CIT 103 & CIT 104

Object Oriented Programming

By

Md. Palash Uddin

Lecturer

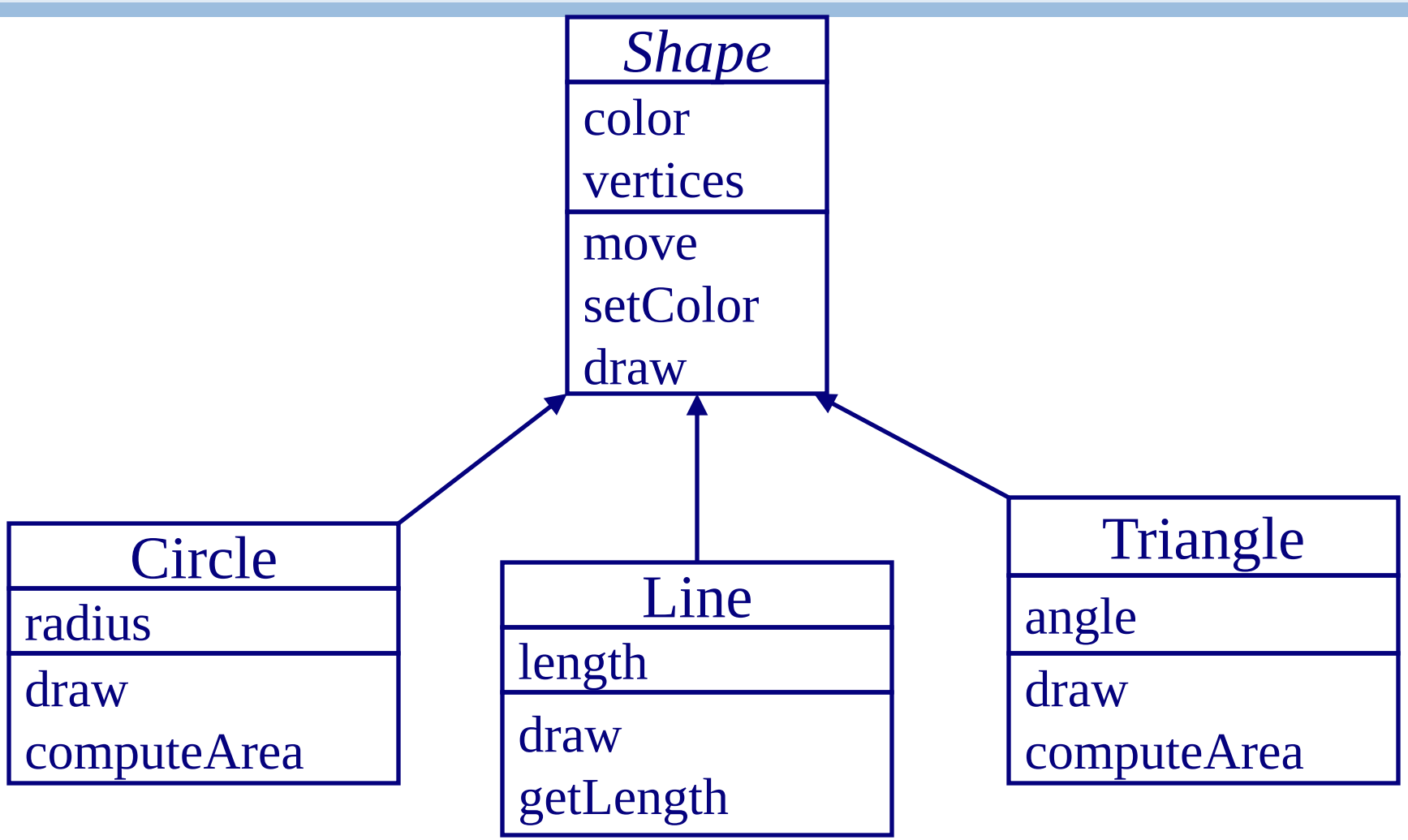
Dept. of CIT

Hajee Mohammad Danesh Science and Technology University, Dinajpur.

