Object-Oriented Analysis and Design using JAVA

B.Tech (CSE/IT) 5th SEM
2021-2022

Module -2 : Object Oriented Analysis
Lecture-13 Identifying classes and objects

Introduction

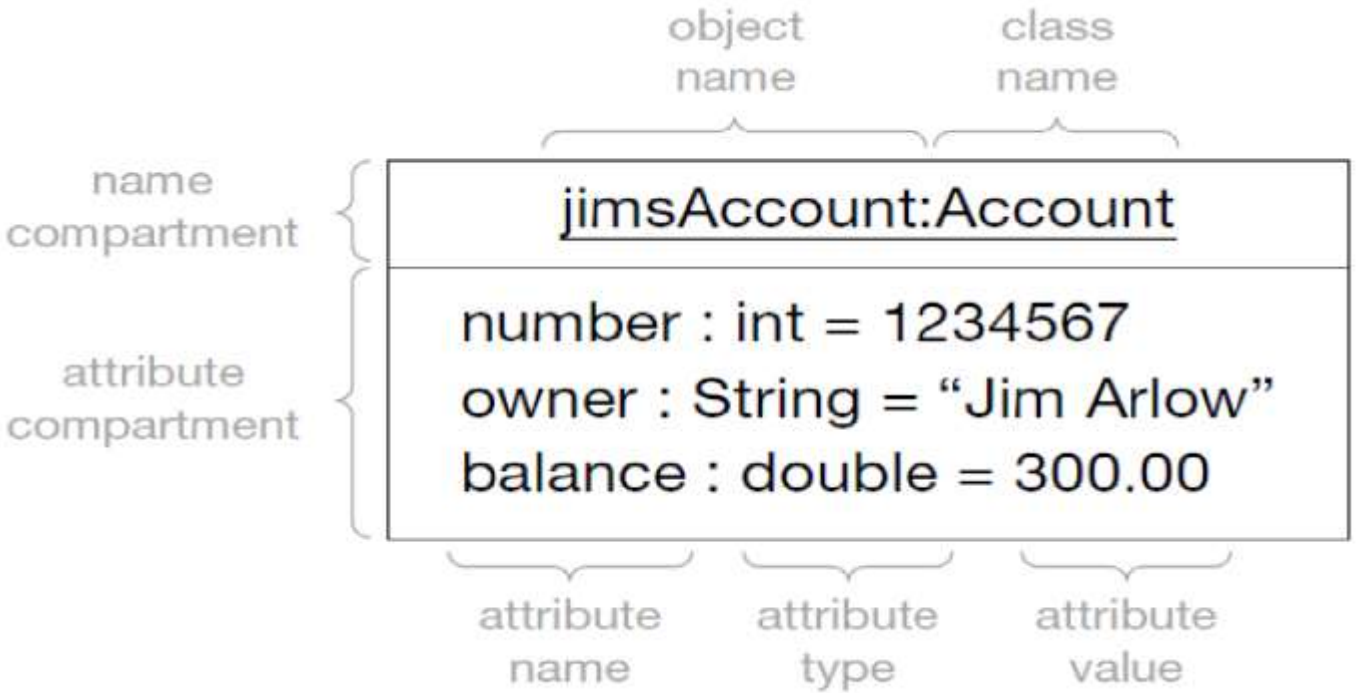
What are objects

"A discreet entity with a well-defined boundary that encapsulates state and behavior; an instance of a class."

Objects combine data and function in a cohesive unit. Every object is uniquely identifiable.

Objects hide data behind a layer of functions. Attribute values hold an object's data.

UML notation of object



Objects of the same class have the same operations and the same attributes, but may have different attribute values.

