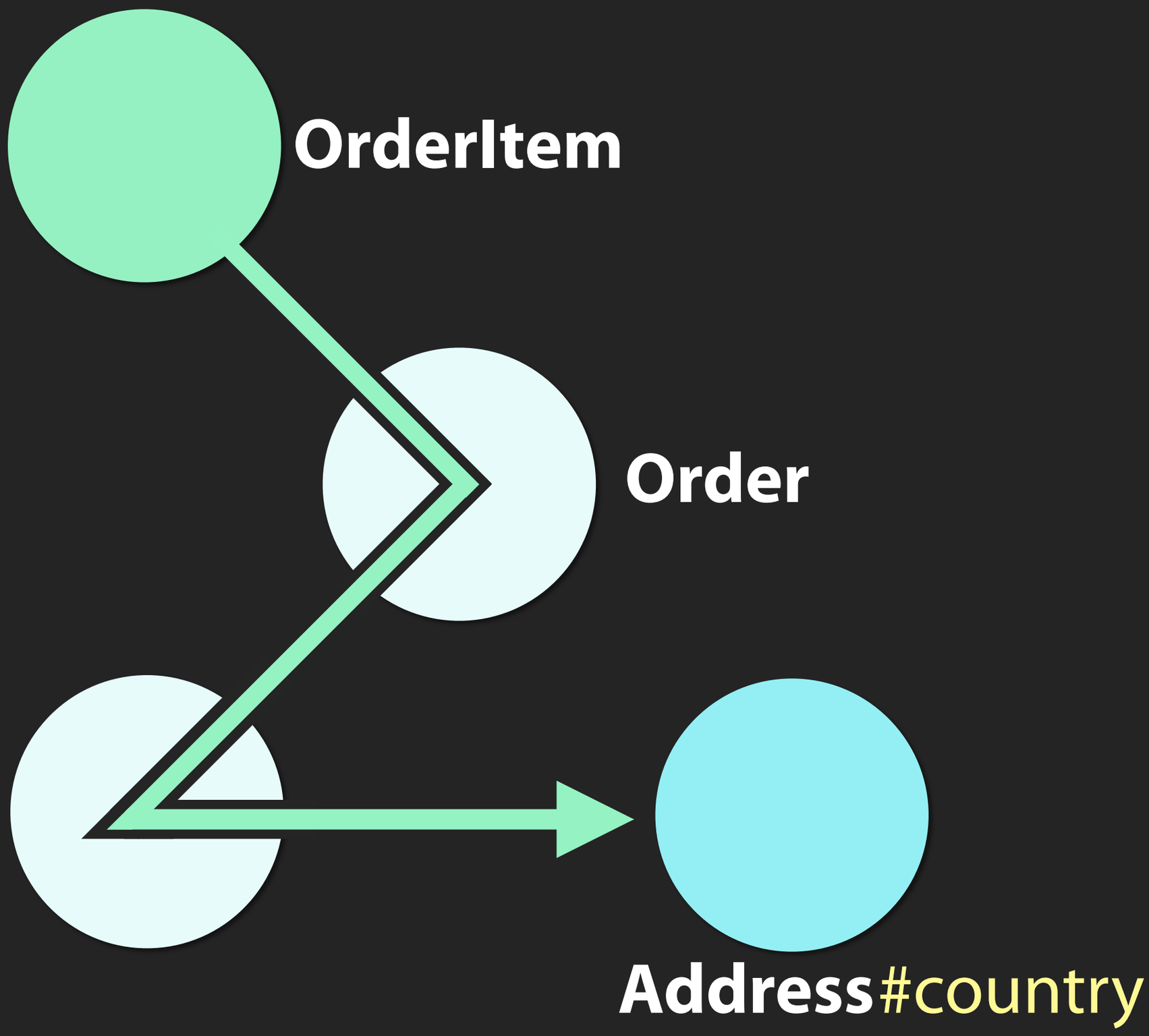
# Local Retention and Protection