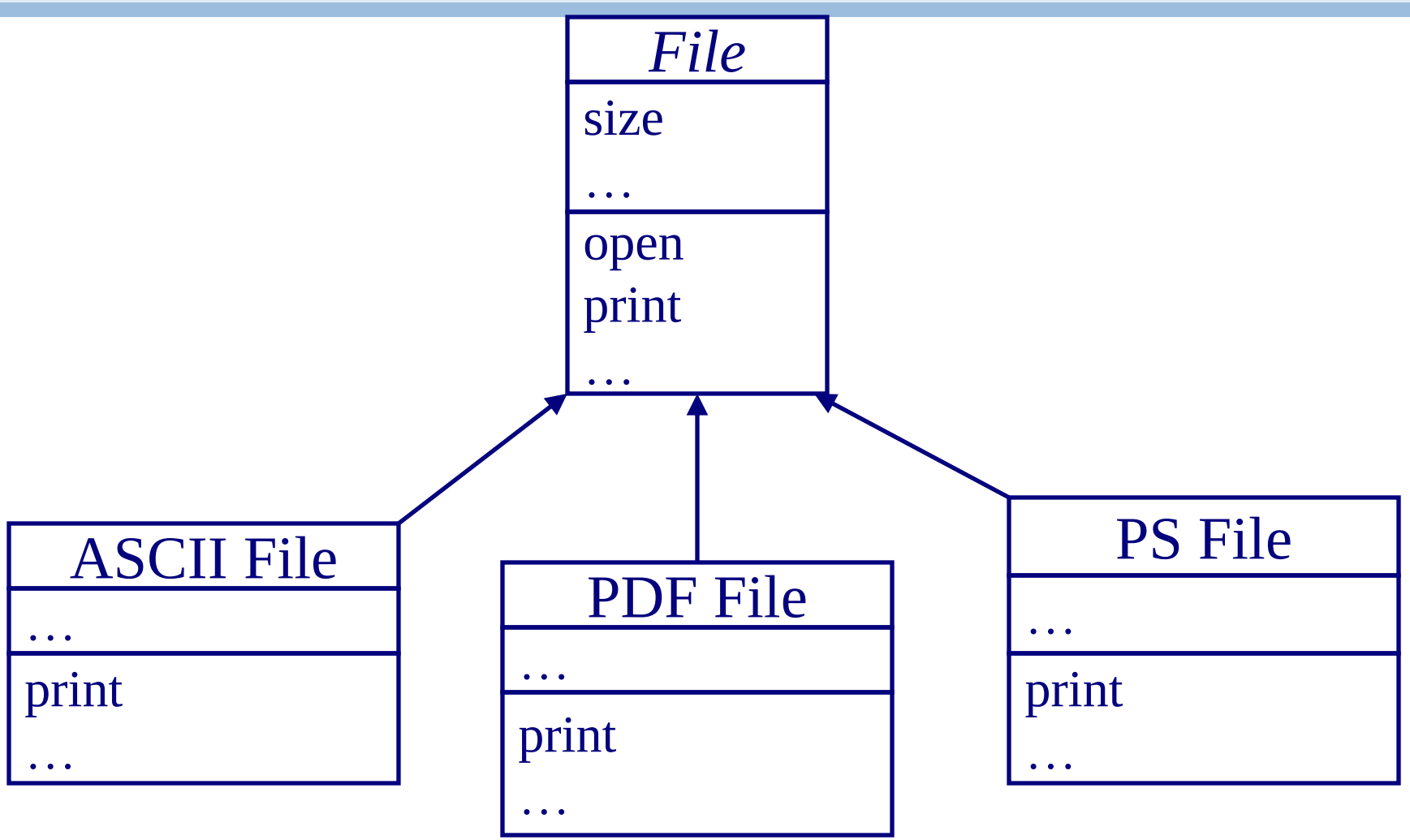
Class Compatibility

- A class is behaviorally **compatible with another** if it supports all the operations of the other class
- Such a class is called **subtype**
- A class can be replaced by its **subtype**
- **Derived class** is usually a **subtype** of the **base class**
- It can handle all the **legal messages** (operations) of the **base class**
- Therefore, **base class** can always be **replaced** by the **derived class**

Example – Class Compatibility



Example – Class Compatibility



Polymorphism

- In general, polymorphism refers to **existence of different forms of a single entity**
- For example, both **Diamond** and **Coal** are different forms of **Carbon**

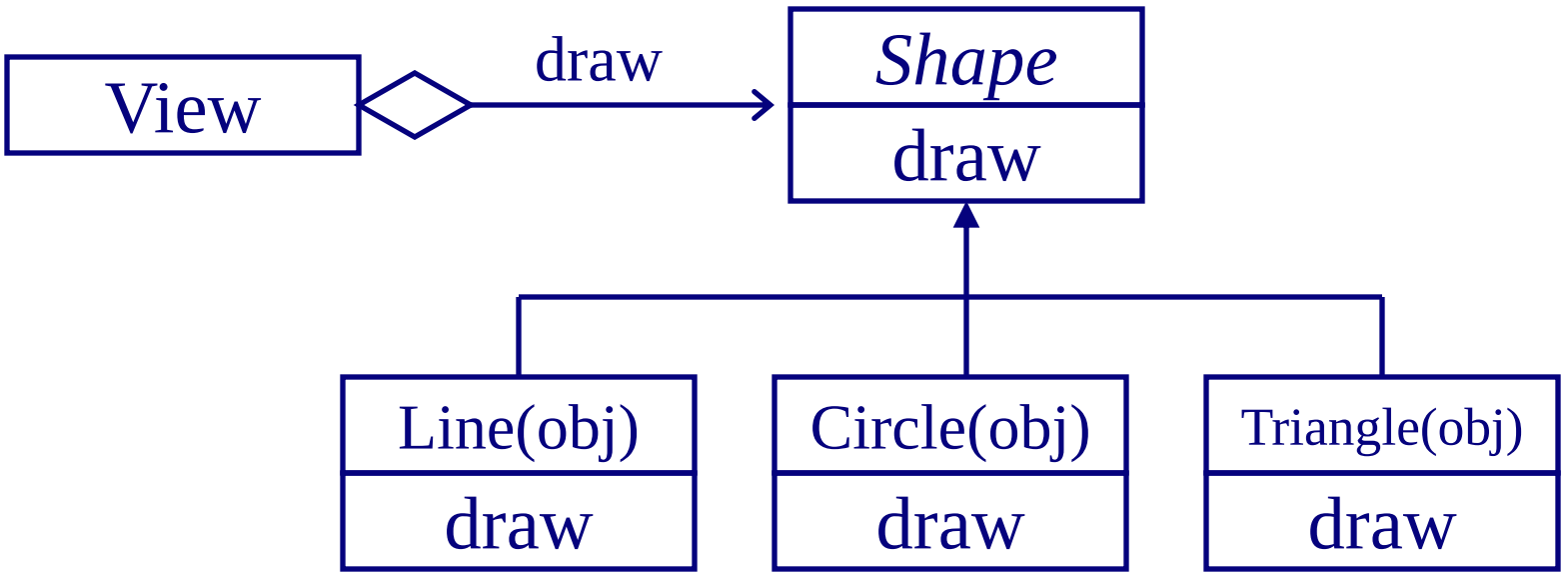
Polymorphism in OO Model

- In OO model, polymorphism means that **different objects** can behave in **different ways** for the **same message** (stimulus)
- Consequently, sender of a message does not need to know exact class of the receiver
- Also termed as **Overloading**

