LBTM X CSCE Club

LBTM Seminar Series

UNT UNIVERSITY UNION ROOM 268
2/27/24 6PM

Luka Bostick

The University of North Texas
LukaBostick@my.unt.edu
https://github.com/LBTM-Luka/LBTM-X-CSCE-2.27.24
github.com/LukaBostick

Materials To Complete this Talk

Overview

- Introduction
- Object-Oriented Programming (oop)
  - Procedural Vs. Object Orientated (vs. Functional/Procedural)
  - Object-Oriented Fundamentals
  - Encapsulation
  - Polymorphism
  - Abstraction
  - Relationships
    - Generic Association (or just Association)
    - Aggregation
    - Composition
    - Inheritance or ancestry
- Live Demo

Procedural Vs Object Orientated[1]

`OOP Pro`	Divide and Conquer
`OOP Pro`	Code Reuse
`OOP Pro`	Maintenance and Adaptability

`OOP Con`	Inefficiencies
`OOP Con`	OOP is easy to learn; but hard to master
`OOP Con`	It takes time and a bit of trust to learn OOP

Keep In Mind Many students are only briefly introduced to OOP, but few see the advantages when combined with standardized design principles.

[1] Programming Fundamentals II
https://hb2504.dcccd.edu/Syllabi/2024SP-COSC-1437-82004.pdf

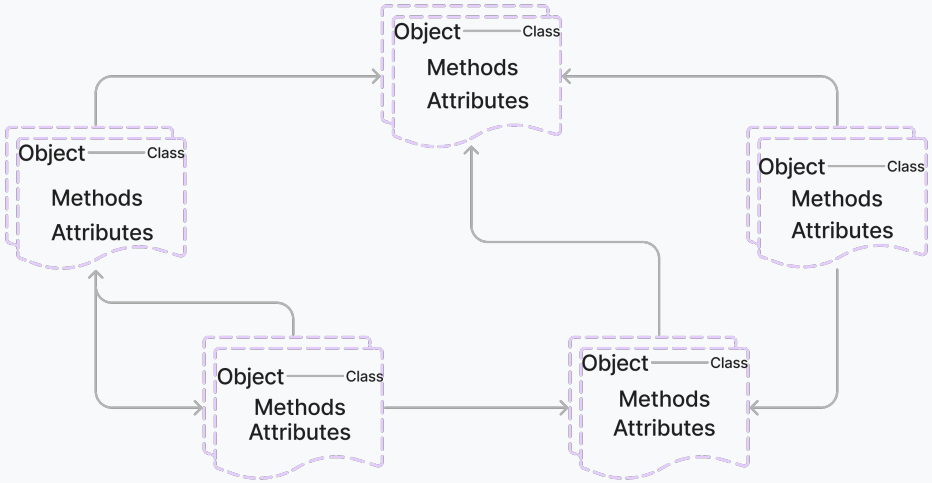
Object Oriented Fundamentals[2]

1. Objects
2. Classes
3. Methods
4. Attributes

[2] Structure of OOP
www.freecodecamp.org/news/what-is-object-oriented-programming/

