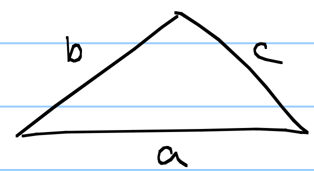
Comp 304 REEHAN SHAIKH

Note Title 3/29/2006

Class invariants

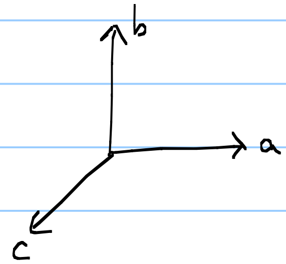
A condition/constraint all objects of a class must satisfy at all times → actually at init. time and before/after public method calls

Triangle
$a \in \mathbb{R}$ $b \in \mathbb{R}$ $c \in \mathbb{R}$
scale (factor)
Inv: $a+b \geq c$ $a+c \geq c$ $b+c \geq a$



← a new part that states class invariants

Statespace (Triangle) = $\{ \mathbb{R} \times \mathbb{R} \times \mathbb{R} \}$ ← not good



this is whole 3D continuous space w/o the invariants

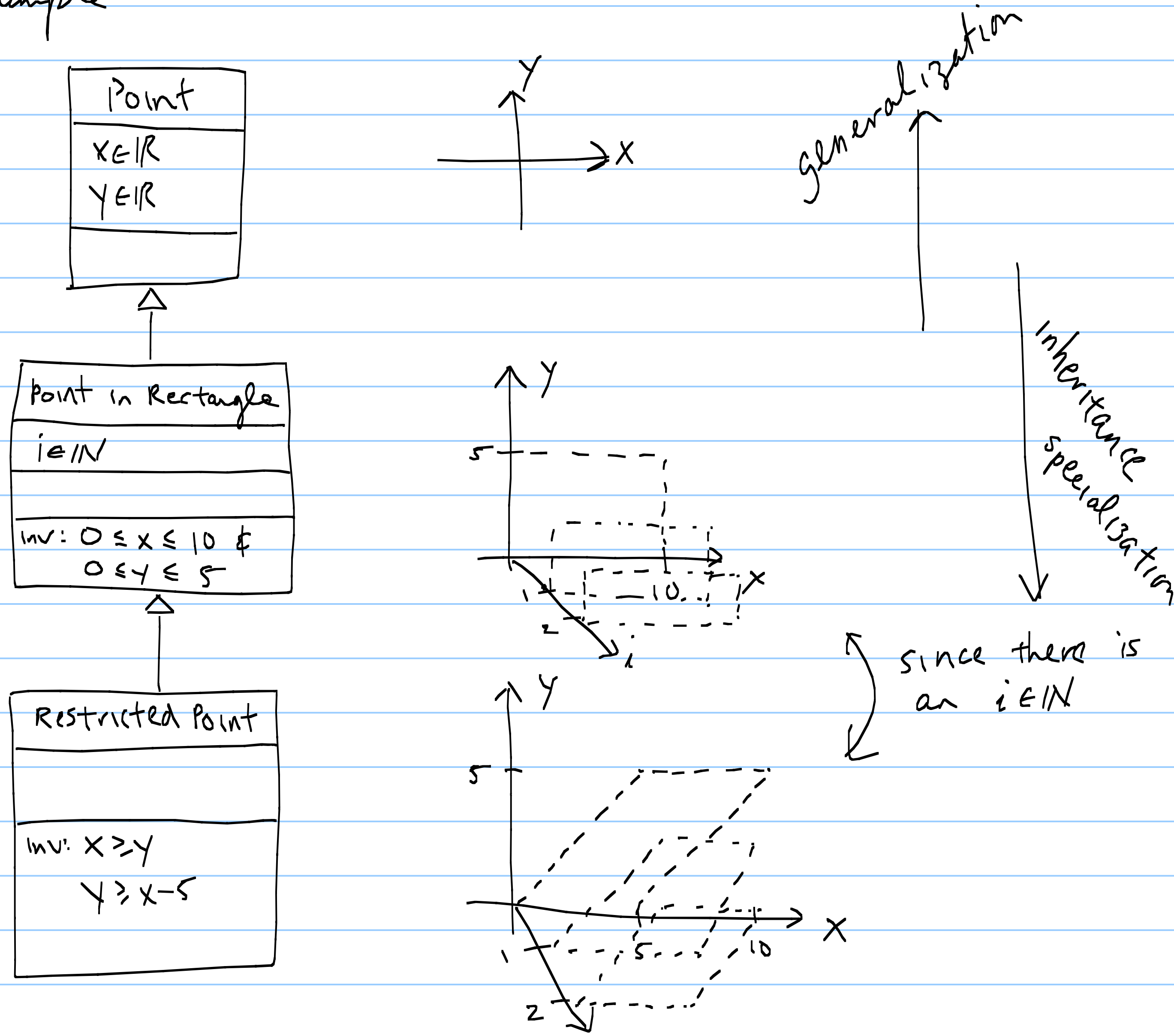
So, we add invariants. Now,

Statespace (Triangle) = $\{ (a,b,c) \mid a,b,c \in \mathbb{R},\ a+b \geq c, \ldots \}$

The reason we only care about public methods is because the outside world must see the proper constraints. Internal private methods that operate on some value may actually break the constraint.

Think of multiplying the triangle sides by f. At some point, $a+b \geq c$ will be wrong since c might be scaled first and it will be greater than $a+b$.

Example:



With inheritance, invariants are ANDED together in inherited class

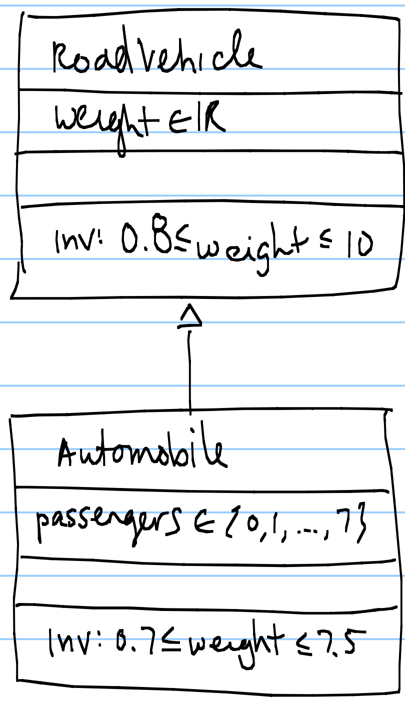
Statespace of subclass

if B is a subclass of A, then the projection of B's statespace on A must be entirely contained w/in A's statespace. statespace(B) $\subseteq$ stateSpace(A)

ie: if we project Restricted Point, we get something in the x-y plane, and it is contained in Point in Rectangle. projecting Point in Rectangle gives something in x-y plane and it is contained in Point.

This gives rise to LSP (Liskov substitutability principle)

Example:



Dimension changed but projection is still good.