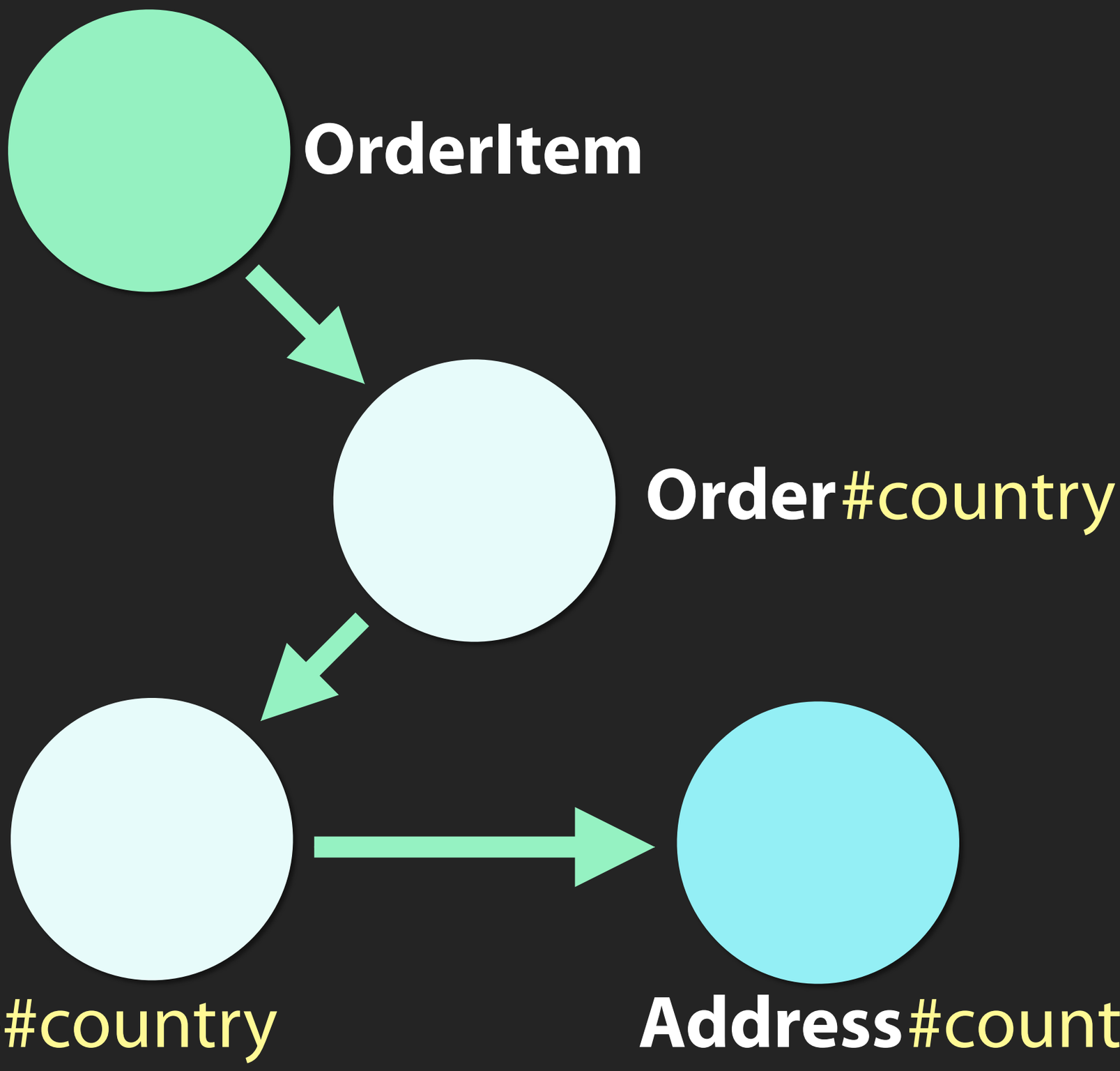
**Law of Demeter** or Principle of Least Knowledge