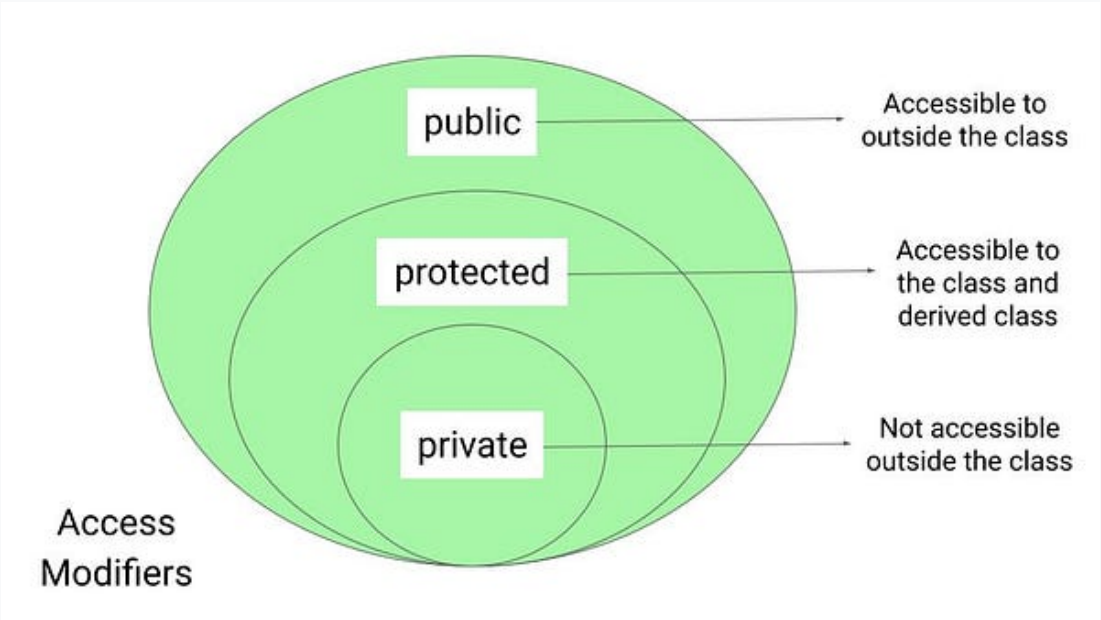
Object Oriented Fundamentals Cont

1. Objects → **Instance of a class**
2. Classes → **Blueprints of objects attributes and methods**
3. Methods→ **The behavior of an object**
4. Attributes→ **The state of an object**



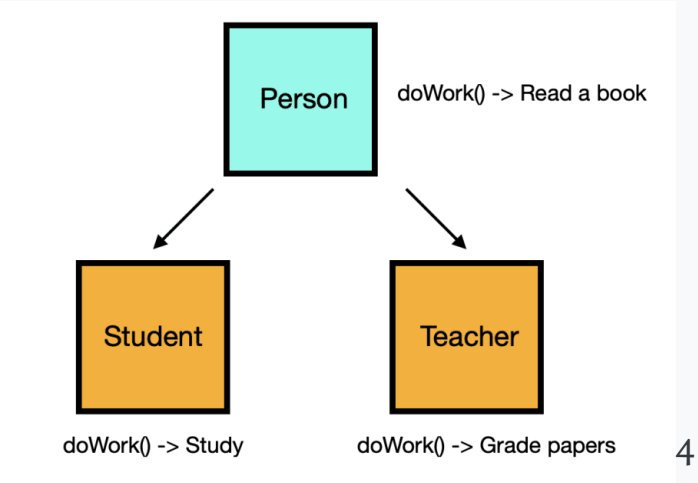
OOP Focuses | Encapsulation

- Objects bind the behaviors (methods/functions) together with the data members
- A singular object is only responsible for a small portion
- Each object protects its interests.



OOP Focuses | Polymorphism

- Objects can be repeated as needed,
  - One object handles all of the students
  - Meaning you would make a simpler/smaller student object.
  - Each student object can more easily be protected.
- This comes in handy when objects are similar (ancestors of each other) but not exact.



