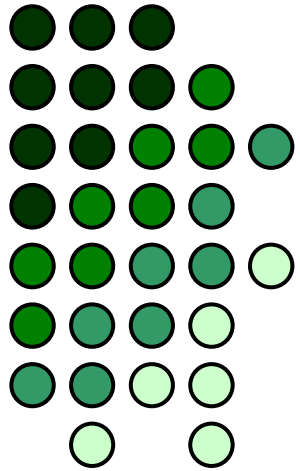
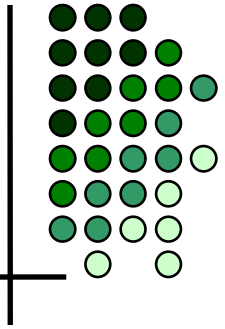
Class Relationships

Paul Inventado

De La Salle University



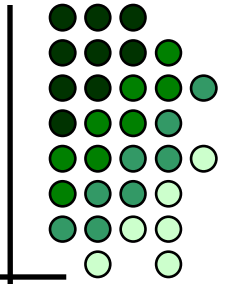
Class Relationships



Types of relationships

- Association
- Generalization / Inheritance
- Aggregation
- Composition

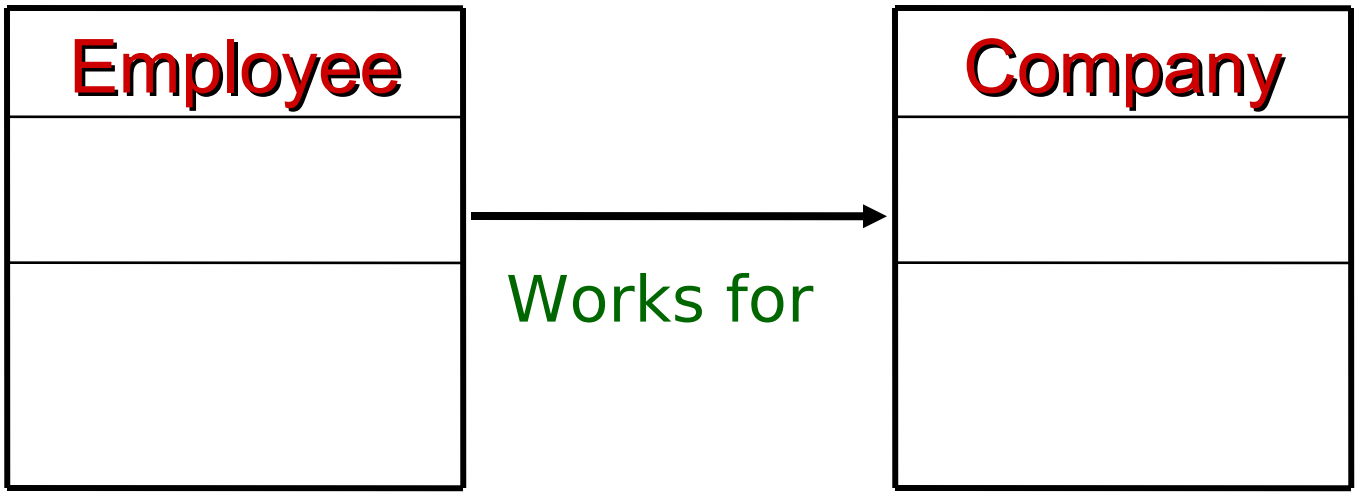