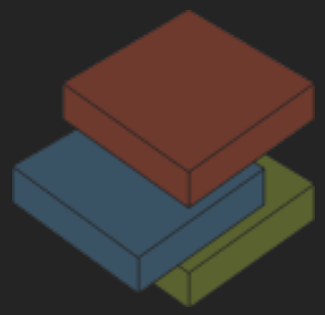


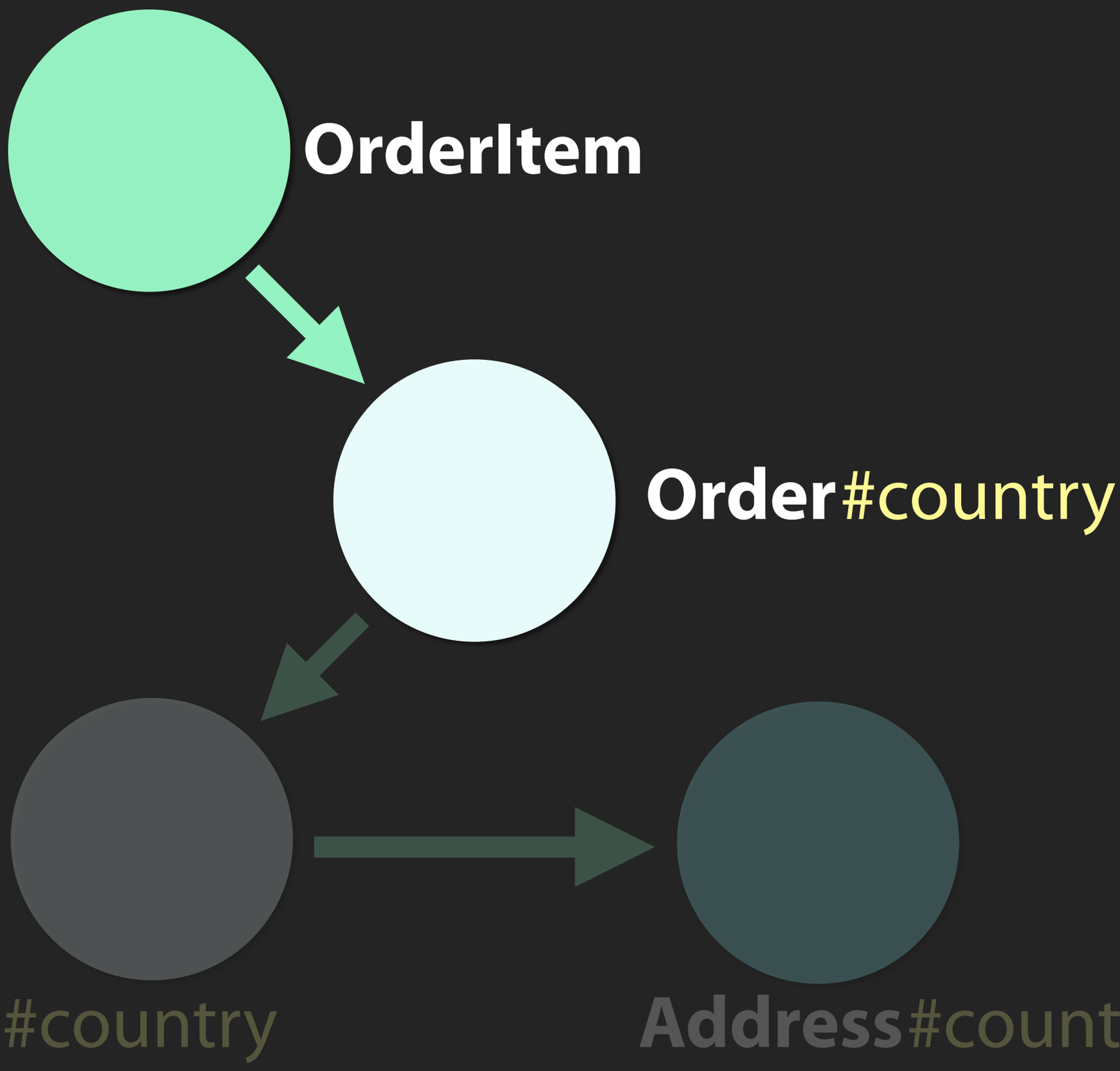
```
class OrderItem
  # ...

  def country
    if order.store.country.nil?
      “US”
    else
      order.store.address.country
    end
  end
end
```




