Relationships(oop)

قبل متكلم عن relationships in oop كلمك الأول عن يعني ايه :

oopيعني مجموعه كلاسيز بنقوم بتكوينها لتقوم بمهام معينه .

في بعض الأحيان الكلاسيز دي بيكون لها علاقه مع بعض علشان يقوموا بمهمه اكبر علشان نفهم حاجه زي دي هنحتاج نفهم يعني ايه relationships.

Relationships:هي عبارة عن طرق ربط الكلاسات ببعضها ويوجد 4انواع من ال relationshipsوهي:

1-inheritance(is a)relationships

2-Association(has-a)relationships

3-aggregation(has-a)relationships

4-composition (part-a)relationships

بعد معرفنا ايه هي أنواع ال relationshipsهنتكلم عن كل نوع لوحده وهنبدأ ب

Inheritance

يعني إن **كلاس (class)** جديد يقدر ياخد الخصائص (Variables) والوظائف (Methods) من كلاس تاني قديم.

زي مثلا ممكن الابن يورث من ابوه لون الشعر او العين او بعض الصفات التانيه .

طب يجي واحد يقولي ليه أصلا بنعمل وراثه قولك انا بص يعم علشان نوفر على نفسنا وقت في اعادت كتابة الكود تاني بتاع الاب في الابن وكمان علشان الكود بتاعنا يكون منظم وكمان علشان نطبق عليه مبدا is a relationshipمثال:

لو عنك كلاس اسمه انسان (ده الاب) وكلاس تاني اسمه الطلاب(ده الابن) اذا العلاقه بينهم بتكون ان (الطالب يكون انسان)وده منطقي ان الطاالب في بعض صيفات الانسان .

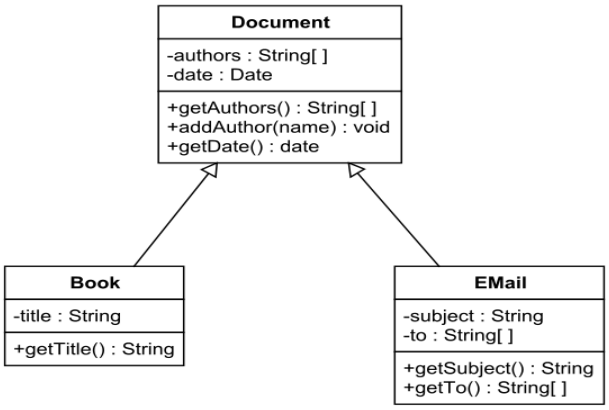
مثال بالكود:

```java
public class Main {
    public static void main(String[] args) {
        Student s1 = new Student();
        s1.name = "Ali";
        s1.age = 20;
        s1.school = "Cairo University";

        s1.sayHello();     // من كلاس Person
        s1.study();        // من كلاس Student
    }
}
```

```java
// الكلاس الأساسي (اللي كل الناس بترث منه)
class Person {
    String name;
    int age;

    void sayHello() {
        System.out.println("Hello, my name is " + name);
    }
}

// الكلاس الجديد اللي بيورث من Person
class Student extends Person {
    String school;

    void study() {
        System.out.println(name + " is studying at " + school);
    }
}
```

هنا الاب هو personوالابن هو ال student في الصوره ال على اليمين طيب تعالى بقى نبص على الشمال انا روحت عملت اوبجيكت من كلاس الابن وروحت جاي مدخل قيمه ال name ,ageمع العلم ان الاتنين variableدول موجودين في كلاس الاب بس طلما انا عملت توريث منه يبقى انا وريثت أي حاجه فيه سواء كانت variable or methodsبشرط متكونش private

دلوقت هنشوف ازاي هنمثل ال inheritanceباستخدام uml



تعالى اشرحلك الصوره هنا خلي بالك من السهم علشان ده الهيوضحلك نوع العلاقه هنا السهم راسه مفرغه والصورة دي توضح ان الاب هو decumentولان راس السهم يؤشر اليه بينما emailو bookدول الأبناء

Association

هكلمك دلوقت عن Association بص هو علاقه بين كلاسين بس بشرط ميكونش أي كلاس جزء من التاني يعني كل كلاس مستقل بمعنى اخر لو أي كلاس من الكلاسين حصله مشكله او قفل التاني ملهوش دعوة طب علشان اسهلك الدنيا اديك مثال حي:

مثلا الدكتور بيشتغل في المستشفى :دلوقت الدكتور مستقل والمستشفى مستقله بينما تربطهم علاقه شغل

والعلاقه دي يطلق عليها has-a relationshipsوممكن تكون

One-to-one || many-to-one || many-to-many || one-to-many.