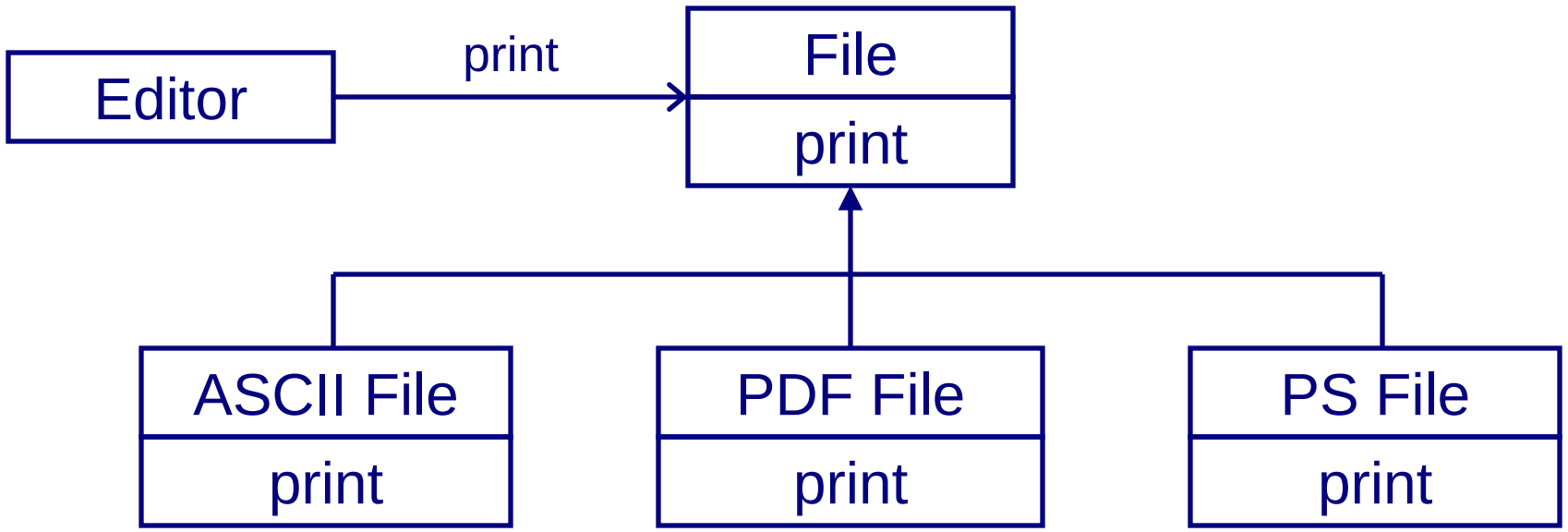
Polymorphism in OO Model

- **Two Types:**
- **Operator overloading**
  - An **operation** may exhibit **different behaviors** in **different instances**
  - The **behavior depends** upon the **types of data used** in the operation
  - For example, consider the operation of **addition**. For **two numbers**, the operation will generate a **sum**. If the operands are strings, the operation would produce a third string by **concatenation**
- **Function overloading**
  - Using a single **function name** to perform **different types of tasks** depending on the **different number and different types of arguments**

Example – Polymorphism

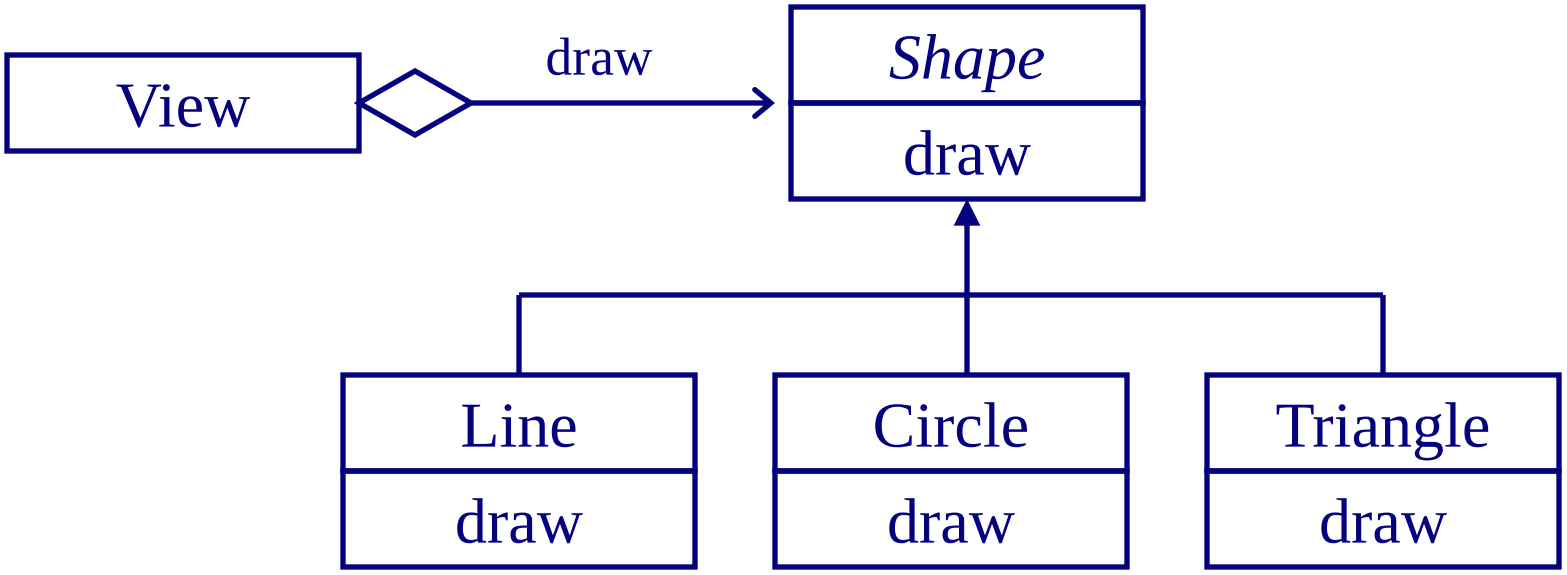


Example – Polymorphism



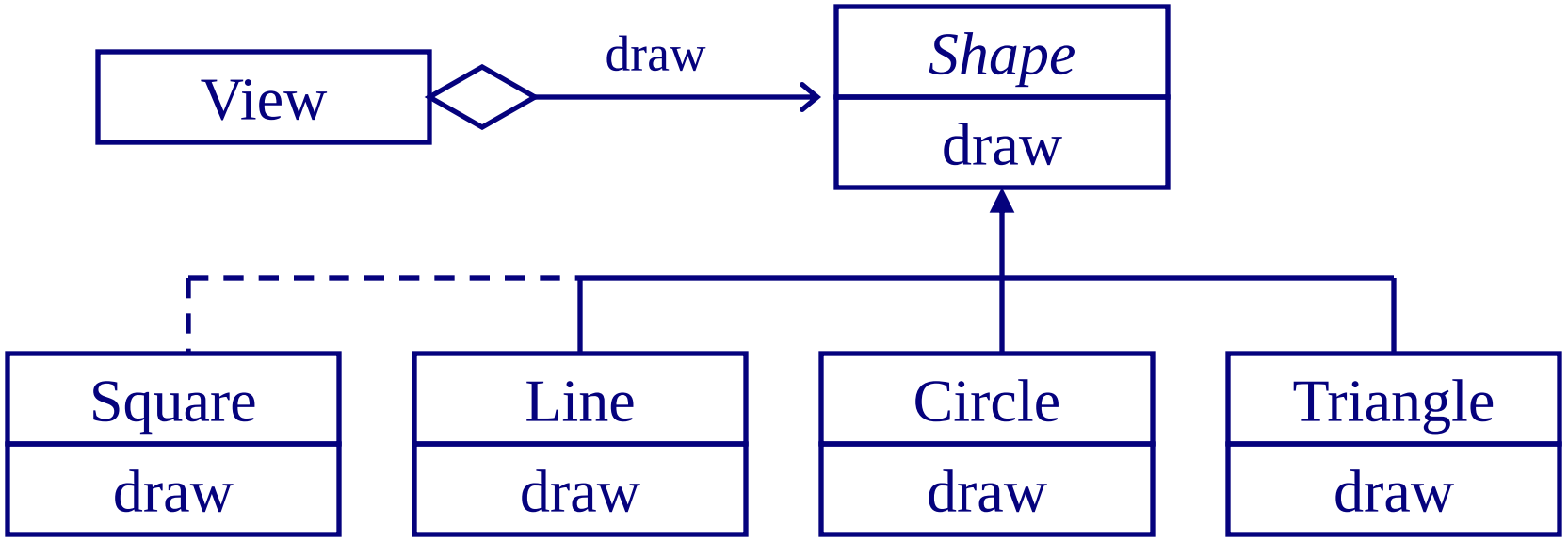
Polymorphism – Advantages

- Messages can be **interpreted** in different ways depending upon the receiver class