Class Relationships



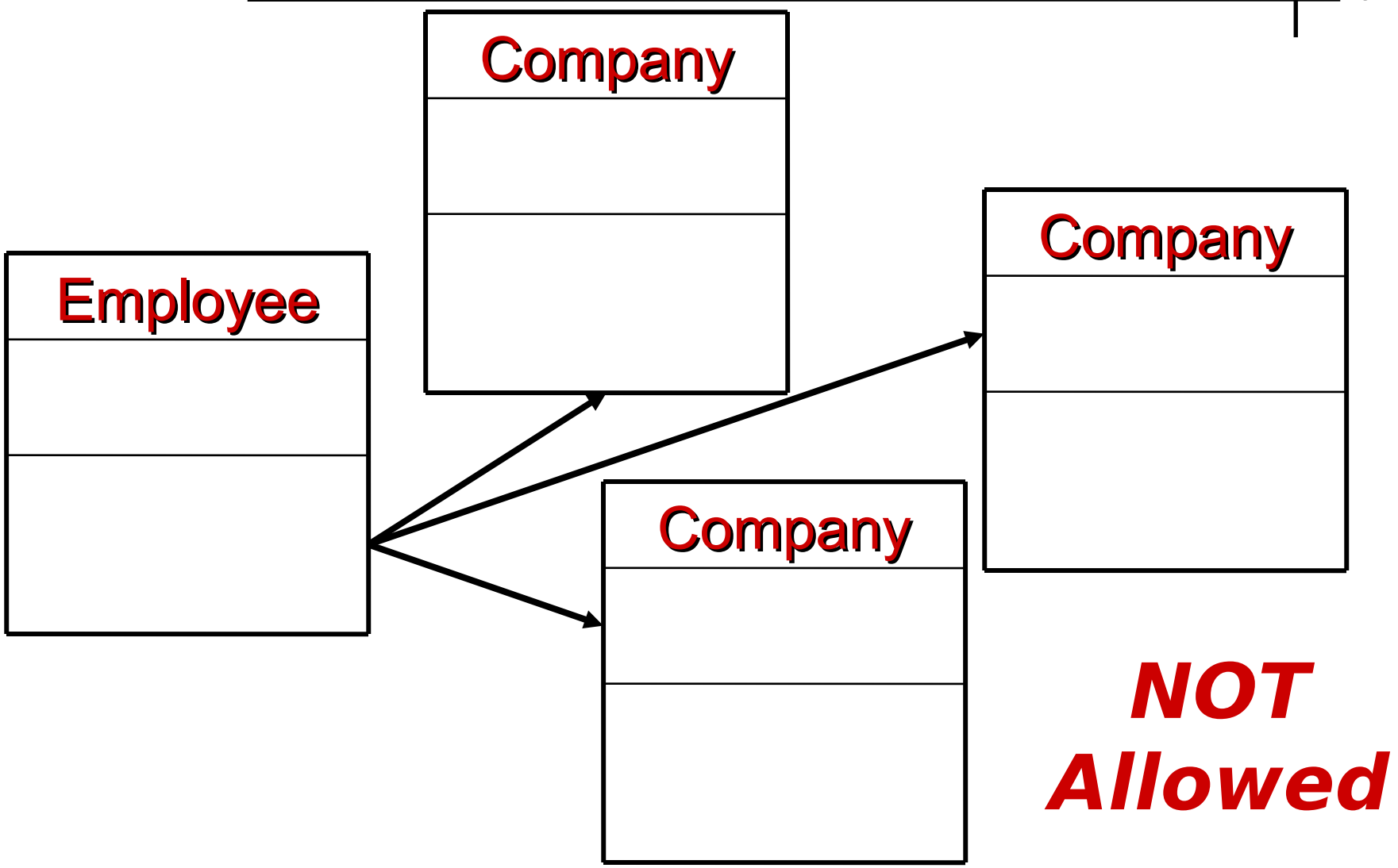
Association

- Shows the relationship between instances of classes
- Represented by a line between two classes
- May be given a name to describe the relationship it represents
- Impossible for the same pair of classes to be linked twice by the same relationship

