The Abdus Salam International Centre for Theoretical Physics

Object-oriented design

David Grellscheid

UNIVERSITETET I BERGEN

The Abdus Salam International Centre for Theoretical Physics

The Abdus Salam International Centre for Theoretical Physics

Programming paradigm examples

Structured / Non-Structured

Declarative / Imperative

Procedural

Object-oriented

Functional

(Almost) any style can be implemented in any language

The Abdus Salam
International Centre
for Theoretical Physics

Grady Booch

"Object-oriented analysis and design"

2nd edition

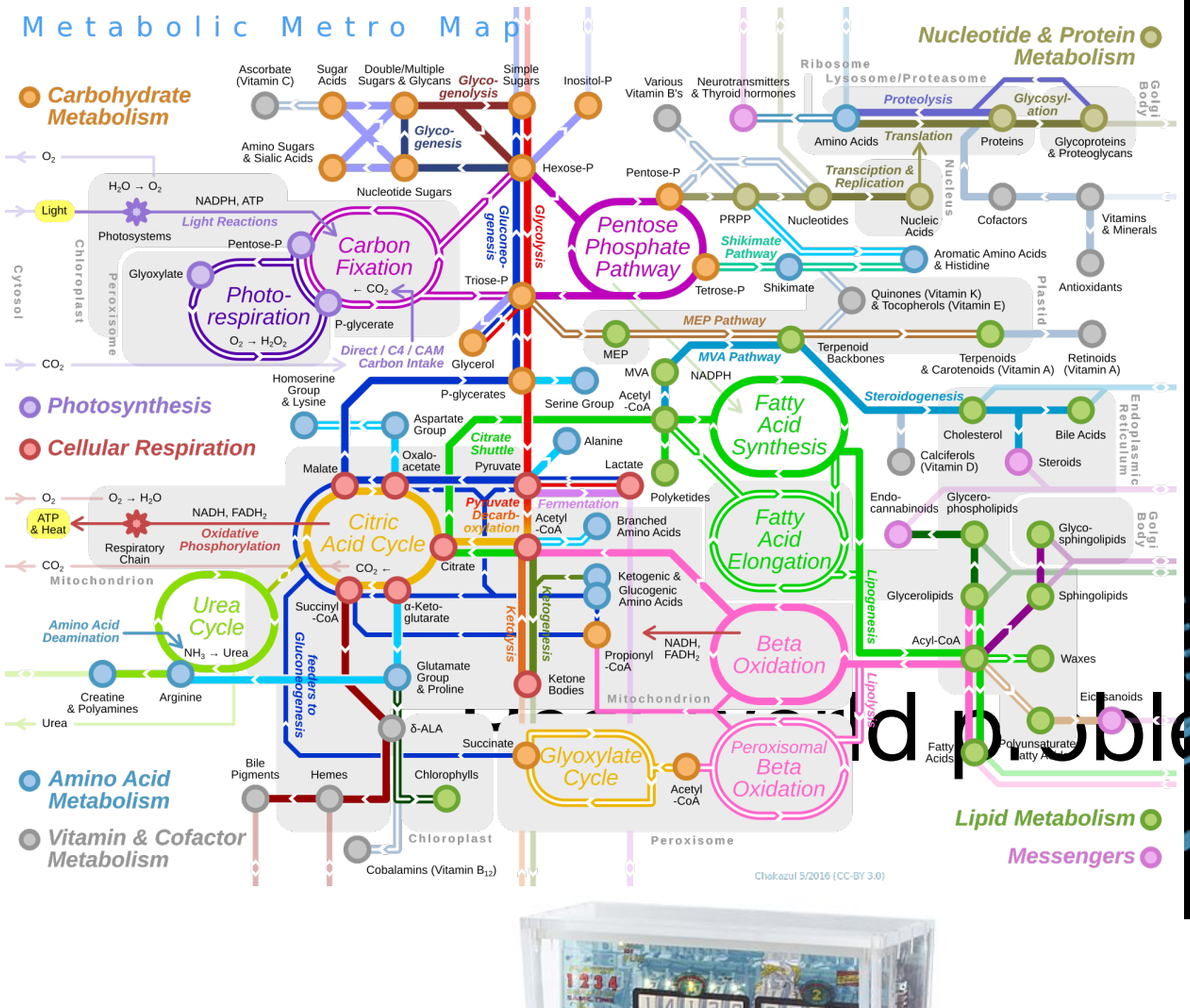
Addison-Wesley, 1994

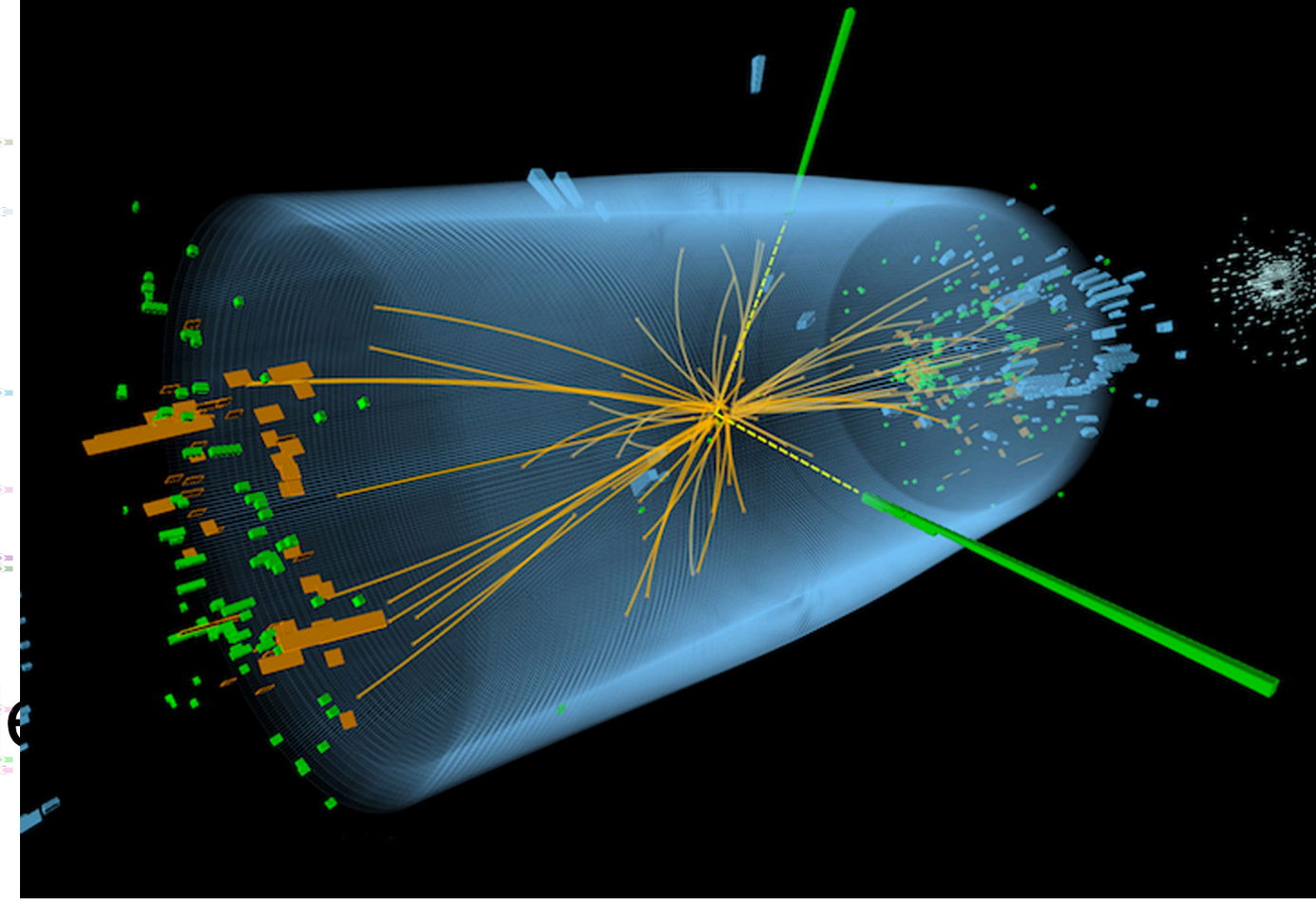
The Abdus Salam International Centre for Theoretical Physics