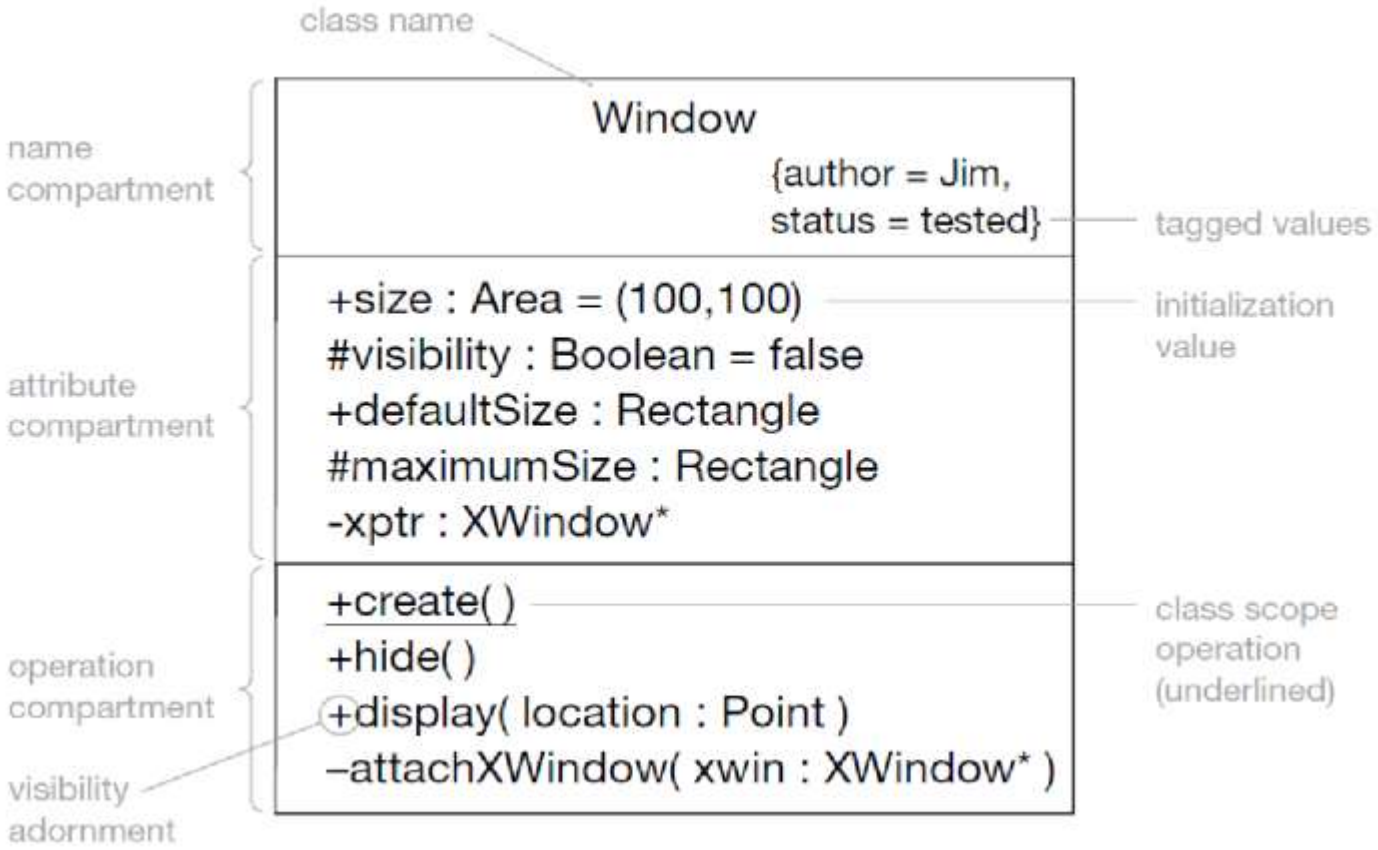
What are classes

"The descriptor for a set of objects that share the same attributes, operations, methods, relationships, and behavior."

A class describes the features of a set of objects. Every object is an instance of exactly one class.

Finding the right classification scheme is one of the keys to successful OO analysis..

UML notation of class



Name compartment

Class name is in CamelCase – it begins with an uppercase letter, and then is in mixed upper and lowercase, with each word beginning in uppercase. Special symbols such as punctuation marks, dashes, underscores, ampersands, hashes, and slashes are always avoided..

Avoid abbreviations at all costs. Class names should always reflect the names of real-world entities without abbreviation. For example, FlightSegment is always preferable to FltSgmnt, DepositAccount is always preferable to DpstAccnt

Attribute compartment

The only mandatory part of the UML attribute syntax is the attribute name:

Adornment	Visibility Name	Semantics
+	Public visibility	Any element that can access the class can access any of its features with public visibility
–	Private visibility	Only operations within the class can access features with private visibility
#	Protected visibility	Only operations within the class, or within children of the class, can access features with protected visibility
~	Package visibility	Any element that is in the same package as the class, or in a nested subpackage, can access any of its features with package visibility

Attribute compartment-multiplicity

The only mandatory part of the UML attribute syntax is the attribute name:

Multiplicity is widely used in design, but may also be used in analysis models as it can provide a concise way to express certain business constraints relating to the "number of things" participating in a relationship.
In fact, multiplicity allows you to model two distinctly different things by using a multiplicity expression

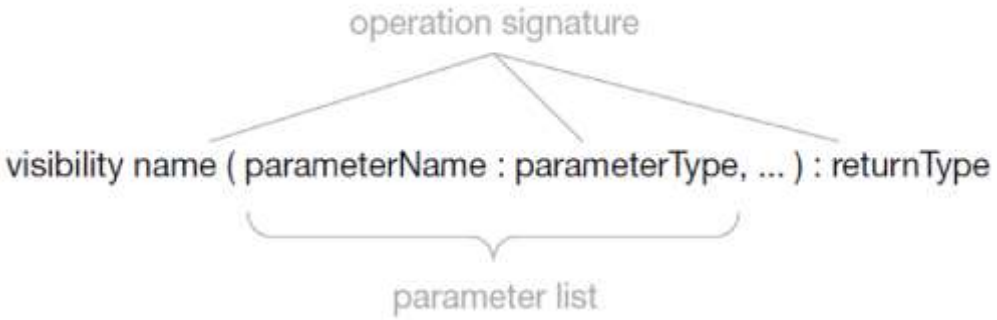
multiplicity expression

address [3]: String	an address is composed of an array or three Strings
name [2..*] : String	a name is composed of two or more Strings
emailAddress [0..1] : String	an emailAddress is composed of one String or null