OrderItem
Order
Store
Address#country

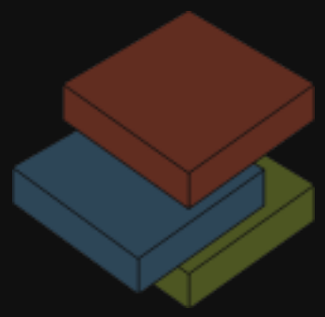

OrderItem
Order#country
Store#country
Address#country



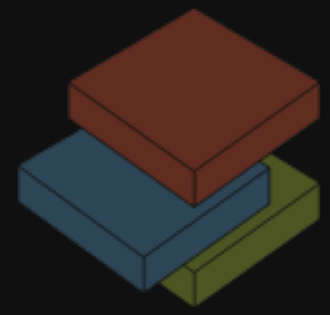

OrderItem
Order#country
Store#country
Address#country

# Information Hiding

## Encapsulation

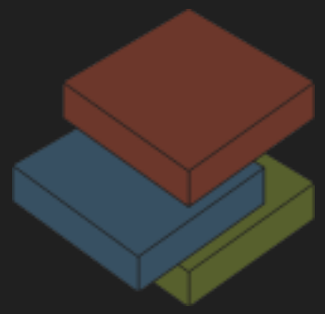


```ruby
class PaymentGateway
  attr_reader :options

  # ...
end
```

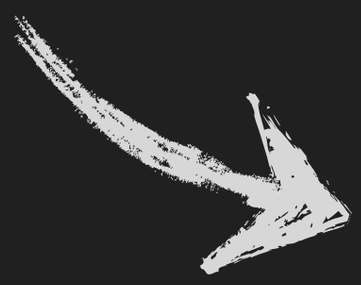


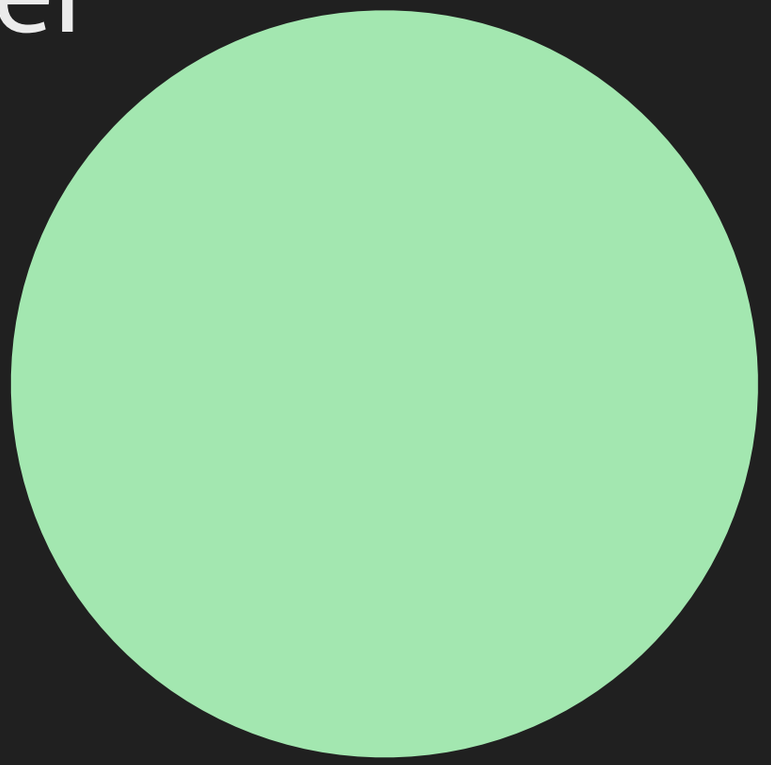
```
class PaymentGateway
  # ...

  private

  attr_reader :options
end
```