Polymorphism – Advantages

- New classes can be added without changing the existing model



Polymorphism – Advantages

- In general, polymorphism is a powerful tool to develop **flexible and reusable systems**

Dynamic Binding

- Binding refers to the **linking of a procedure call** to the code to be executed **in response to the call**
- Dynamic binding (late binding) means that the **code** associated with a given procedure call is **not known until** the time of the call at **run-time**
- A function call associated with a polymorphic reference depends on the dynamic type of that references
- It is associated with **polymorphism** and **inheritance**

Object-Oriented Modeling

An Example

Problem Statement

- Develop a graphic editor that can draw different geometric shapes such as line, circle and triangle. User can select, move or rotate a shape. To do so, editor provides user with a menu listing different commands. Individual shapes can be grouped together and can behave as a single shape.

Identify Classes

- **Extract nouns in the problem statement**
- Develop a graphic **editor** that can draw different geometric **shapes** such as **line**, **circle** and **triangle**. **User** can select, move or rotate a **shape**. To do so, **editor** provides **user** with a **menu** listing different **commands**. Individual **shapes** can be grouped together and can behave as a single **shape**.

…Identify Classes

- Eliminate irrelevant classes
- Editor – Very broad scope
- User – Out of system boundary

…Identify Classes

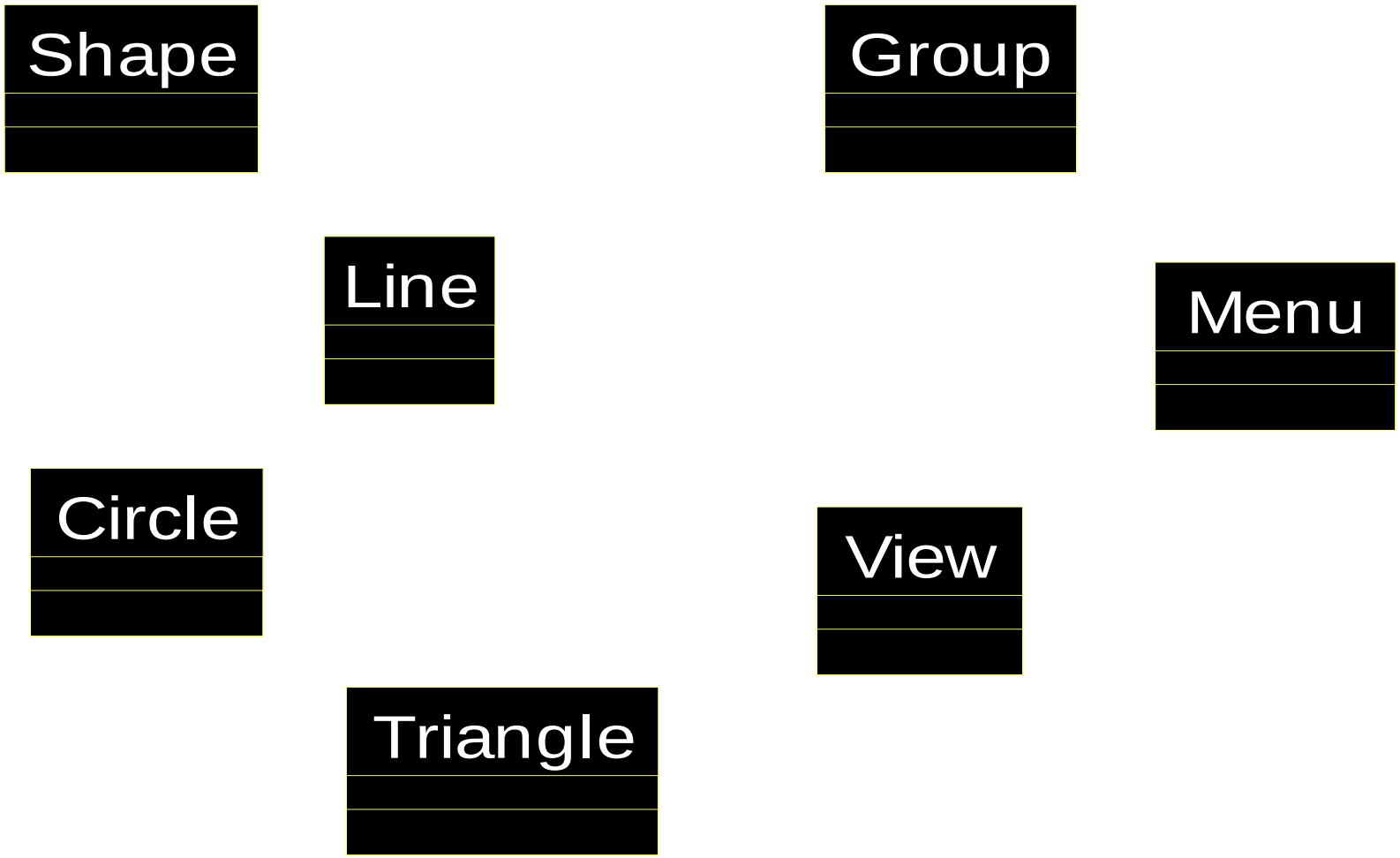
- Add classes by analyzing requirements
- **Group** – required to behave as a shape
  - "Individual shapes can be **grouped** together and can behave as a single shape"
- **View** – editor must have a display area