Operation compartment

Operations are functions that are bound to a particular class. As such, they have all of the characteristics of functions:

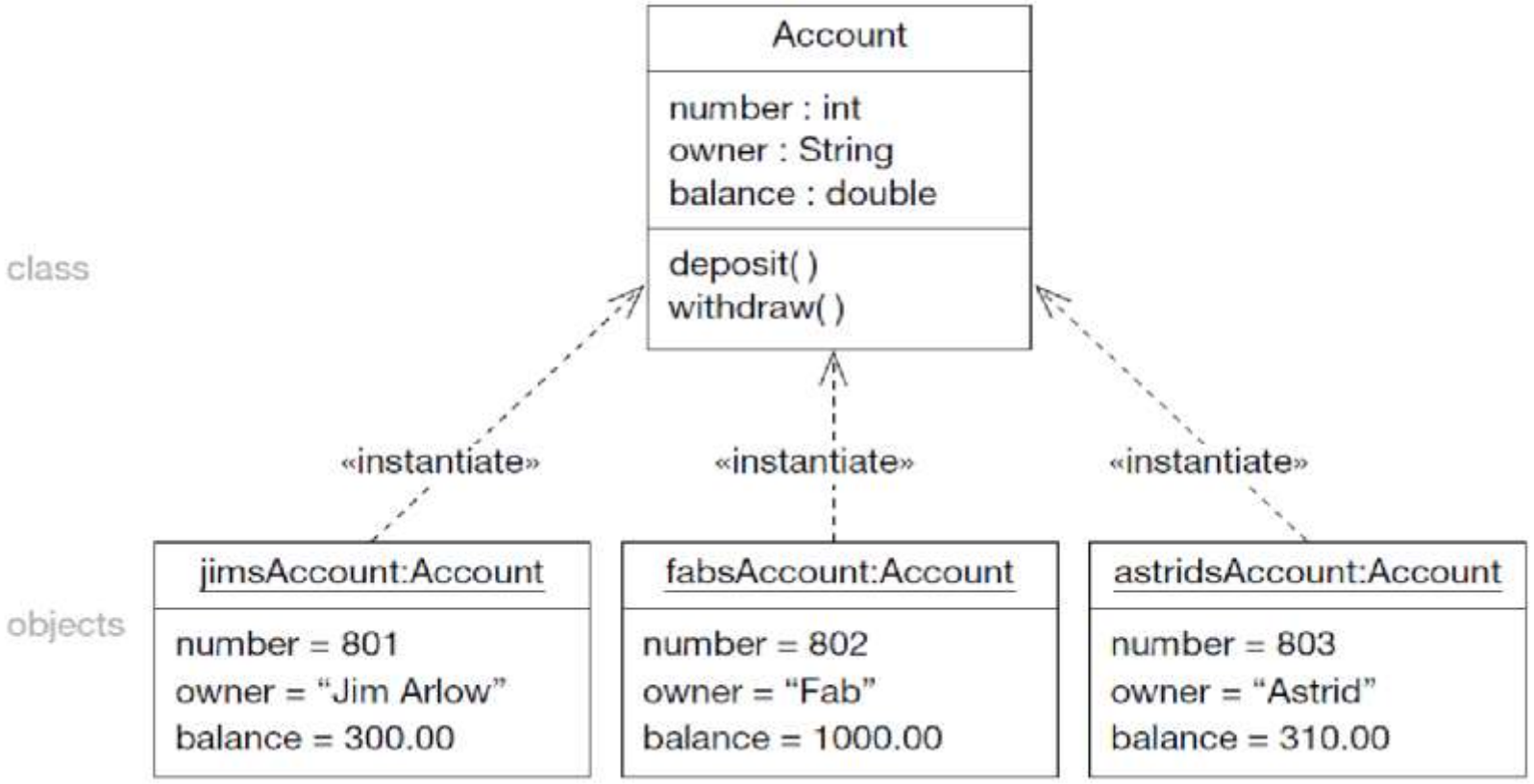
- name;
- parameter list;
- return type.



The combination of the operation name, types of all the parameters, and the return type is the operation signature. Every operation of a class must have a unique signature, as it is this signature that gives the operation its identity. When a message is sent to an object, the message signature is compared to the operation signatures defined in the object's class, and if a match is found the appropriate operation is invoked on the object.

Relationship between classes and objects

The relationship between a class and objects of that class is an «instantiate» Relationship



Key references

UML and the Unified Process Practical object-oriented analysis and design-By-Jim Arlow , Neustadt

Thank You