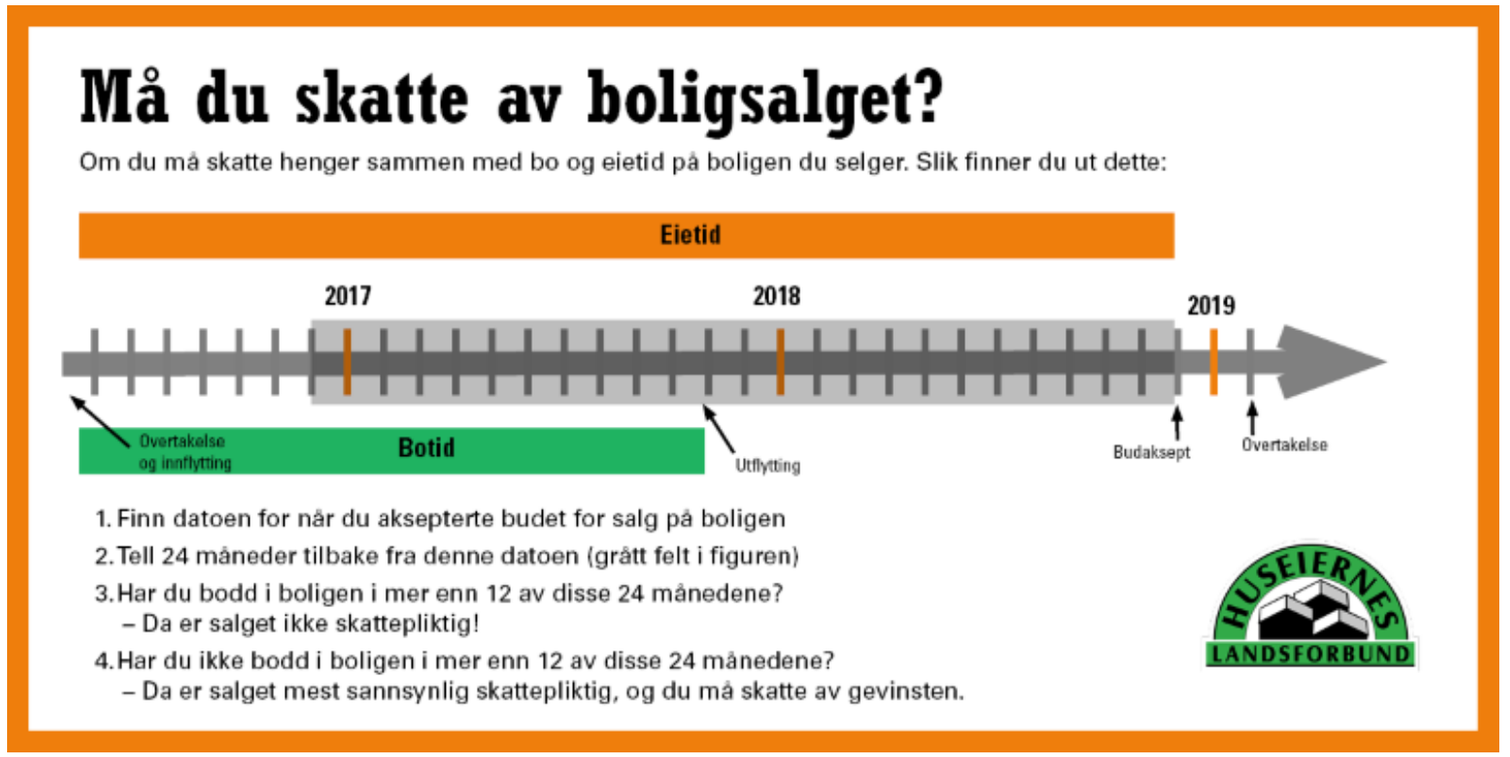
Main goal: manage complexity

Different approaches, OO is just one of them!

see e.g. Haskell for a completely different approach to complexity handling: functional programming


Metabolic Metro Map
Carbohydrate Metabolism
Photosynthesis
Cellular Respiration
Amino Acid Metabolism
Vitamin & Cofactor Metabolism
Nucleotide & Protein Metabolism
Lipid Metabolism
Messengers
Ascorbate (Vitamin C)
Sugar Acids
Double/Multiple Sugars & Glycans
Simple Sugars
Inositol-P
Amino Sugars & Sialic Acids
Nucleotide Sugars
Hexose-P
Pentose-P
Light
Photosystems
NADPH, ATP
Light Reactions
Carbon Fixation
Photo-respiration
Glyoxylate
Triose-P
P-glycerate
Glycerol
Pentose Phosphate Pathway
Glycolysis
Gluconeo-genesis
Glycogenolysis
Glycogenesis
Direct / C4 / CAM Carbon Intake
Homoserine Group & Lysine
Aspartate Group
Oxalo-acetate
Malate
Pyruvate
Citrate Shuttle
Alanine
Lactate
Fermentation
Polyketides
Serine Group
P-glycerates
Acetyl -CoA
Citric Acid Cycle
Citrate
Pyruvate Decarb-oxylation
Oxidative Phosphorylation
Respiratory Chain
ATP & Heat
Branched Amino Acids
Ketogenic & Glucogenic Amino Acids
Urea Cycle
Amino Acid Deamination
Succinyl -CoA
α-Keto-glutarate
Glutamate Group & Proline
Creatine & Polyamines
Arginine
Urea
Gluconeogenesis feeders to
Ketolysis
Ketogenesis
Propionyl -CoA
Ketone Bodies
Beta Oxidation
Mitochondrion
δ-ALA
Succinate
Glyoxylate Cycle
Peroxisomal Beta Oxidation
Bile Pigments
Hemes
Chlorophylls
Cobalamins (Vitamin B12)
Chloroplast
Peroxisome
Cytosol
Ribosome
Lysosome/Proteasome
Various Vitamin B's
Neurotransmitters & Thyroid hormones
Proteolysis
Glycosyl-ation
Amino Acids
Translation
Proteins
Glycoproteins & Proteoglycans
Golgi Body
Transciption & Replication
Nucleus
PRPP
Nucleotides
Nucleic Acids
Cofactors
Vitamins & Minerals
Shikimate Pathway
Aromatic Amino Acids & Histidine
Tetrose-P
Shikimate
Antioxidants
Quinones (Vitamin K) & Tocopherols (Vitamin E)
Plastid
MEP Pathway
MEP
MVA Pathway
MVA
NADPH
Terpenoid Backbones
Terpenoids & Carotenoids (Vitamin A)
Retinoids (Vitamin A)
Steroidogenesis
Cholesterol
Bile Acids
Calciferols (Vitamin D)
Steroids
Endoplasmic Reticulum
Fatty Acid Synthesis
Fatty Acid Elongation
Lipogenesis
Lipolysis
Endo-cannabinoids
Glycero-phospholipids
Glyco-sphingolipids
Glycerolipids
Sphingolipids
Acyl-CoA
Waxes
Eicosanoids
Fatty Acids
Polyunsaturated Fatty Acids
Chakazul 5/2016 (CC-BY 3.0)