…Identify Classes

- Shape
- Line
- Circle
- Triangle
- Menu

- Group
- View

Object Model – Graphic Editor

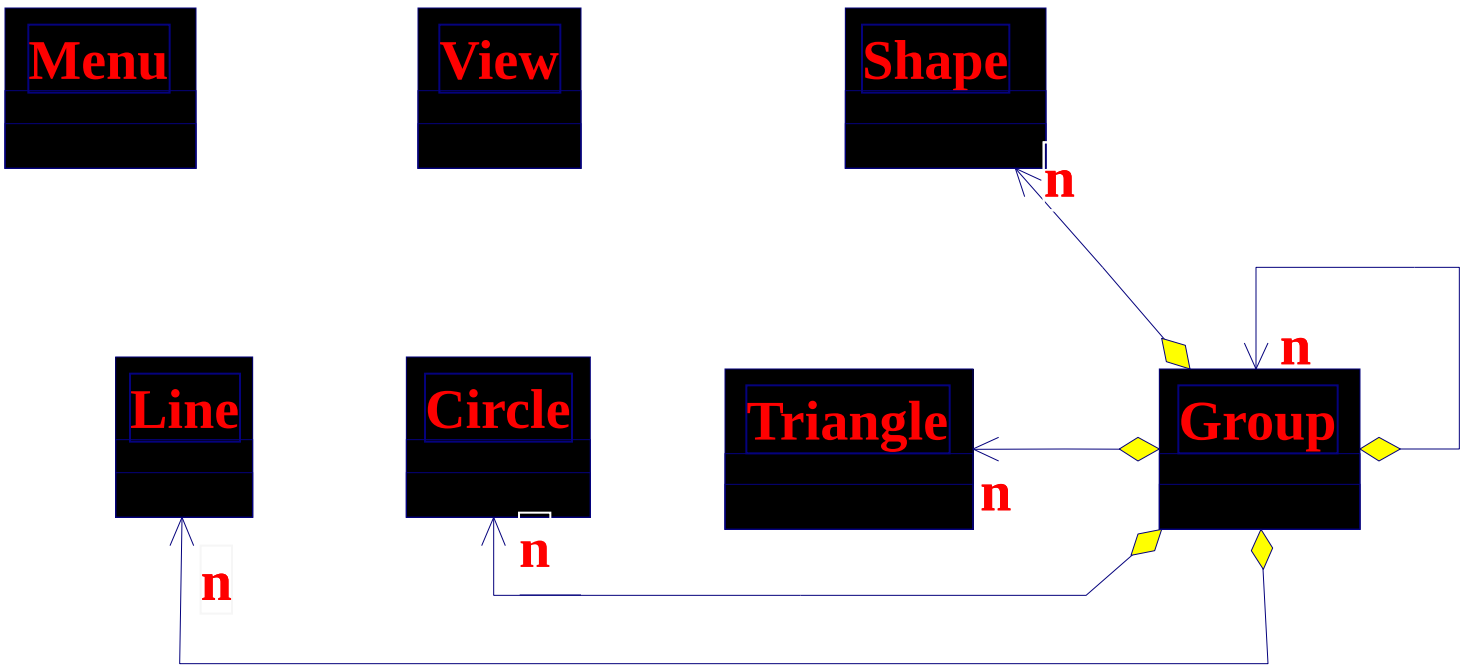


Identify Associations

- Extract verbs connecting objects
- "Individual shapes can be **grouped** together"
  - Group **consists** of lines, circles, triangles
  - Group can also **consists** of other groups

(Composition)

Object Model – Graphic Editor

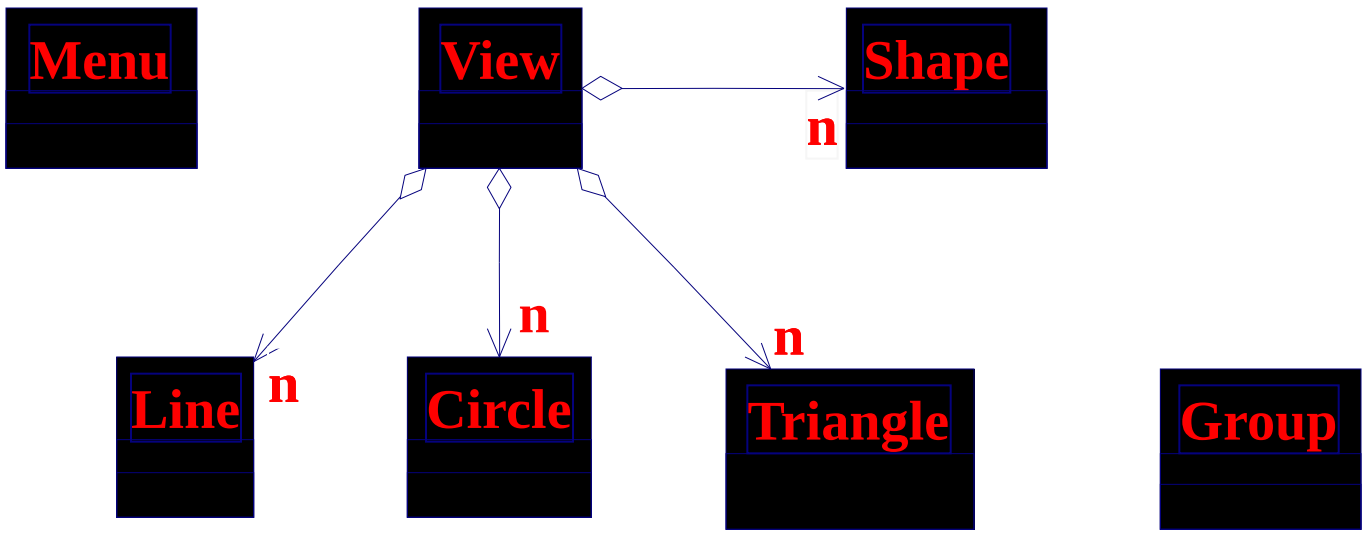


… Identify Associations

- Verify access paths
- View contains shapes
  - View contains lines
  - View contains circles
  - View contains triangles
  - View contains groups

(Aggregation)

Object Model – Graphic Editor

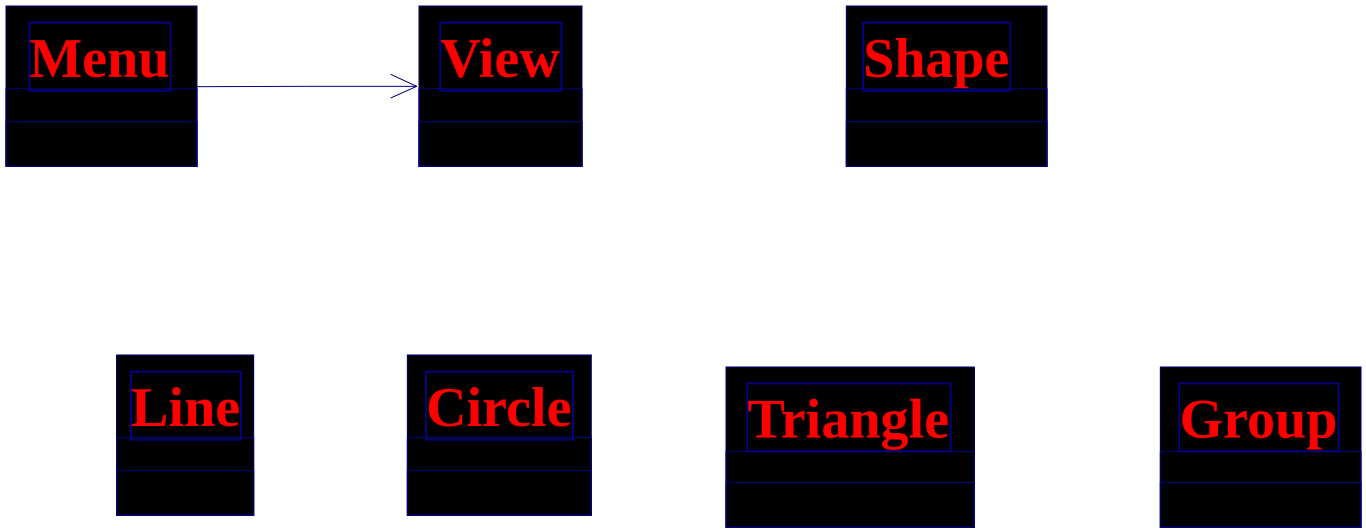


… Identify Associations

- Verify access paths
- **Menu sends** message to **View**

  (Simple One-Way Association)