Class Relationships



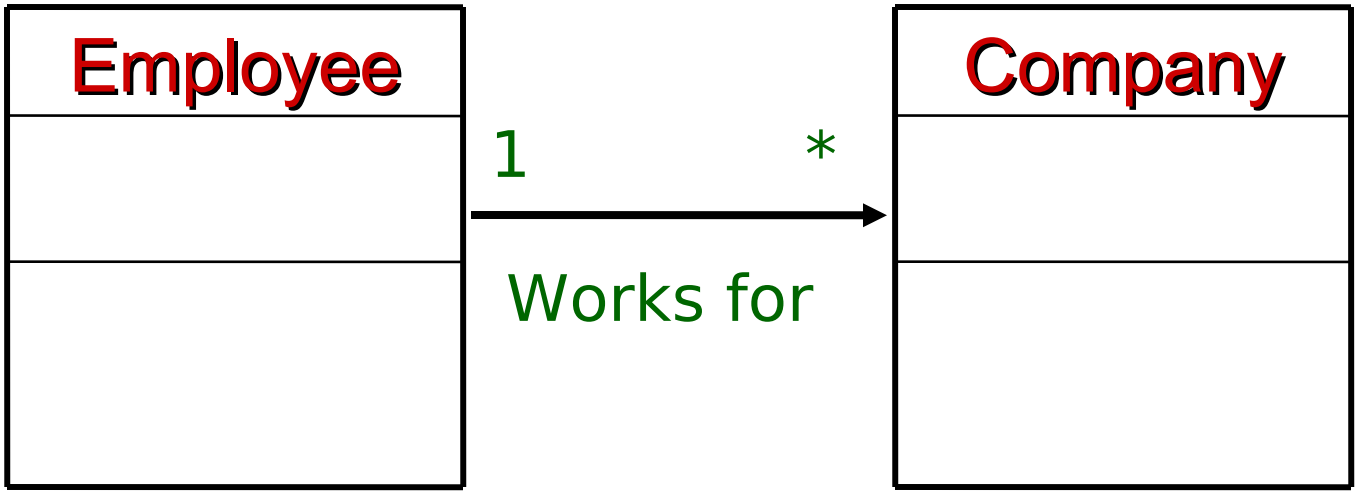
Class Relationships



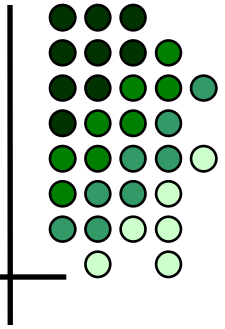
Class Relationships

Multiplicity

- Shows the extent of participation of a class in a relationship



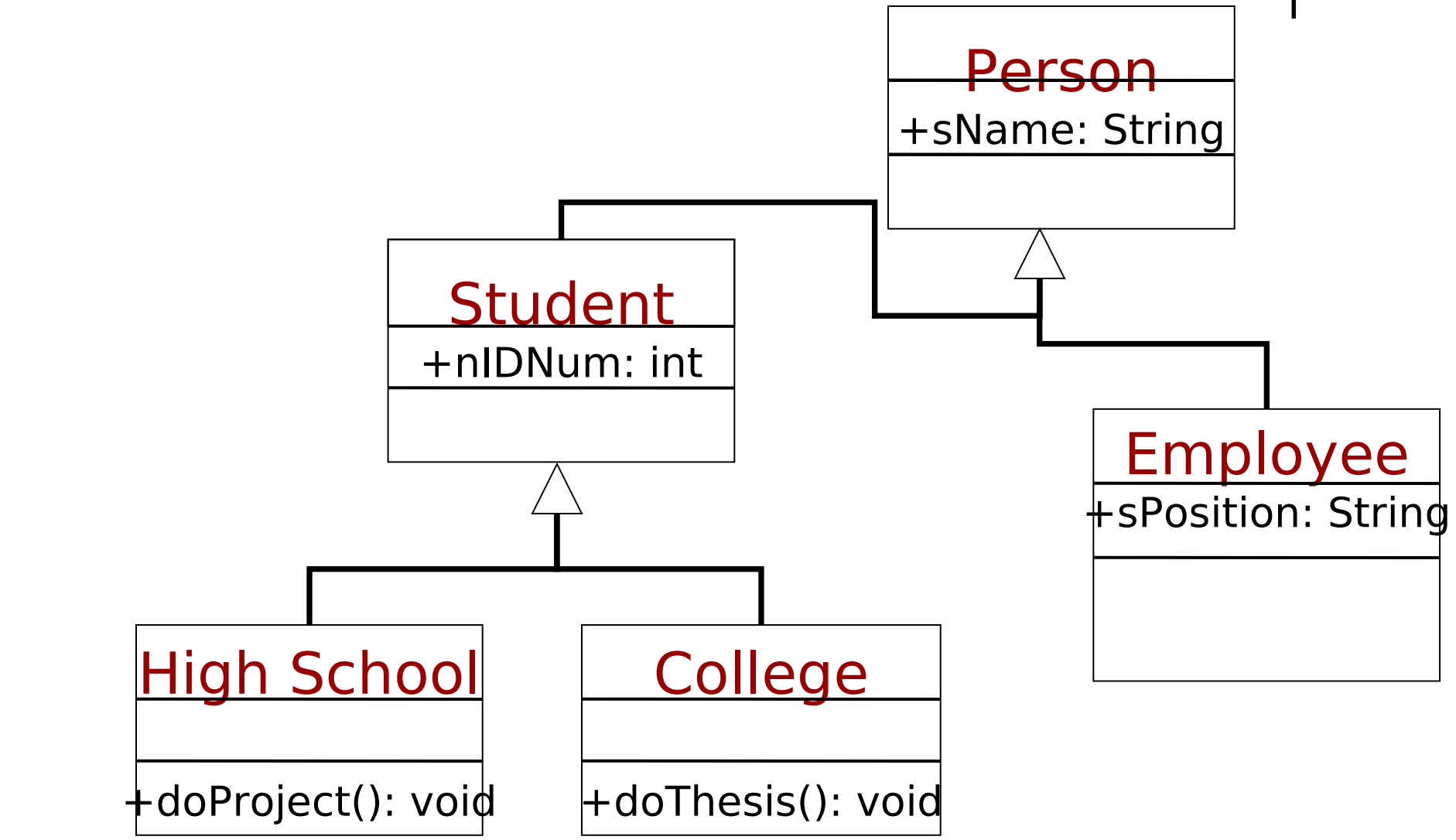
Class Relationships



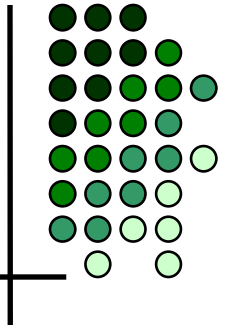
Generalization

- Needed when multiple classes share the same attributes and operations
- Removes complexity in the model and the program
- Kind-of relationship
- Used to represent inheritance