Må du skatte av boligsalget?
Om du må skatte henger sammen med bo og eietid på boligen du selger. Slik finner du ut dette:
Eietid
2017
2018
2019
Overtakelse og innflytting
Botid
Utflytting
Budaksept
Overtakelse
1. Finn datoen for når du aksepterte budet for salg på boligen
2. Tell 24 måneder tilbake fra denne datoen (grått felt i figuren)
3. Har du bodd i boligen i mer enn 12 av disse 24 månedene?
– Da er salget ikke skattepliktig!
4. Har du ikke bodd i boligen i mer enn 12 av disse 24 månedene?
– Da er salget mest sannsynlig skattepliktig, og du må skatte av gevinsten.
HUSEIERNES LANDSFORBUND

The Abdus Salam International Centre for Theoretical Physics

* Complexity of the problem domain
  external; requires software maintenance, evolution, preservation

* Development process
  impossible for one developer to understand large projects completely

* Software is boundlessly flexible
  able to work at any level of abstraction; no fixed quality standards

* Behaviour of discrete system
  natural world physics is local and continuous
  program state is not: combinatoric, small change -> large effect

Metabolic Metro Map

Carbohydrate Metabolism

Ascorbate (Vitamin C)
Sugar Acids
Double/Multiple Sugars & Glycans
Glyco-genolysis
Simple Sugars
Inositol-P
Amino Sugars & Sialic Acids
Glyco-genesis
Nucleotide Sugars
Hexose-P
Glycolysis
Gluconeo-genesis
Triose-P
Glycerol
Pentose Phosphate Pathway
Pentose-P
Tetrose-P
Shikimate Pathway
Shikimate
Aromatic Amino Acids & Histidine
Quinones (Vitamin K) & Tocopherols (Vitamin E)
Antioxidants
Vitamins & Minerals
Cofactors
Various Vitamin B's
Neurotransmitters & Thyroid hormones
Ribosome
Lysosome/Proteasome
Proteolysis
Glycosyl-ation
Golgi Body
Amino Acids
Translation
Proteins
Glycoproteins & Proteoglycans
Transciption & Replication
Nucleus
PRPP
Nucleotides
Nucleic Acids
Plastid
MEP Pathway
MEP
Terpenoid Backbones
Terpenoids & Carotenoids (Vitamin A)
Retinoids (Vitamin A)
MVA Pathway
MVA
NADPH
Steroidogenesis
Cholesterol
Bile Acids
Calciferols (Vitamin D)
Steroids
Endoplasmic Reticulum
Fatty Acid Synthesis
Acetyl -CoA
Polyketides
Fatty Acid Elongation
Endo-cannabinoids
Glycero-phospholipids
Glyco-sphingolipids
Golgi Body
Lipogenesis
Glycerolipids
Sphingolipids
Acyl-CoA
Waxes
Eicosanoids
Polyunsaturated Fatty Acids
Fatty Acids
Lipolysis
Beta Oxidation
Propionyl -CoA
NADH, $FADH_2$
Ketone Bodies
Ketolysis
Ketogenesis
Ketogenic & Glucogenic Amino Acids
Mitochondrion
Peroxisomal Beta Oxidation
Glyoxylate Cycle
Acetyl -CoA
Succinate
Peroxisome

Photosynthesis

O_2
$H_2O \rightarrow O_2$
Light
Photosystems
NADPH, ATP
Light Reactions
Chloroplast
Cytosol
Peroxisome
Pentose-P
Carbon Fixation
$\leftarrow CO_2$
Glyoxylate
Photo-respiration
$O_2 \rightarrow H_2O_2$
P-glycerate
CO_2
Direct / C4 / CAM Carbon Intake

Cellular Respiration

O_2
$O_2 \rightarrow H_2O$
ATP & Heat
Respiratory Chain
$NADH, FADH_2$
Oxidative Phosphorylation
CO_2
Mitochondrion
Citric Acid Cycle
$CO_2 \leftarrow$
Citrate
Malate
Oxalo-acetate
Citrate Shuttle
Pyruvate
Pyruvate Decarb-oxylation
Fermentation
Acetyl -CoA
Alanine
Lactate
Branched Amino Acids
Succinyl -CoA
α-Keto-glutarate
Urea Cycle
Amino Acid Deamination
$NH_3 \rightarrow$ Urea
Creatine & Polyamines
Arginine
Urea
feeders to Gluconeogenesis
Glutamate Group & Proline
δ-ALA
Succinate
Bile Pigments
Hemes
Chlorophylls
Chloroplast
Cobalamins (Vitamin B_{12})