# Polymorphism

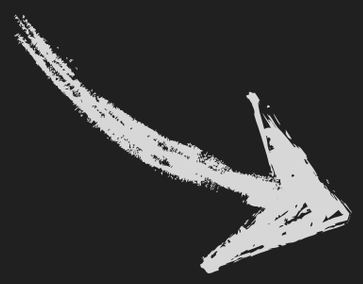


# Gateway wrapper



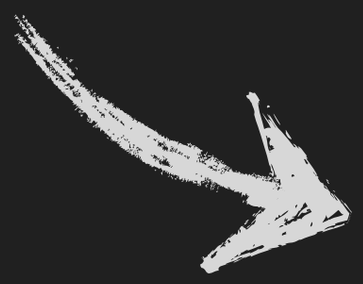


Gateway wrapper

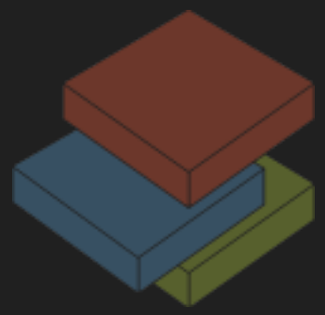


**#charge**
**#refund**

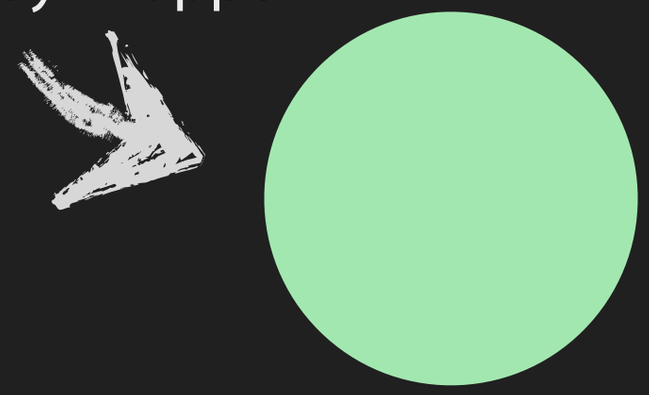
Gateway wrapper

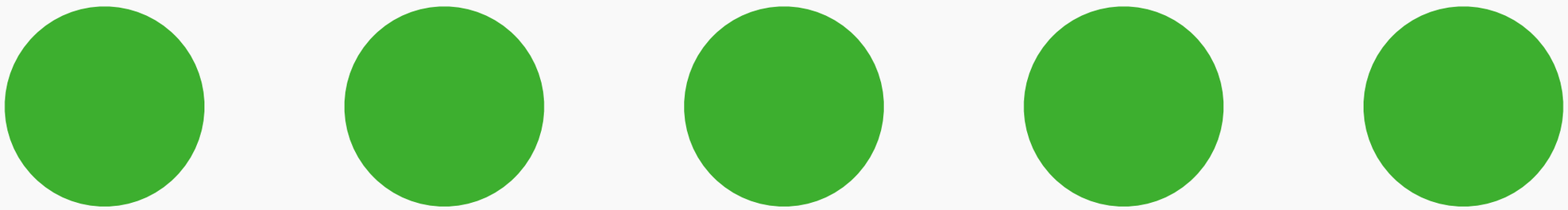


**#charge**
**#refund**

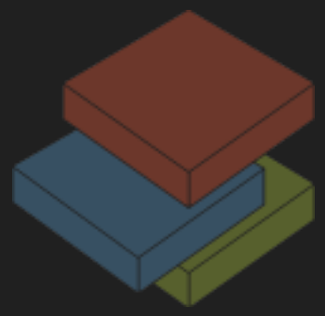


Gateway wrapper

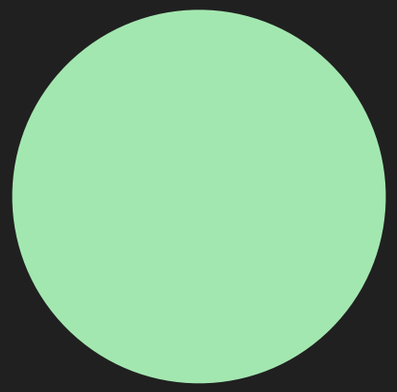


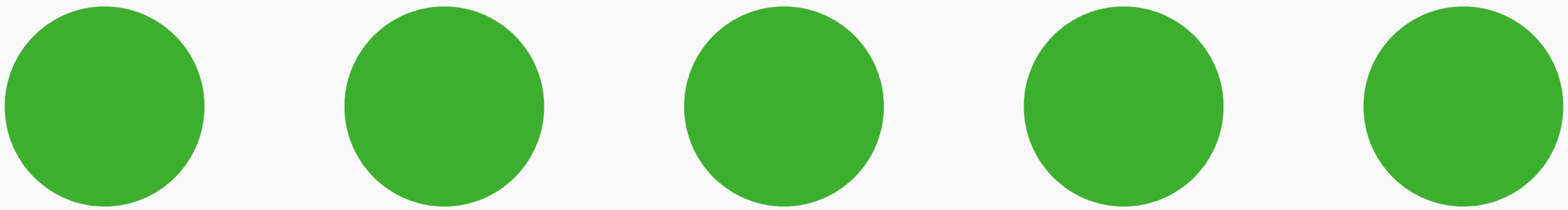


Processors