Object Model – Graphic Editor



Identify Attributes (Abstraction)

- Extract **properties (Data)** of the object
  - From the problem statement
- Properties are not mentioned

Identify Attributes (Abstraction)

Extract properties of the object

— From the domain knowledge

- **Line**
  - Color
  - Vertices
  - Length
- **Circle**
  - Color
  - Vertices
  - Radius
- **Triangle**
  - Color
  - Vertices
  - Angle
- **Shape**
  - Color
  - Vertices

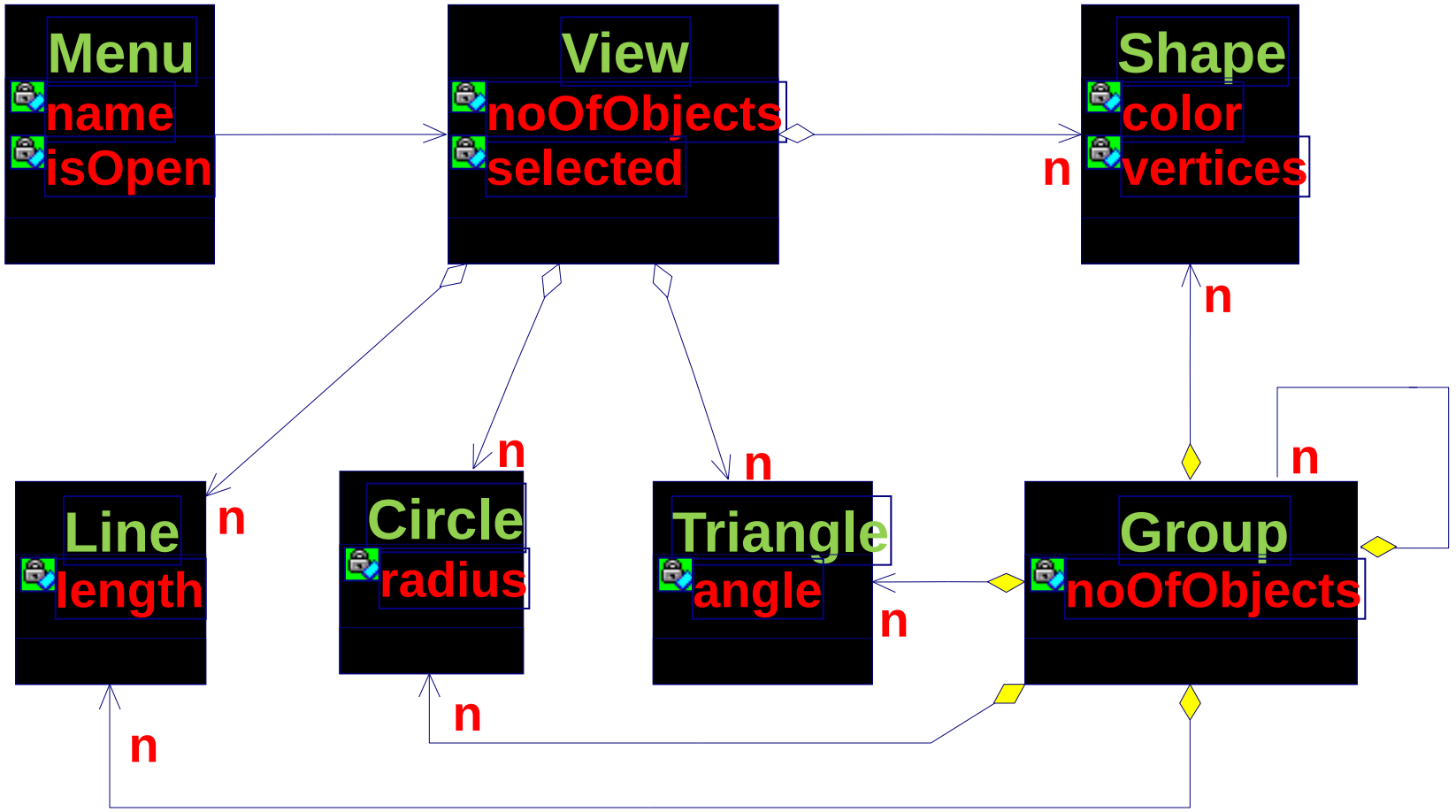
Identify Attributes (Abstraction)

Extract properties of the object

— From the domain knowledge

- Group
  - noOfObjects
- View
  - noOfObjects
  - selected
- Menu
  - Name
  - isOpen

Object Model – Graphic Editor



Identify Operations (Abstraction)

- Extract **verbs** connected with an object
- **Develop** a graphic editor that can **draw** different geometric shapes such as line, circle and triangle. User can **select**, **move** or **rotate** a shape. To do so, editor **provides** user with a menu listing different commands. Individual shapes can be **grouped** together and can **behave** as a single shape.

Identify Operations (Abstraction)

- Eliminate irrelevant operations
- **Develop** – out of system boundary
- **Behave** – have broad semantics

Identify Operations (Abstraction)

Following are selected operations:

- **Line**
  - Draw
  - Select
  - Move
  - Rotate
- **Circle**
  - Draw
  - Select
  - Move
  - Rotate

Identify Operations (Abstraction)

Following are selected operations:

- **Triangle**
  - Draw
  - Select
  - Move
  - Rotate
- **Shape**
  - Draw
  - Select
  - Move
  - Rotate

Identify Operations (Abstraction)