Class Relationships



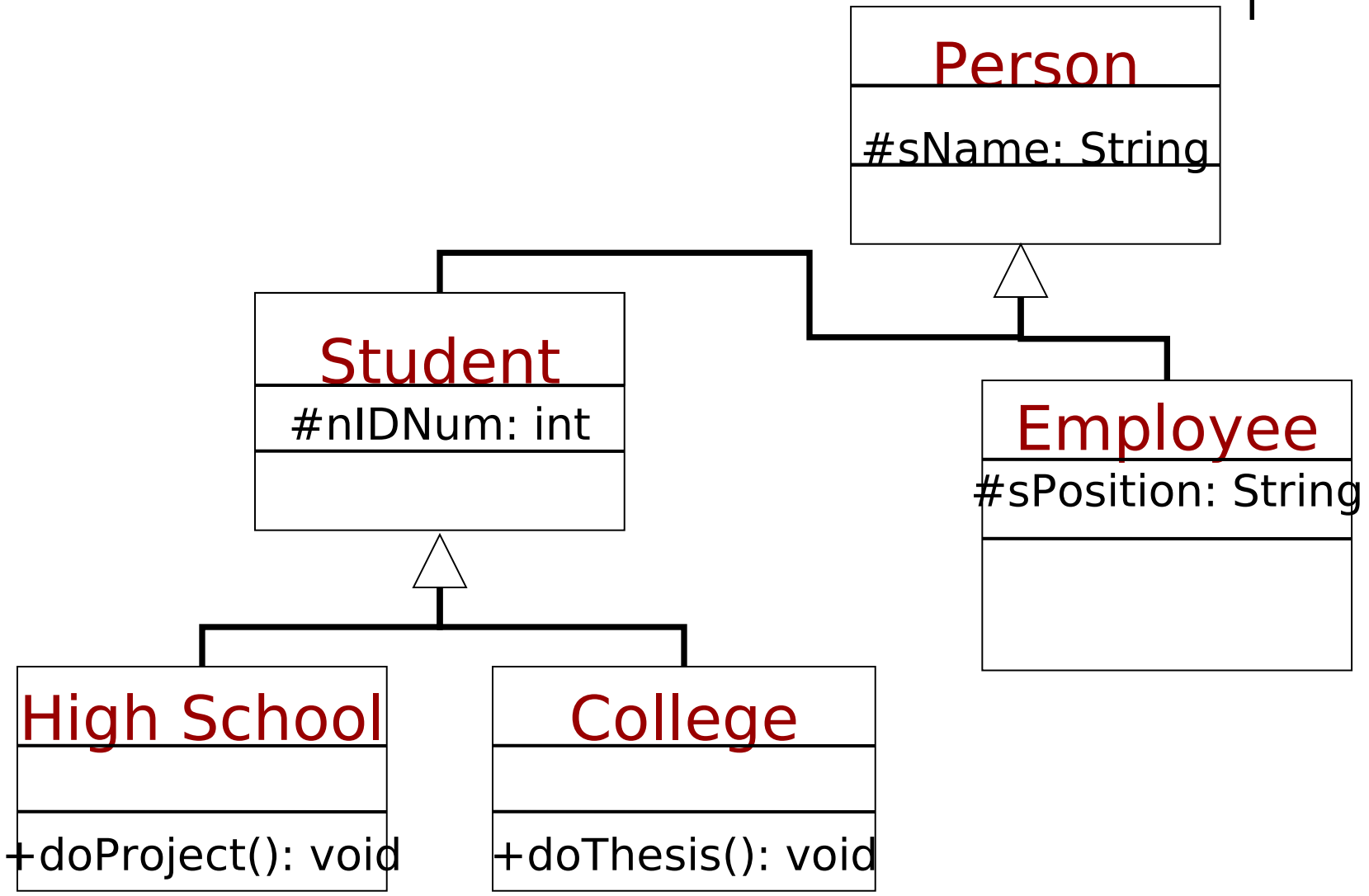
Class Relationships



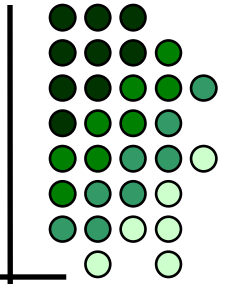
Protected Access Modifier

- Used when an attribute or method can be inherited by its subclasses
- In Java, objects that are in the same package could also access protected members directly

