

Types of Relationships

Shirley B. Chu

June 18, 2020

De La Salle University
College of Computer Studies

Association

Association happens when an object has a method that interacts with another object, i.e. the parameter to a method of an object is an object too.

Examples:



Boy eats fruit.



Soldier equips weapon.

Association



Association



1. The arrow points from the object who is initiating the interaction to the object who is being interacted with.

Association



1. The arrow points from the object who is initiating the interaction to the object who is being interacted with.
2. There should be a **label** describing the kind of association.

Boy Eats Fruit

Boy

- name : String
- sugarLevel : int

+ Boy (name : String)
+ eat (f: Fruit) : void
+ getName () : String
+ getSugarLevel () : int
+ toString () : String

Fruit

- name : String
- sugar : int

+ Fruit (name:String, s:int)
+ getName () : String
+ getSugar () : int

Boy Eats Fruit

Boy

```
- name : String
- sugarLevel : int

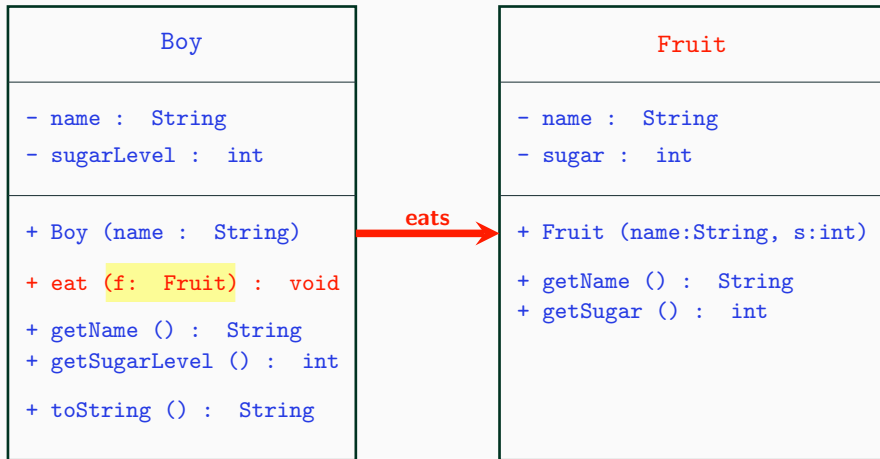
+ Boy (name : String)
+ eat (f: Fruit) : void
+ getName () : String
+ getSugarLevel () : int
+ toString () : String
```

Fruit

```
- name : String
- sugar : int

+ Fruit (name:String, s:int)
+ getName () : String
+ getSugar () : int
```

Boy Eats Fruit



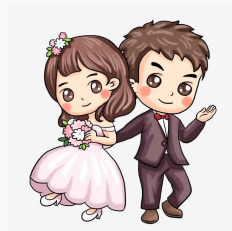
Aggregation

Aggregation happens when an object is owned by another object, and that object is **assigned** to become part of that object, i.e. one of the properties of an object is another object.

Examples:



Shopping cart has products.



Person has a spouse.

Aggregation



Aggregation



The diamond should be **beside the owner** in the relationship.

Shopping Cart

ShoppingCart

- products : Product[]
- count : int
- CAPACITY : int

- + ShoppingCart (capacity : int)
- + add (p: Product) : void
- + remove (p: Product) : void
- + listProducts () : void
- + getCapacity () : int
- + getLoad () : int
- + toString () : String

Product

- name : String
- weight : int

- + Product (n:String, w:int)
- + getName () : String
- + getWeight () : int
- + toString () : String
- + equals(obj:Object):boolean

Shopping Cart

ShoppingCart

```
- products : Product[]  
- count : int  
- CAPACITY : int  
  
+ ShoppingCart (capacity : int)  
+ add (p: Product) : void  
+ remove (p: Product) : void  
+ listProducts () : void  
+ getCapacity () : int  
+ getLoad () : int  
+ toString () : String
```

Product

```
- name : String  
- weight : int  
  
+ Product (n:String, w:int)  
+ getName () : String  
+ getWeight () : int  
+ toString () : String  
+ equals(obj:Object):boolean
```