Amino Acid Metabolism

Homoserine Group & Lysine
Aspartate Group
P-glycerates
Serine Group

Vitamin & Cofactor Metabolism

Lipid Metabolism

Messengers

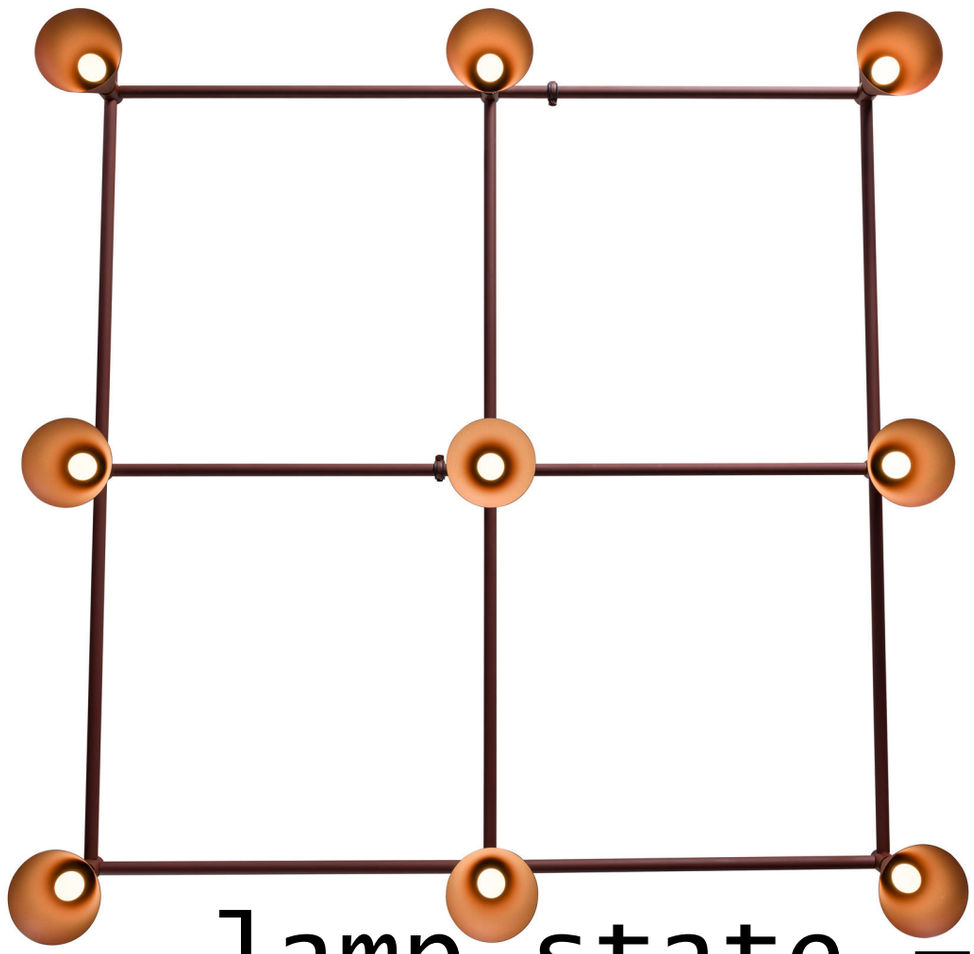
The Abdus Salam International Centre for Theoretical Physics

* Complexity is hierarchical grouping of subsystems, down to elementary components
* Choice of elementary blocks is mostly arbitrary
* Links and interactions within a component are much stronger than between components
* Hierarchy uses only a few different subsystems in different combinations
* Working complex systems evolve from working simple systems

The Abdus Salam International Centre for Theoretical Physics

* Deal with complexity by decomposition
* Algorithmic decomposition:
  which steps in which order?
* OO decomposition:
  which “real-world” entities are involved?
  how do they relate to each other?

Topic 1: object state



```
lamp_state = [0,1,1,0,0,0,1,1,1]
lamp_state = [[0,1,1],[0,0,0],[1,1,1]]
lamp_state[1][2] == 1
```

Also need two angles for the pointing direction:

```
thetas = [[0.3, 0.4, 0.5],[…],[…]]
  phis = [[0.7, 1.1, 0.0],[…],[…]]
```

Parallel lists are clumsy to use

3 arrays of values

```
lamp_state = [[ 0, 1, 1],[…],[…]]
    thetas = [[0.3, 0.4, 0.5],[…],[…]]
      phis = [[0.7, 1.1, 0.0],[…],[…]]
```

One possible solution: group the other way

```
lamps = [[ , , ],[ , , ],[ , , ]]
```

an array of lamps

Now, ▮ represents one lamp

Class

```
Lamp

    -is_on
    -theta
    -phi
```

attributes / member variables

Constructor

```
is_on = 1
theta = 0.1
phi = 0.7
```

object

```
is_on = 0
theta = 0.3
phi = 0.1
```

object

```
is_on = 1
theta = 0.2
phi = 0.37
```

object

```
is_on = 1
theta = 0.9
phi = 1.9
```

object

...

```
class Lamp:
    def __init__(self, on=0, th=0, ph=0):
        self.is_on = on
        self.theta = th
        self.phi = ph
```