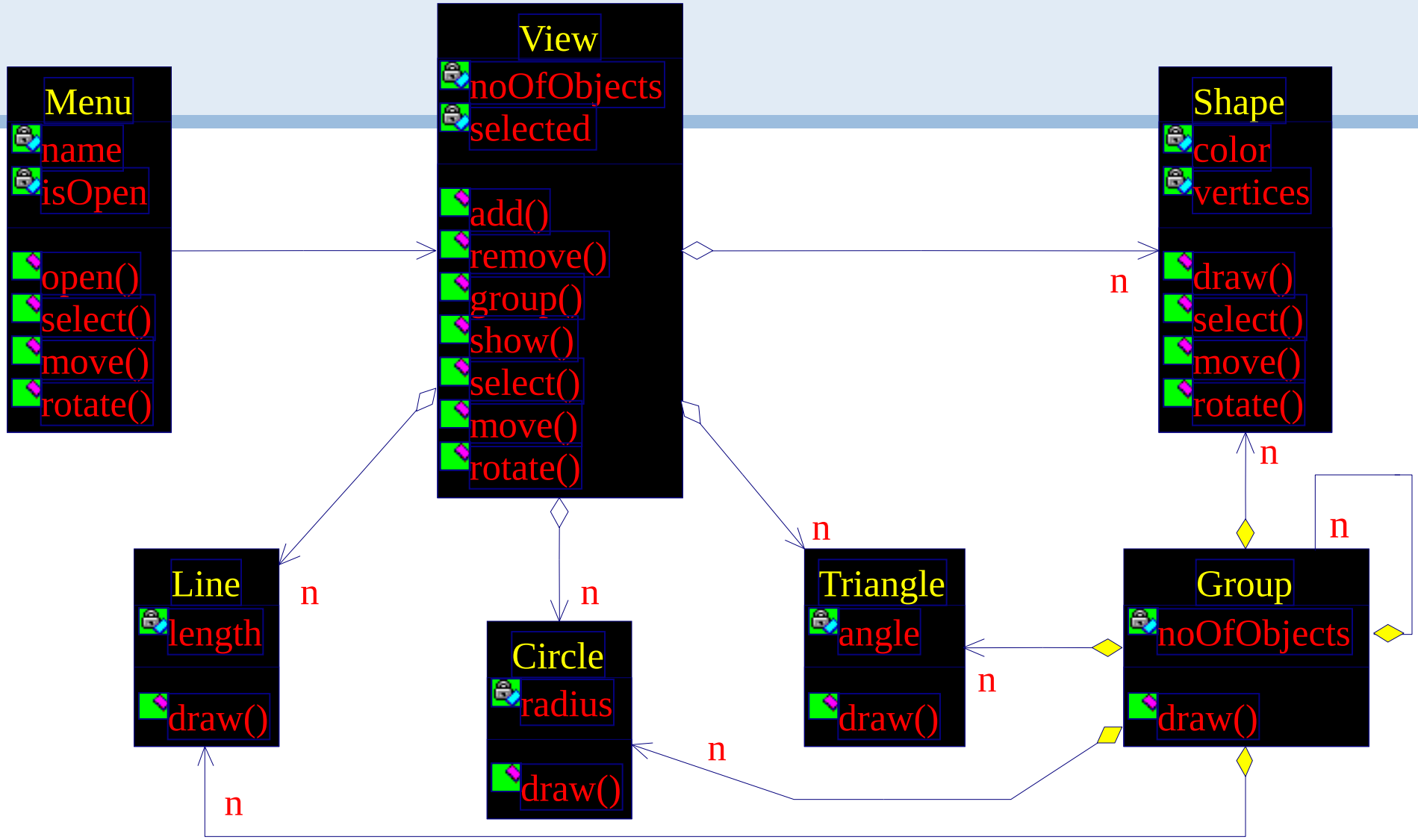
Following are selected operations:

- **Group**
  - Draw
  - Select
  - Move
  - Rotate
- **Menu**
  - Open
  - Select
  - Move
  - Rotate

Identify Operations (Abstraction)

Extract operations using domain knowledge

- View
  - Add
  - Remove
  - Group
  - Show
  - Select
  - Move
  - Rotate


View
noOfObjects
selected
add()
remove()
group()
show()
select()
move()
rotate()
Menu
name
isOpen
open()
select()
move()
rotate()
Shape
color
vertices
draw()
select()
move()
rotate()
n
Line
length
draw()
Circle
radius
draw()
Triangle
angle
draw()
Group
noOfObjects
draw()

Identify Inheritance

- **Search** "**is a kind of**" by looking at keywords like "**such as**", "for example", etc

  "…shapes **such as** line, circle and triangle…"

  Line, Circle and Triangle **inherits** from **Shape**

…Identify Inheritance

- By analyzing requirements
  - "Individual shapes can be grouped together and can behave as a single shape"
  - **Group inherits from Shape**

Refining the Object Model

- Application of inheritance demands an iteration over the whole object model
- In the inheritance hierarchy,
  - All attributes are shared
  - All associations are shared
  - Some operations are shared
  - Others are overridden

…Refining the Object Model

- Share associations
  - View contains all kind of shapes
  - Group consists of all kind of shapes

…Refining the Object Model

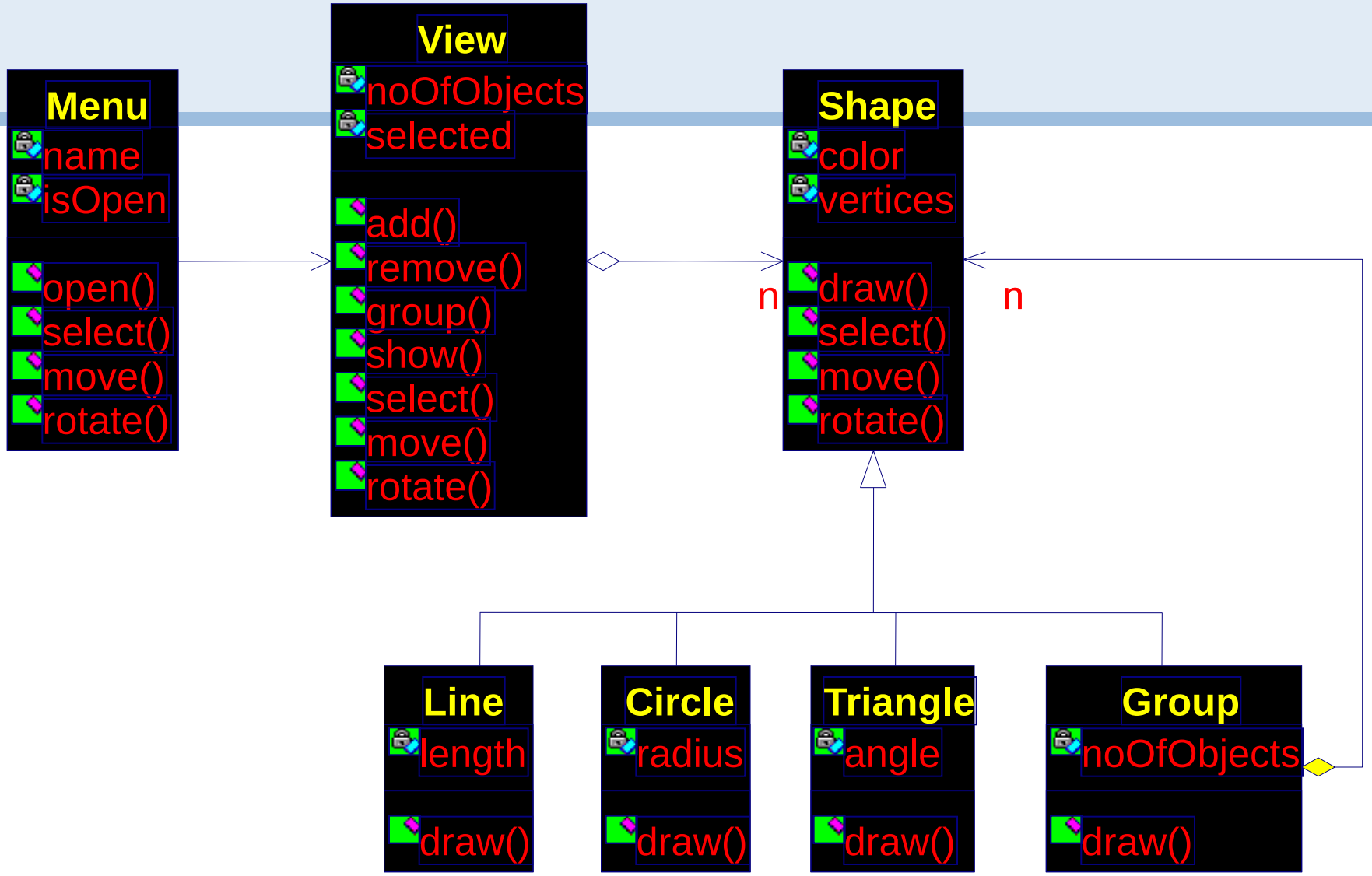
- **Share attributes**
  - Shape – Line, Circle, Triangle and Group
    - *Color, vertices*