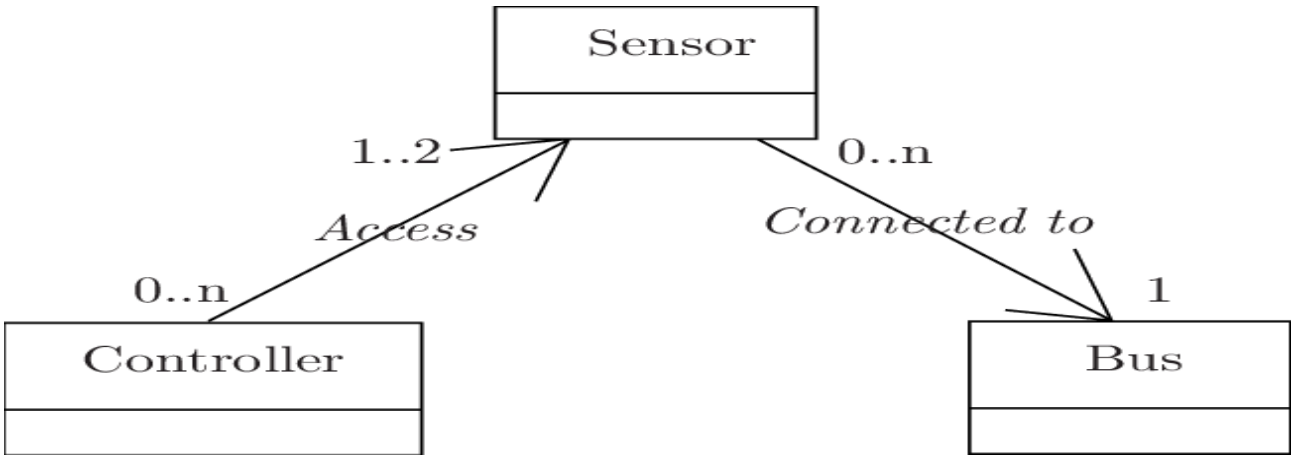
طيب ازاي نكتب العلاقه دي بالكود

```java
class Teacher {
    String name;

    Teacher(String name) {
        this.name = name;
    }

    void teach() {
        System.out.println(name + " is teaching...");
    }
}

class Student {
    String name;
    Teacher teacher; // Association هنا

    Student(String name, Teacher teacher) {
        this.name = name;
        this.teacher = teacher;
    }

    void attend() {
        System.out.println(name + " is attending class with " + teacher.name);
    }
}
```

```java
public class Main {
    public static void main(String[] args) {
        Teacher t1 = new Teacher("Dr. Ahmed");
        Student s1 = new Student("Mostafa", t1);

        s1.attend();   // Mostafa is attending class with Dr. Ahmed
        t1.teach();    // Dr. Ahmed is teaching...
    }
}
```

بص على اليمين في الكلاس الابن ال هو studentجه عمل اوجيكت من الكلاس الاب جوه طب حد يجي يقولي هما كده مش جزء من بعض اقولك لا تعالى معايا بس على الشمال هتلاقيه معرف كل كلاس لوحده مستقل فلو مات كلاس منهم التاني هيكون شغال عادي .

تعالى بقى نشوف بيتمثل على umlازاي



السهم ال على اليمين يعني العلاقه many-to-one

السهم الشمال يعني العلاقه many-to-many

و الAssociationينقسم الى نوعين وهم(Aggregation && composition).

Aggregation

انا مش شايف فرق جوهري بينها وبين Associationالممكن أقوله انها ممكن تفرق بس في المعنى هتقولي ازاي اقولك بص الAggregationبتدل على ملكيه بسيطه للشئ بس ده ميمنعش سوء كان Associationاو Aggregationهما عباره عن كلاسين مستقلين هتقولي مش فاهم اقولك حقك تعالى اقولك مثال

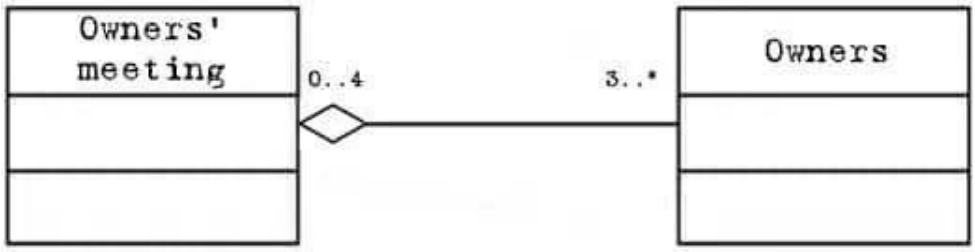
```
class Department {
    String name;
}

class University {
    String name;
    Department dept;  // الجامعة تحتوي على قسم.. لكن القسم ممكن يكون موجود في كذا جامعة
}
```

```
class Teacher {
    String name;
}

class Student {
    String name;
    Teacher teacher;  // الطالب مرتبط بمدرس.. لكن مش معلوك ايه
}
```

بس في المثال ال على اليمين هنا انا استخدمت كلاس ال Teatcherبس المعنى مفهوش أي ملكيه حتى لو بنسبه بسيطه يعنى ولا المدرس ملك الطالب ولا الطالب ملك المدرس وينفع المدرس يكون مستقل وينفع الطالب يكون مستقل هتقولي يعني ايه مستقل اقولك يعني المدرس لو مشى الطالب هيروح عند غيره عادي الدنيا موقفتش يعني.