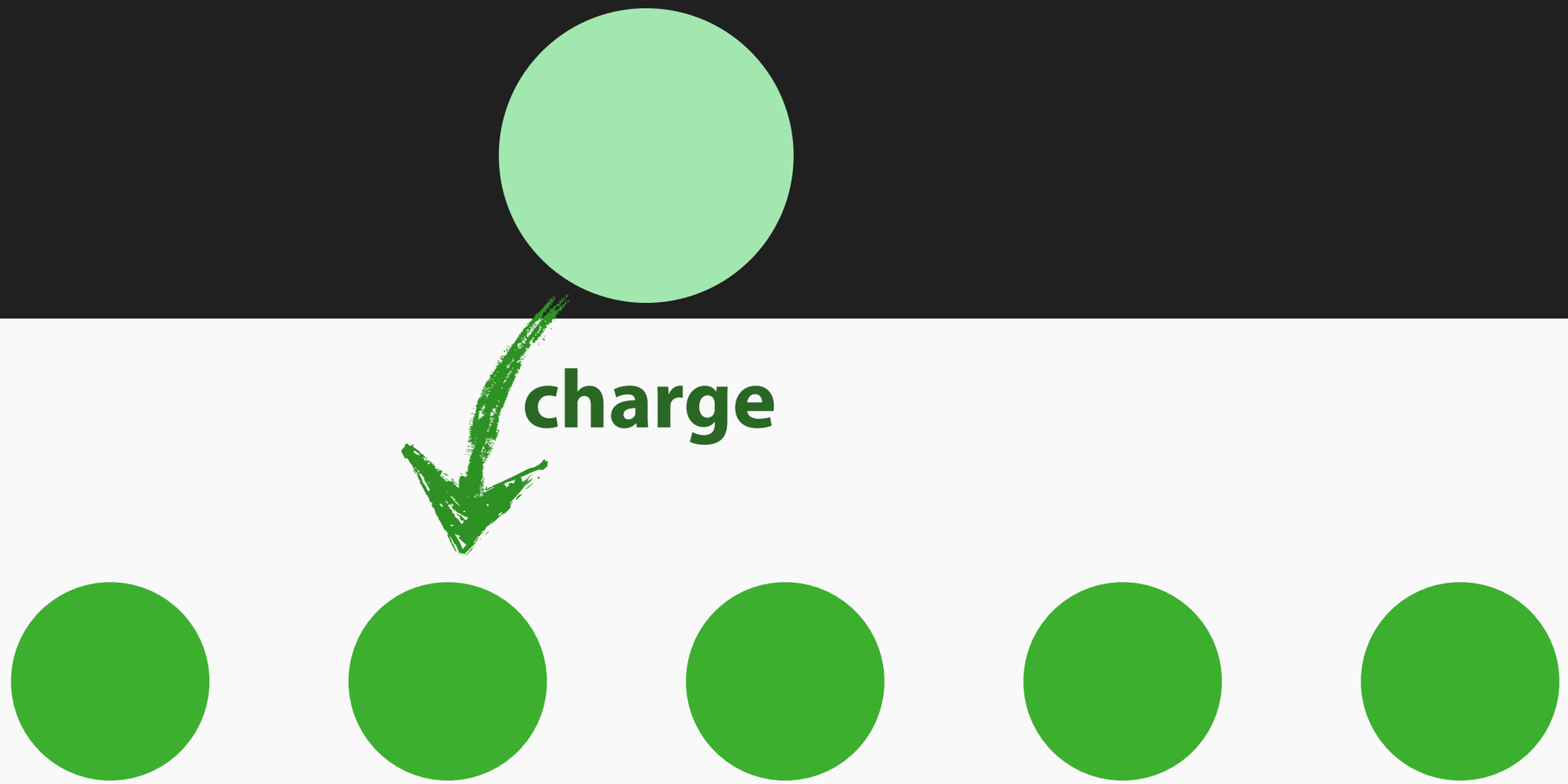
Gateway wrapper



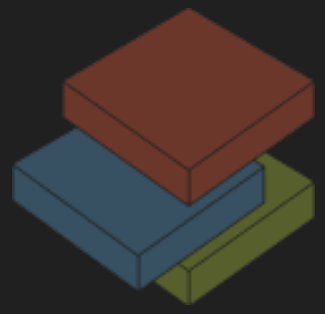


Processors

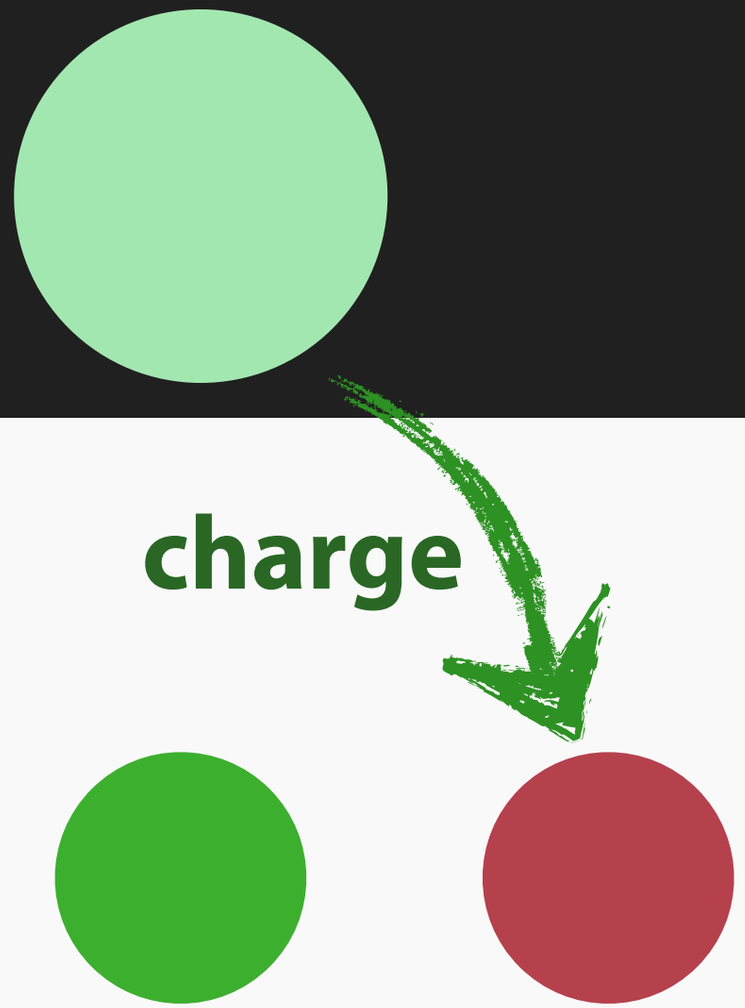
# Gateway wrapper




Processors



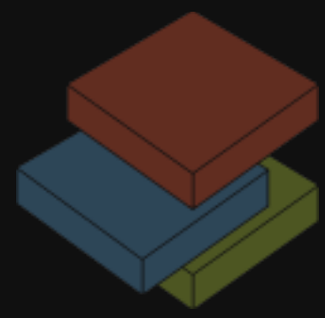
# Gateway wrapper



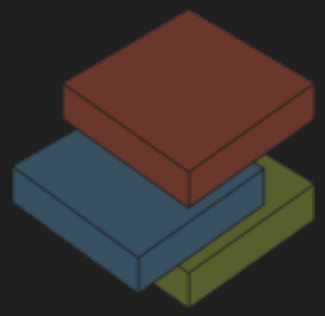

Processors

# Just works™

```ruby
class CanadaProcessor
  def charge(amount)
    # own XML builder
  end
end
```



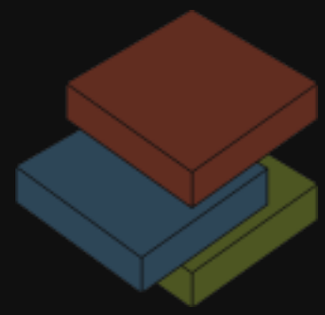
```ruby
class GreeceProcessor
  def charge(amount)
    # own XML builder
  end