[4]

[4] https://dev.to/jburroughs/stay-in-the-l-oop-with-object-oriented-programming-basics-1dn

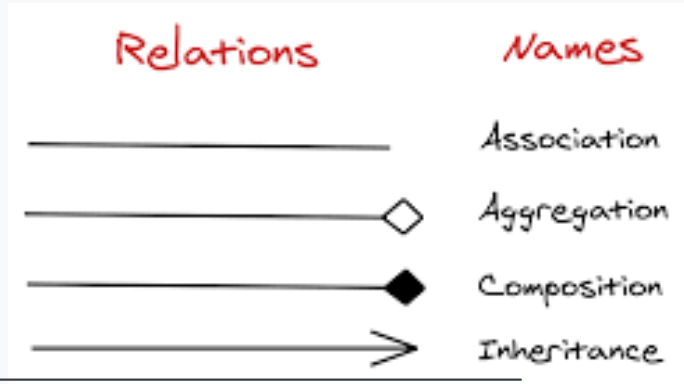
OOP Focuses | Relationships

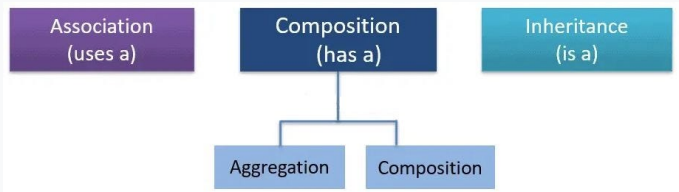
- Objects can work together by defining their relationships to each other
  - An object may create another instant
  - An object can store a set of objects
  - There are four main relationships in Java
    1. Generic Association (or just Association)
    2. Aggregation
    3. Composition
    4. Inheritance or Ancestry

OOP Focuses | Relationships

1. Generic Association (or just Association)
2. Aggregation
3. Composition
4. Inheritance or Ancestry

5 6



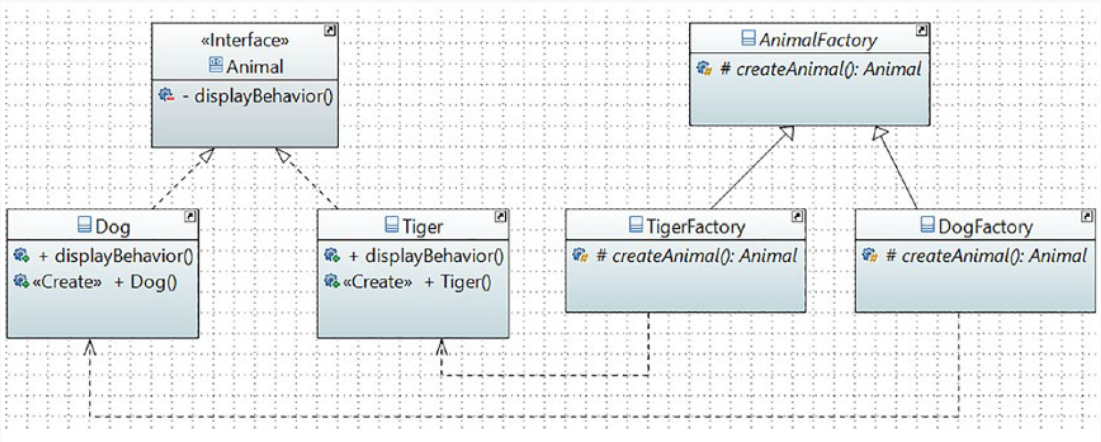


[5] dev.to/jburroughs/stay-in-the-l-oop-with-object-oriented-programming-basics-1dn

[6] https://medium.com/@esrasahince/oop-relations-between-objects-association-composition-inheritance-eab3df6afcbc

Switching gears to our ***DESIGN PATTERN DEMO***.

Demo: Factory Method Pattern



7

[7] Vaskaran Sarcar 2022 V. Sarcar, Java Design Patterns, https://doi.org/10.1007/978-1-4842-7971-7_3