Constructor

attributes

objects

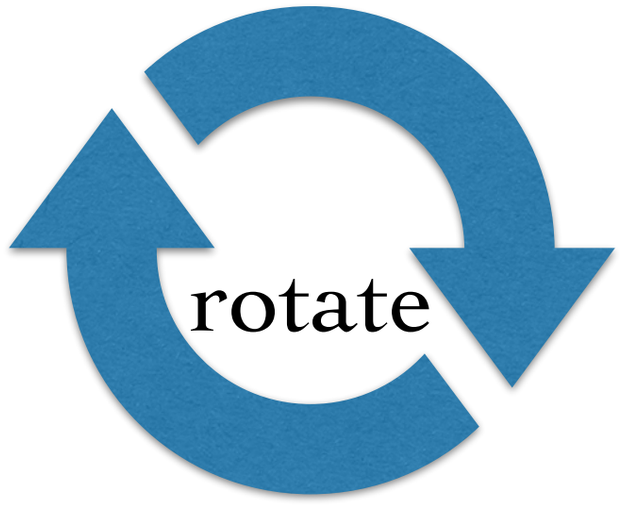
```
l1 = Lamp(1, 0.4, 0.7)
l2 = Lamp(0, 1.1, 0.3)

print(l1.is_on)
print(l2.is_on)

print(l2.theta)
```

Topic 2: object behaviour

We created objects with internal state.
What about lamp behaviours?



```
class Lamp:
    def __init__(self, on=0, th=0, ph=0):
        self.is_on = on
        self.theta = th
        self.phi = ph

    def turn_on(self):
        self.is_on = 1

    def turn_off(self):
        self.is_on = 0

    def rotate(self,angle):
        self.phi += angle
```

Constructor

attributes

methods / member functions

Object methods allow us to use language from the problem domain rather than basic types:

```
lamp_A = Lamp(1, 0.4, 0.7)

lamp_A.turn_off()
lamp_A.rotate(0.2)
```

The Abdus Salam International Centre for Theoretical Physics

Object

* **State**: inner structure with current values
* **Behaviour**: external interaction and state changes (construct / destruct // modify / select / iterate)
* **Identity:** distinct to all other objects

  It's not the name, one object can have many names!
  Identity considerations are relevant when looking at copying, lifetime and ownership behaviour.

The Abdus Salam International Centre for Theoretical Physics

Class

Objects with common structure and behaviour belong to a **class**. The class defines both.

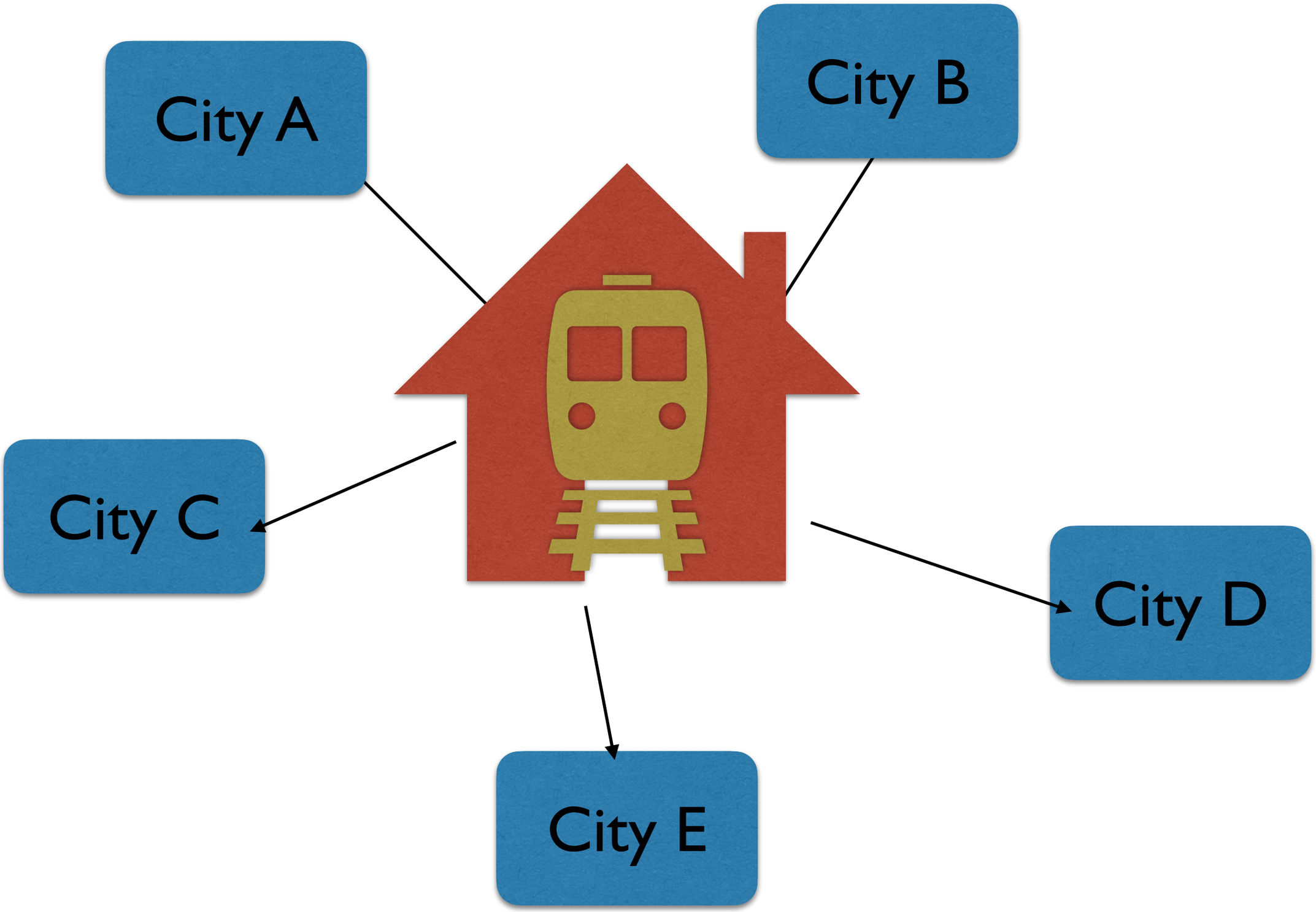
An object is an **instance** of a class.