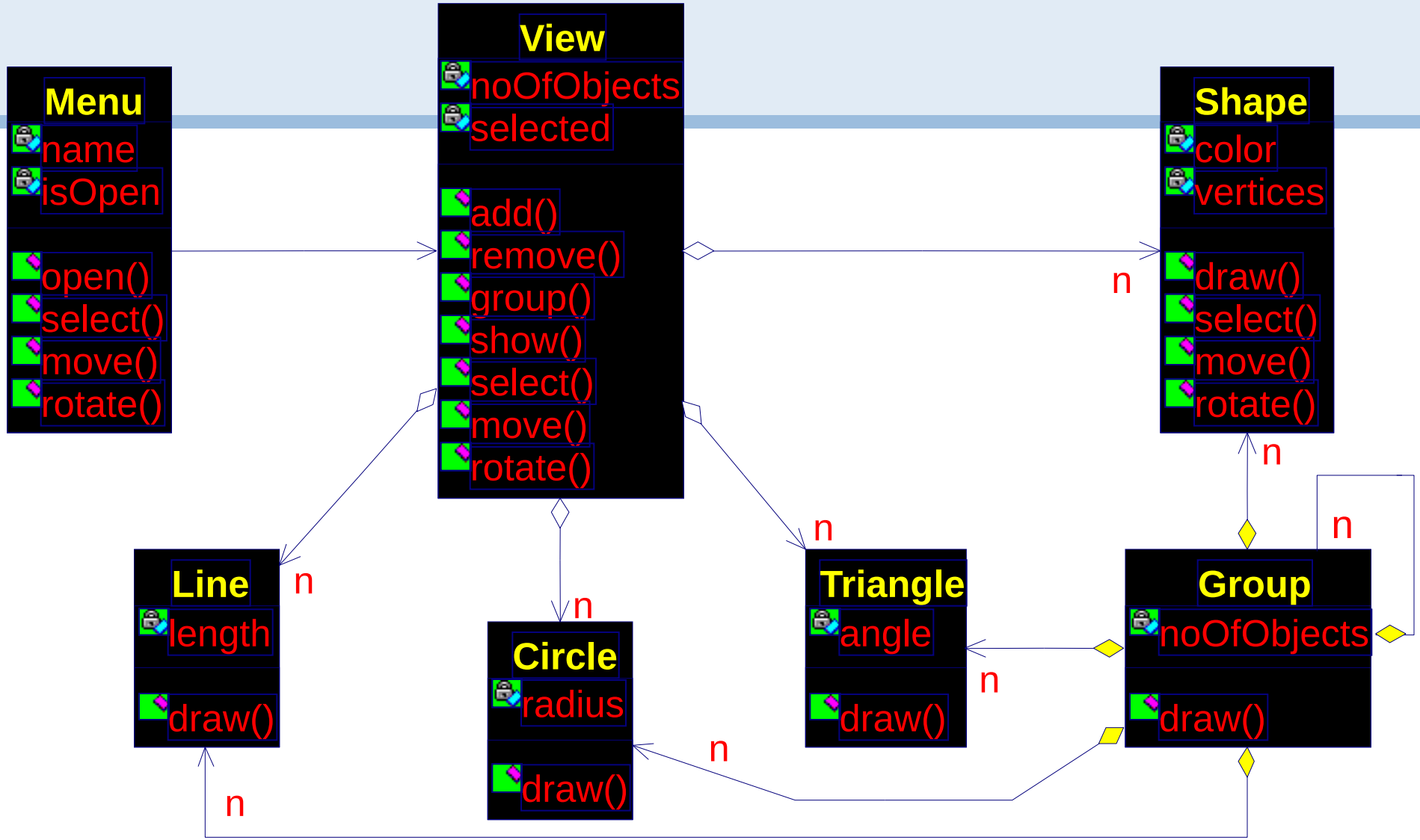
…Refining the Object Model

- Share operations
  - Shape – Line, Circle, Triangle and Group
    - *Select*
    - *Move*
    - *Rotate*

…Refining the Object Model

- Share the interface and override implementation
  - Shape – Line, Circle, Triangle and Group
    - *Draw*


Menu
name
isOpen
open()
select()
move()
rotate()
View
noOfObjects
selected
add()
remove()
group()
show()
select()
move()
rotate()
Shape
color
vertices
draw()
select()
move()
rotate()
n
n
Line
length
draw()
Circle
radius
draw()
Triangle
angle
draw()
Group
noOfObjects
draw()


View
noOfObjects
selected
add()
remove()
group()
show()
select()
move()
rotate()
Menu
name
isOpen
open()
select()
move()
rotate()
Shape
color
vertices
draw()
select()
move()
rotate()
n
Line
length
draw()
Circle
radius
draw()
Triangle
angle
draw()
Group
noOfObjects
draw()