Demo: Bridge Pattern

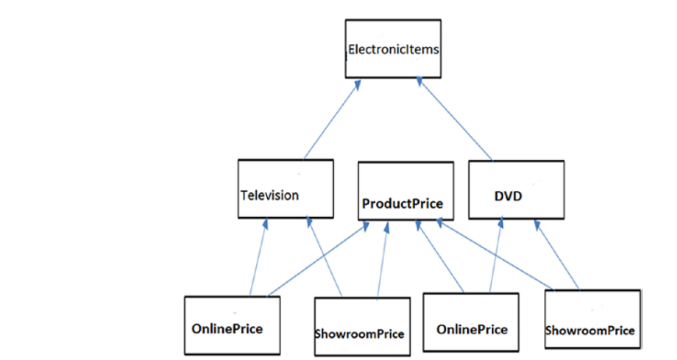


Figure 14-1. *Approach 1*

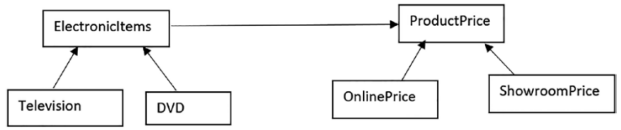


Figure 14-3. *Maintaining two separate hierarchies using the Bridge pattern*

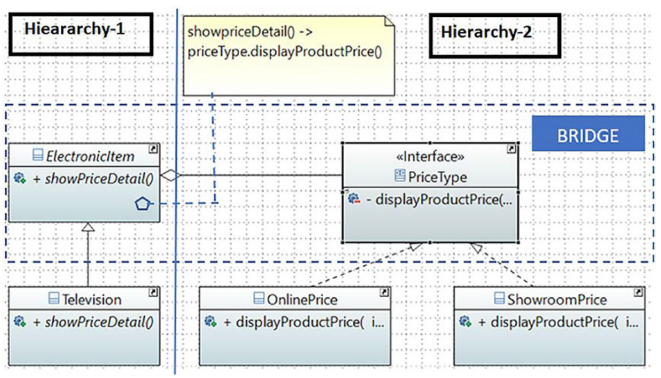
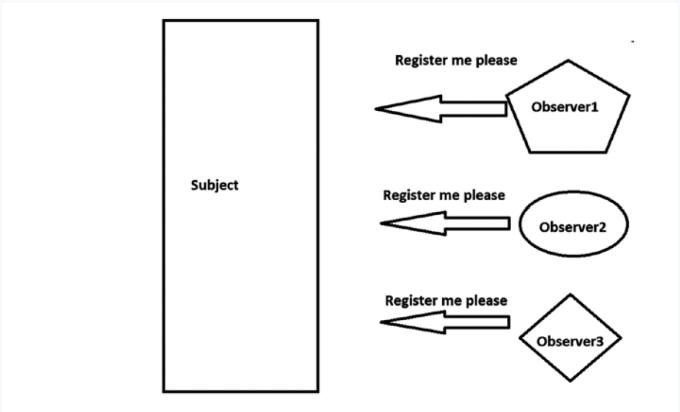
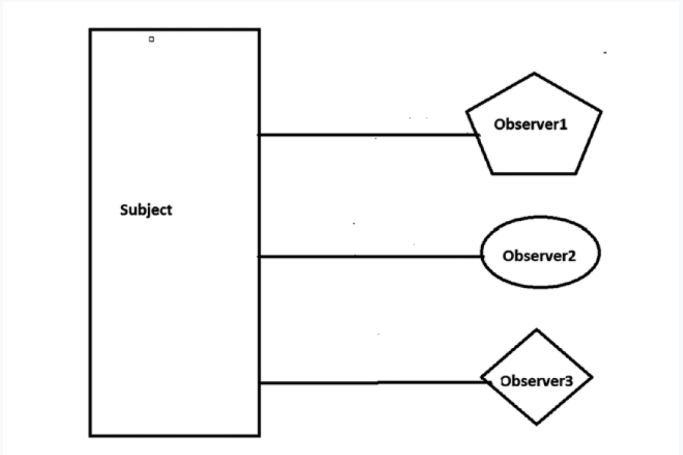
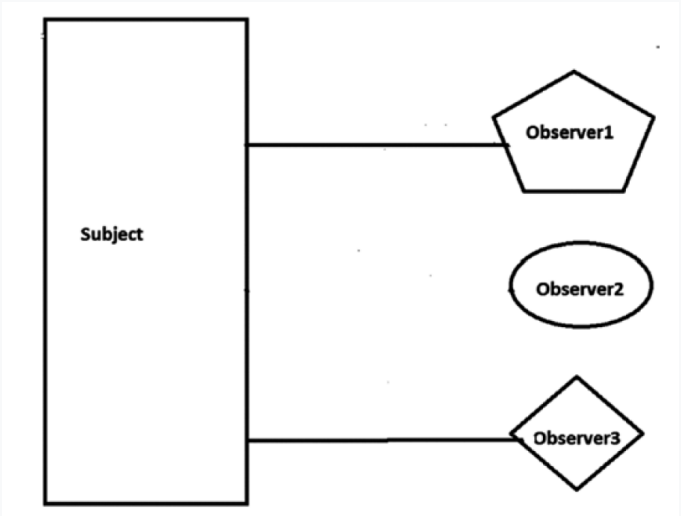