Class Relationships

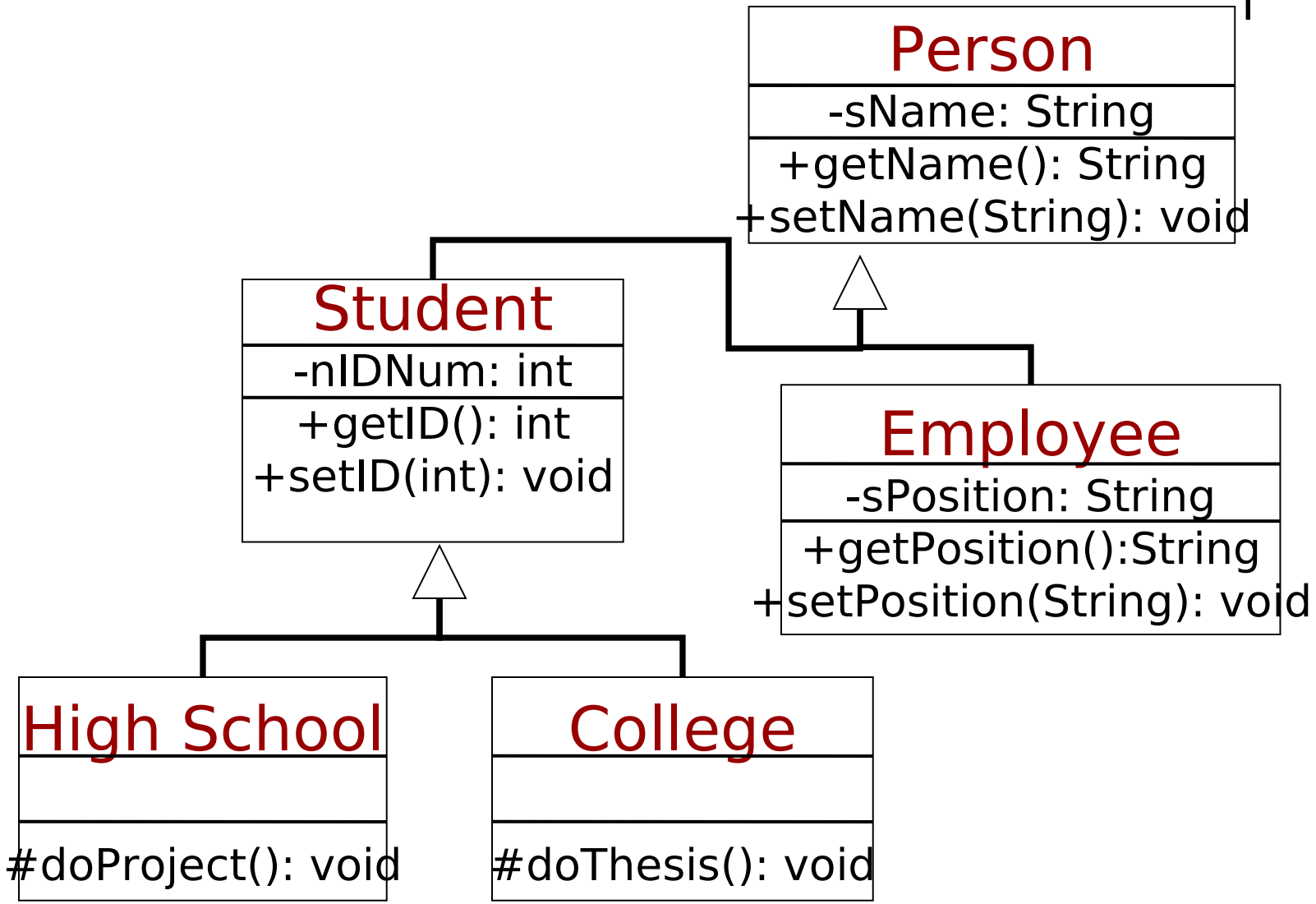


Class Relationships

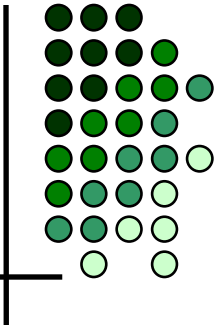


- Although protected elements make easier access, it is less encapsulated
- Attributes are still better off private
- Access to these attributes are done through access methods (getters / setters)