end
```



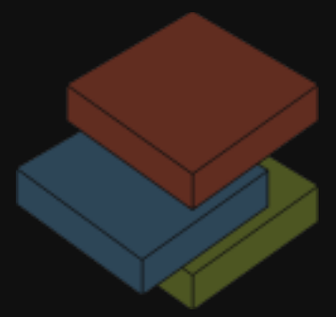
# Classes

# It’s about communication.

# What vs How

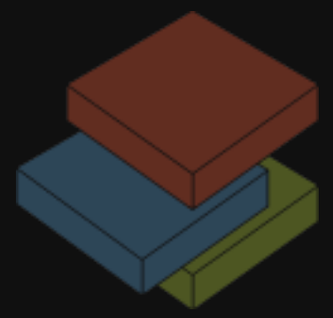


```
class PricingStrategy
  # ...

  def final_price
    @price              +
      (@price*0.1) +
      8.5
  end
end
```

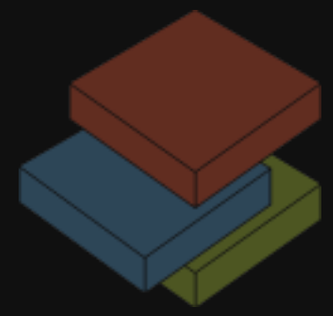


```ruby
class PricingStrategy
  # ...

  def final_price
    @price + commission + shipping
  end

  private

  def comission
    @price * 0.1
  end

  def shipping
    8.5
  end
end
```

# Private methods



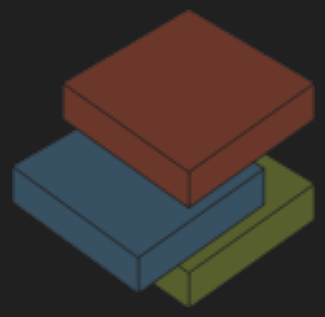
```ruby
class PricingStrategy

  private

  def commission
    @price * 0.1
  end
end

price = PricingStrategy.new(20)
price.send(:commission) # => 0.2
```



```ruby
class PaymentGateway
  def initialize(country)
    @country = country
  end

  def charge(amount)
    gateway.charge(amount)
  end

  private

  def gateway
    "#{@country}Processor".constantize.new
  end
end
```

# Write great code

# Communicate



# Welcome Objects

# Welcome Classes