Figure 14-5. *Class diagram*

Demo| Observer Pattern



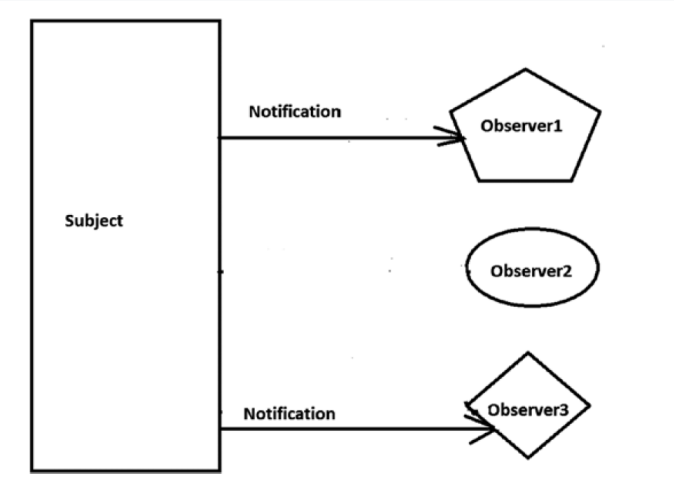


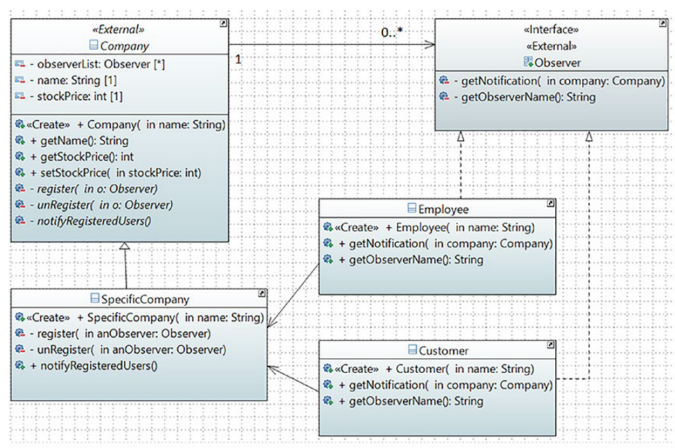


9

[9]Vaskaran Sarcar 2022 V. Sarcar, Java Design Patterns, https://doi.org/10.1007/978-1-4842-7971-7_16

Demo| Observer Pattern





[10] Vaskaran Sarcar 2022 V. Sarcar, Java Design Patterns, https://doi.org/10.1007/978-1-4842-7971-7_16

I want to thank Land Sea, and Sky and my academic review committee. ,
Without your support, the Work we do at LBTM would be nonexistent

LAND SEA & SKY

L B T M

https://github.com/lukaBostick https://www.linkedin.com/in/luka-bostick-379168244/