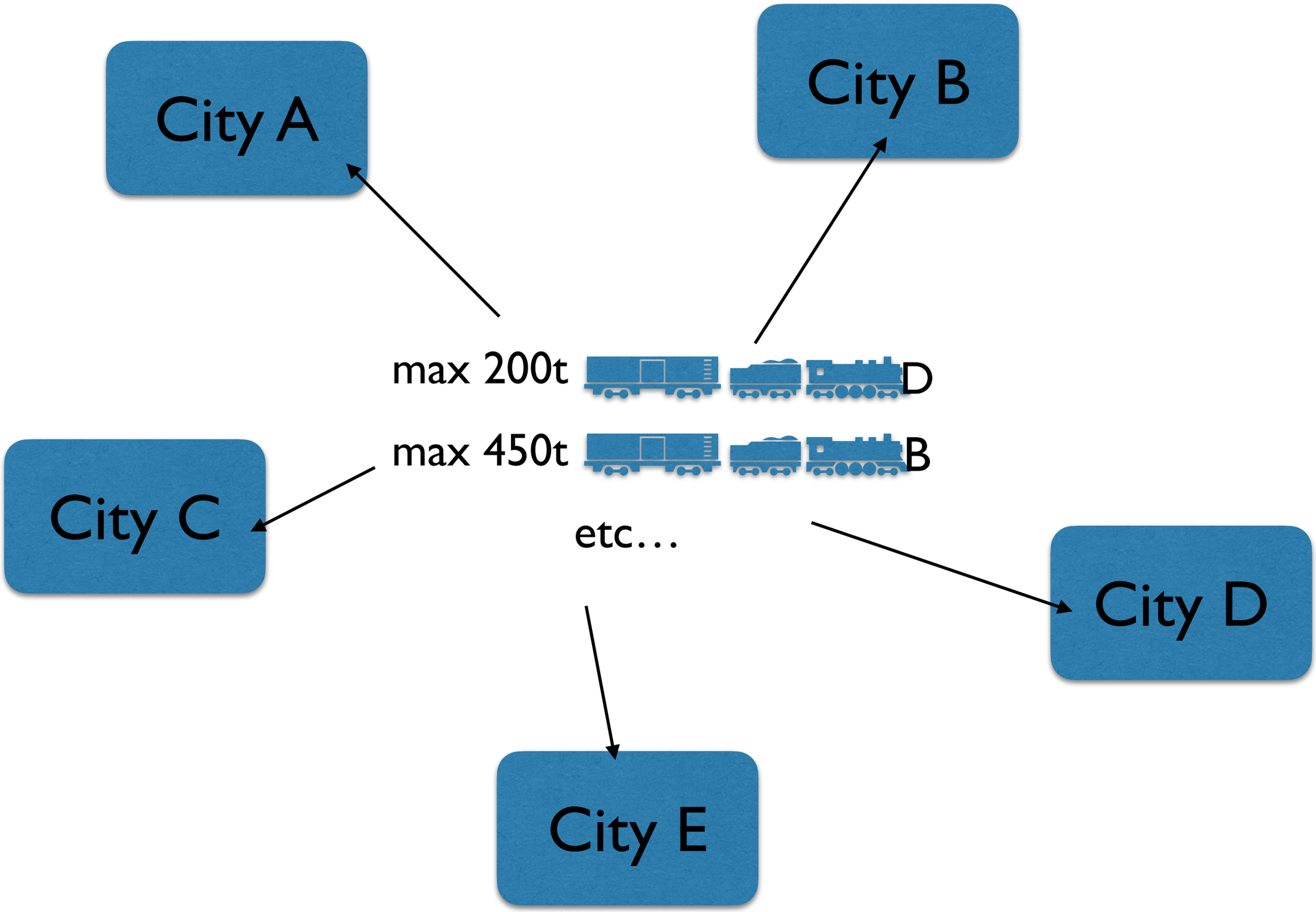
The Abdus Salam
International Centre
for Theoretical Physics