تعالى بقى على الكود ال على اليمين هنا كلاس ال departemtانا استخدمته في كلاس الكليه ولو لحظت ان الكلام في ملكيه بسيطه يعني الكليه مكونه من مجموعه من الأقسام بس ده ميمنعش برضو ان الكلاسين مستقلين برضو يعني لو الكليه اتقفلت ممكن الأقسام تكون في كب=ليه تانيه عادي.

دلوقت اوريك شكلها بال uml



هو ده بتتمثل بسهم على شكل مفرغ ناحيت الحاجه ال انا مستعملها

Composition

ده نوع خاص من العلاقات فده هو الوحيد التقدر تقول عليه ان كائن بيملك كائين تاني تماما هتقولي بيملكه ازاى اقولك يعني لو الكائن الأساسي وقف او تم حذفه التاني هيتم حذفه لو مش فاهم برضو تعالى عطيك مثال من الواقع:

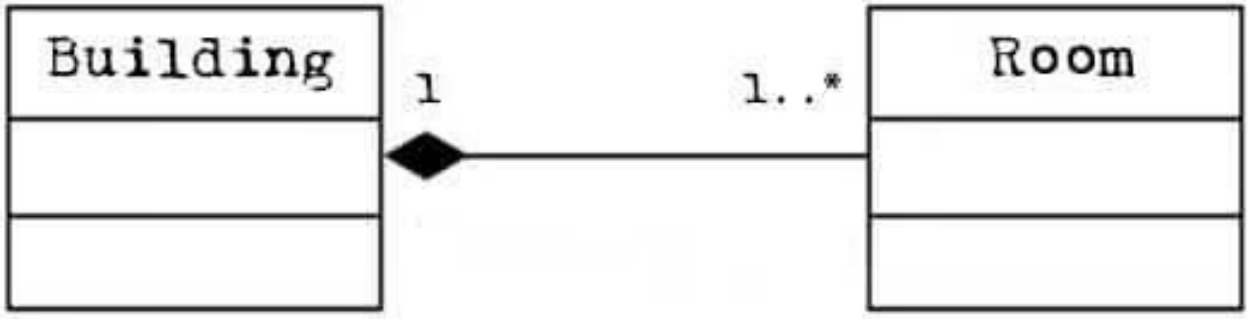
الانسان يمتلك قلب ورئه والخ من الأعضاء طب لو الانسان مات كل الأعضاء دي بتموت هو ده ال composition.

تعالى بقى اقولك ازاى تكتبه كود

```java
class Heart {
    public void beat() {
        System.out.println("Heart is beating...");
    }
}

class Person {
    private Heart heart;

    public Person() {
        heart = new Heart();  // الـ Person بيخلق الـ Heart بنفسه
    }

    public void live() {
        heart.beat();
    }
}
```

بص هنا الفارق الجوهري بينه وبين أي علاقه تانيه هنا جه عرف الكائن جوه الكائن الرئيسى بص عرف توبجيكت من الكلاس heartجوه كلاس ال personعلى خلاف الباقى كان بيعرف الاوبجيكت في ال mainمنفصل وده يدل ان لو الكلاس الأساسي اتحذف التاني ينحذف على طول

تعالى بقى اقولك ازاي هتمثله بال uml

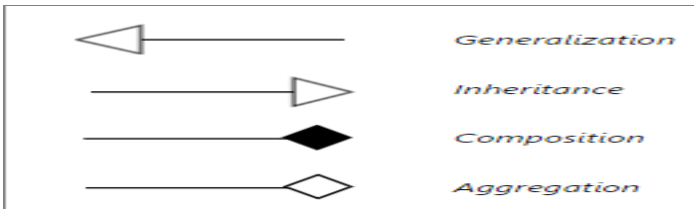


بص هنا مثل السهم بمعين ليس فارغ على عكس ال Aggregation

الملخص

بما اني كلمتك عن كل العلاقات دلوقت تعالى الخصلك الموضوع:

الملكية	يعيش لوحده؟	المثال	نوع العلاقة
لا	نعم	طالب ومدرس	Association
جزئية	نعم	جامعة وأقسام	Aggregation
كاملة	لا	إنسان وقلبه	Composition



دي شكل الأسهم

طب ممكن واحد يجي في دماغه فكرة اني العلاقه دي في جافا بس احب اقولك لا في كل لغات البرمجه الفكره انك تعرف العلاقه الصح بين كل كلاس عندك في البروجيكت وتقدر تنفذها.