

INHERITANCE AND POLYMORPHISM

Generalization

Generalization is a relationship between a more general element and a more specific element, where the more specific element is entirely consistent with the more general element, but contains more information. Generalization implies the very highest level of dependency (and therefore coupling) between two elements. Generalization applies to all classifiers and to some other modeling elements such as associations, states, events, and collaborations.

Class generalization

We create a generalization hierarchy by generalizing from more specialized things and specializing from more general things. A class generalization is given in Figure:1.

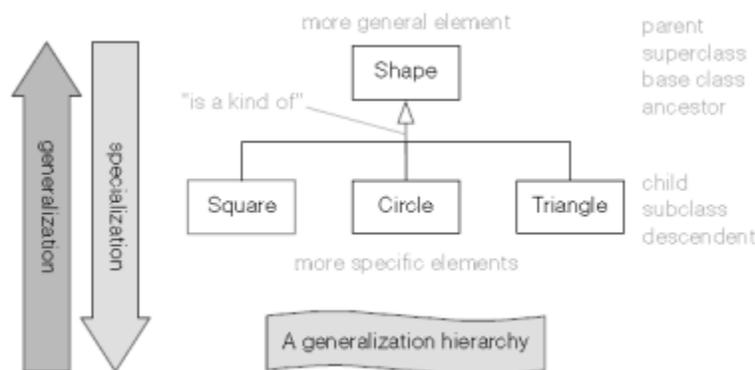


Figure: 1 Class Generalization

Class inheritance

It allows Subclasses to inherit features from their superclass. The features includes attributes, operations, relationships, constraints. Subclasses can also add new features and override superclass operations.

Overriding

Overriding means adding some more specific abilities of a superclass operation by a subclass. Subclass does this by creating a new operation with the same signature as the parent operation it wishes to override. The operation signature consists of an operation name, types of all parameters in order, and return type. The Overriding feature is indicated in Figure:2.

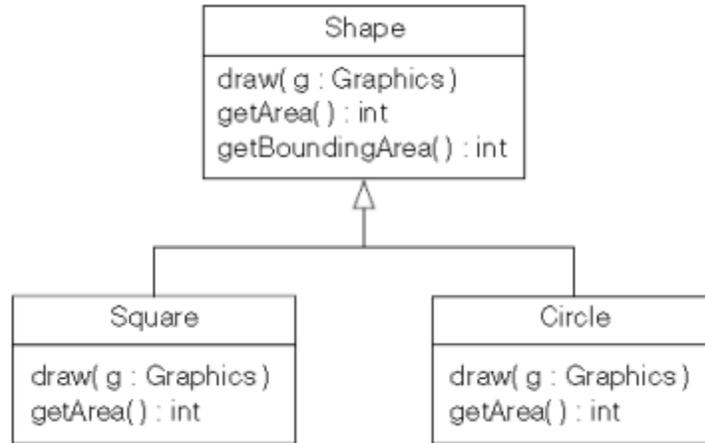


Figure:2 Overriding

Abstract operations and classes

Any class that can be instantiated is known as a **concrete class**.

Abstract operations have no implementation and are denoted in italics. All concrete subclasses must implement all inherited abstract operations.

Abstract classes have one or more abstract operations, and they can't be instantiated. Abstract classes define a contract as a set of abstract operations that concrete subclasses

must implement. These are represented in Figure: 3.

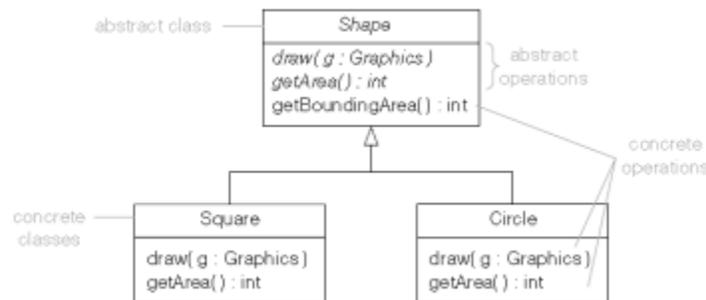


Figure:3 abstract class and concrete class

Multiple Inheritance

A class having more than one direct superclass is called as Multiple Inheritance in UML. Here the subclass inherits from all of its direct superclasses.

Polymorphism

Polymorphism means “many forms”. Polymorphic operations have many

implementations. This is well illustrated by the Figure:4. polymorphism means objects of different classes have operations with the same signature but different implementations.

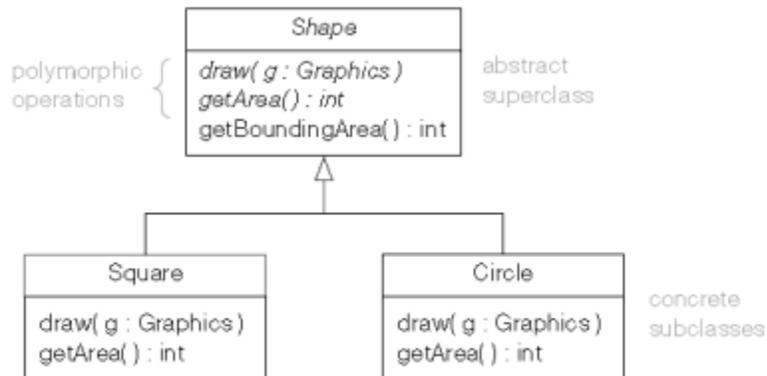


Figure:4 Polymorphism

Advanced Generalization:

Two aspects are there – generalization sets and powertypes

Generalization sets: a set of Subclasses organized according to a particular rule. It has the following constraints.

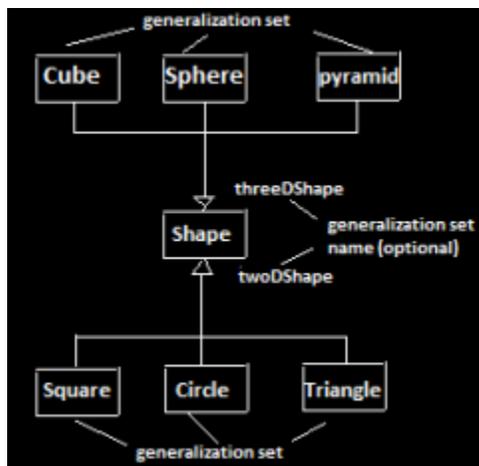
{Complete} – the generalization set contains all possible members.

{Incomplete} – the generalization set does not contain all possible members.

{Disjoint} – an object may be an instance of no more than one of the members of generalization set.

{Overlapping} – an object may be an instance of more than one of the members of generalization set.

{Incomplete, disjoint} – default



Generalization set

Power type – a class whose instances are classes that are also subclass of another class. It is a metaclass. We use prototype «powertype» to indicate it.

To use powertypes we have to partition subclasses into one or more generalization sets
And also apply powertypes to type the generalization sets.

Source : <http://praveenthomasln.wordpress.com/2012/03/01/inheritance-and-polymorphism-s6-it/>