Class Relationships



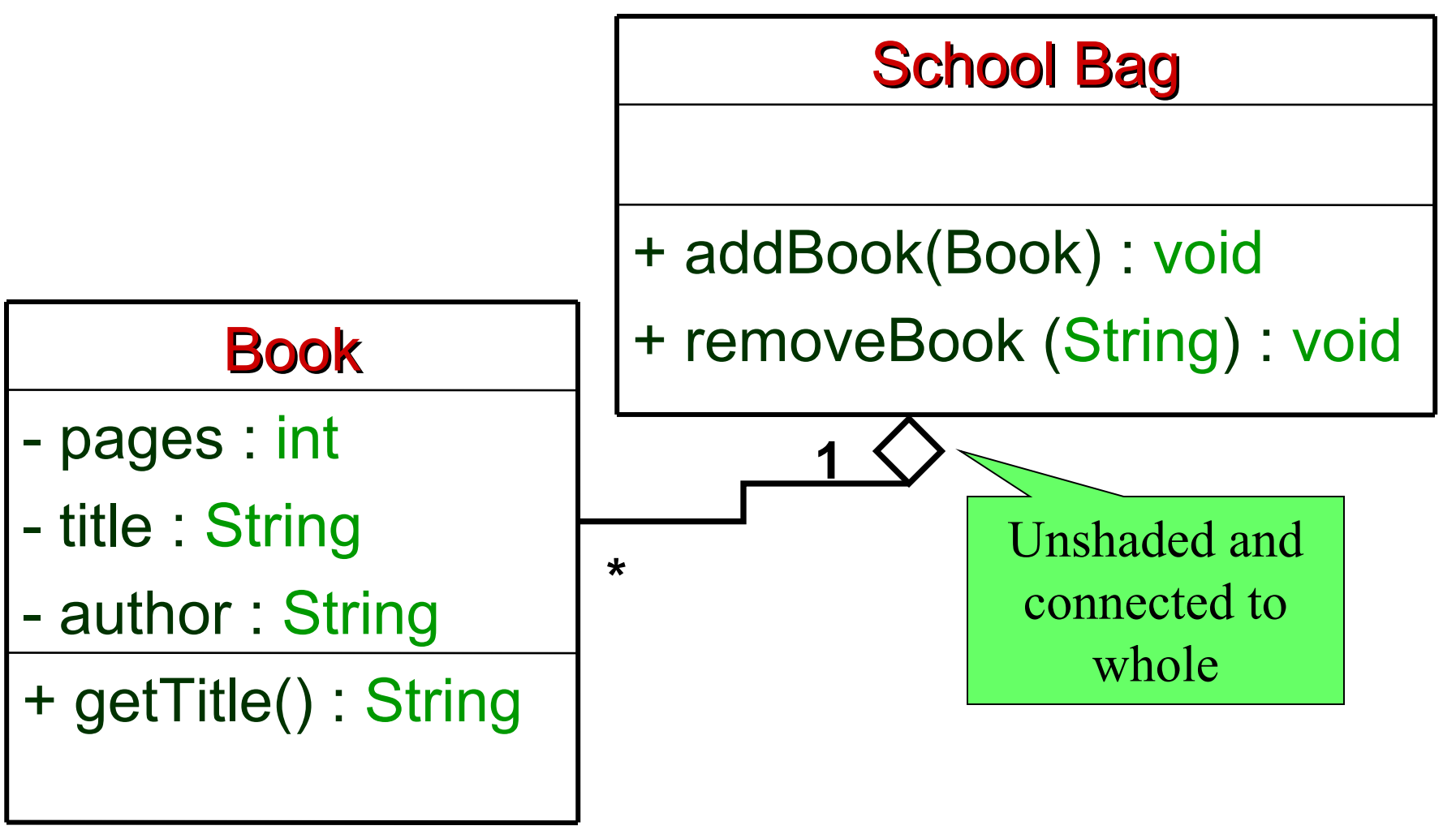
Class Relationships



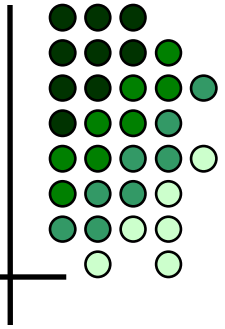
Aggregation

- Represents a part-whole relationship between classes
- Used to denote which class contains another
- Suggests the relationship of the classes without imposing any constraints