Shopping Cart

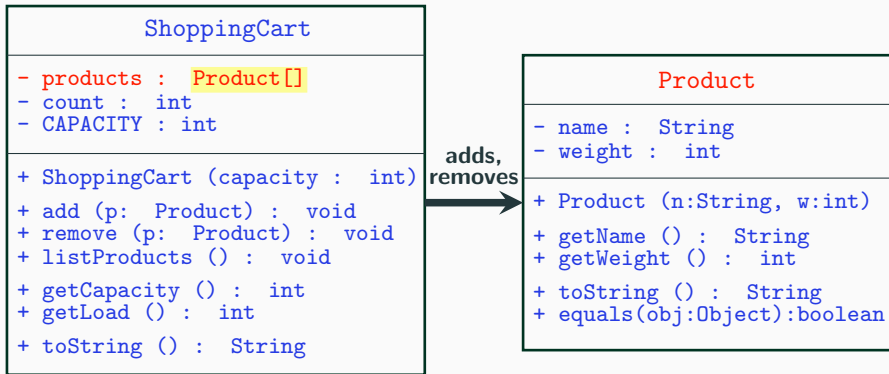
ShoppingCart
<ul style="list-style-type: none">- products : Product[]- count : int- CAPACITY : int
<ul style="list-style-type: none">+ ShoppingCart (capacity : int)+ add (p: Product) : void+ remove (p: Product) : void+ listProducts () : void+ getCapacity () : int+ getLoad () : int+ toString () : String

adds,
removes

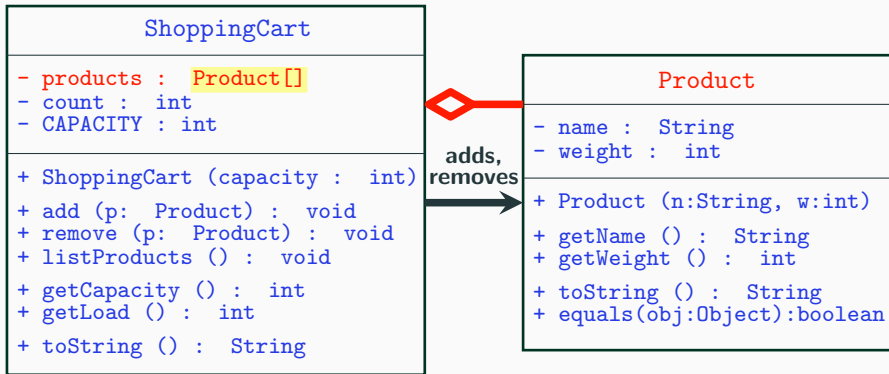


Product
<ul style="list-style-type: none">- name : String- weight : int
<ul style="list-style-type: none">+ Product (n:String, w:int)+ getName () : String+ getWeight () : int+ toString () : String+ equals(obj:Object):boolean

Shopping Cart



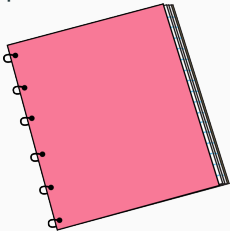
Shopping Cart



Composition

Composition happens when an object is owned by another object, and that object is **an integral part of** that object, i.e. one of the properties of an object is another object, and it is **being built or instantiated inside that object**.

Example:



Notebook has pages,
and the **pages are an integral part of the notebook**.

Composition



Composition



The diamond should be **beside the owner** in the relationship.

Notebook

Notebook

- pages : Page[]
- pageNow : int

- + Notebook (pageCount : int)
- + read () : String
- + write (c: String) : void
- + forward () : void
- + back () : void
- + getCapacity () : int
- + toString () : String

Page

- content : String
- number : int

- + Page (num : int)
- + read () : String
- + write (c: String) : void
- + getPageNo () : int
- + toString () : String

Notebook

Notebook

```
- pages : Page[]  
- pageNow : int
```

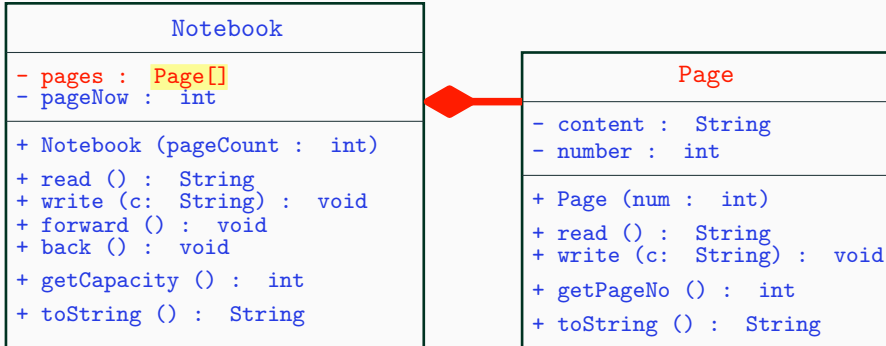
```
+ Notebook (pageCount : int)  
+ read () : String  
+ write (c: String) : void  
+ forward () : void  
+ back () : void  
+ getCapacity () : int  
+ toString () : String
```

Page

```
- content : String  
- number : int
```

```
+ Page (num : int)  
+ read () : String  
+ write (c: String) : void  
+ getPageNo () : int  
+ toString () : String
```

Notebook



😊 Thank you! 😊