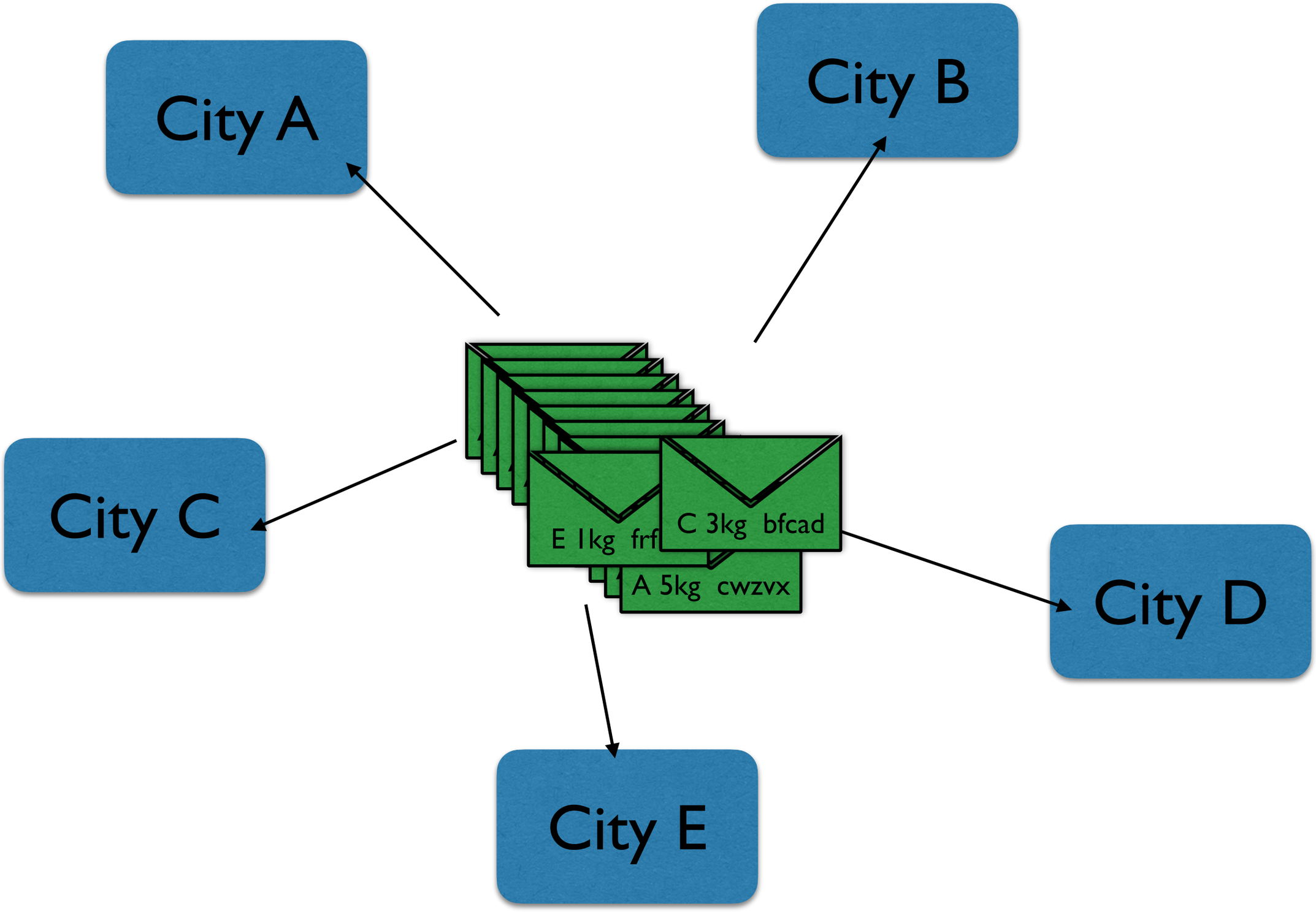
Exercise

OO design, David Grellscheid 2022-11-29

Exercise: a freight station




City A
City B
max 200t
D
max 450t
B
City C
etc…
City D
City E


City A
City B
City C
City D
City E
E 1kg frf
C 3kg bfcad
A 5kg cwzvx

The Abdus Salam International Centre for Theoretical Physics

Design an OO model for the station

classes, objects, interfaces, public/private, which methods/state

but no implementation!

a random train arrives, is loaded with correct mail, leaves, and repeat