Class Relationships



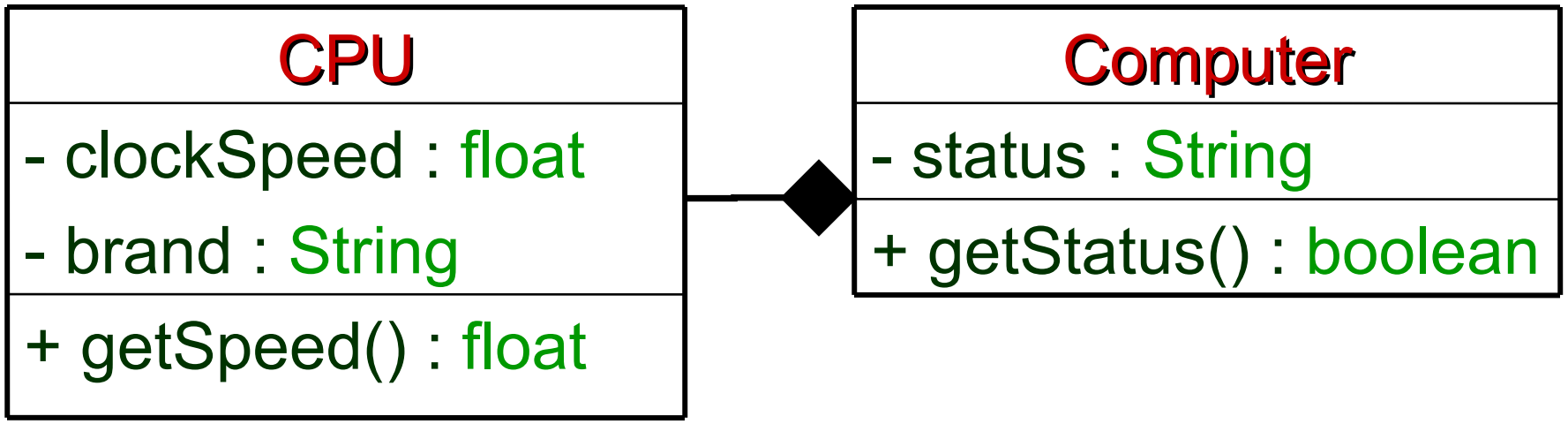
Class Relationships



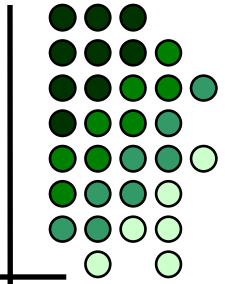
Composition

- Represents a part-whole relationship between the classes
- Used to denote which class contains another
- Denotes that one class will not exist without the other

Class Relationships

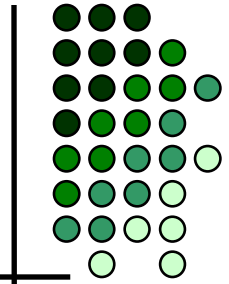


Class Relationships



- Identify the most appropriate relationship in the scenarios listed below
  - Computer – Mouse
  - Car – Engine
  - Employee – Cashier
  - DVD player – DVD
  - Nokia Cellphone – Nokia N95
  - Printer – Ink
  - Stapler – Staple wire

Class Relationships



- Choose one of the relations in the previous problem and represent it using UML. Indicate the attributes and methods of the classes. Limit the attributes and methods to what you feel is the most important to model.

Class Relationships

Paul Inventado

De La Salle University