

## Лабораторийн ажил 6: Бүрдмэл харьцаа

### Лабораторийн ажлын зорилго:

Бүрдмэл харьцаа, бүтэцлэг болон нийлмэл бүрдмэл харьцааг ашиглах болон гишүүн өгөгдөл, параметр, буцаах утга, үйлдэл дотор ашиглах хэлбэрүүдийг ашиглах.

### Лабораторийн ажлын суралцахуйн үр дүнгүүд:

Энэ лабораторийн ажлыг гүйцэтгэснээр оюутан дараах чадваруудтай болсон байна.

д/д	Суралцахуйн үр дүнгүүд	Суралцахуйн үр дүнг илэрхийлэх үйл үг	Суралцахуйн үр дүнгийн түвшин (Блумын)	CLOs хамаарал
1	Бүрдмэл харьцааг ашиглах	Ашиглах (Use)	Хэрэглээ	2, 8
2	Бүтэцлэг болон нийлмэл бүрдмэл харьцааг ашиглах	Ашиглах (Use)	Хэрэглээ	2, 8
3	Гишүүн өгөгдөл, параметр, буцаах утга, үйлдэл дотор ашиглах хэлбэрүүдийг ашиглах	Ашиглах (Use)	Хэрэглээ	2, 8
4	ЮМЛ диаграм зурах	Зурах (Draw)	Синтез	11
5	Тайлан бичих	Зохион бичих (Compose)	Синтез	12
6	Тайлан хамгаалах	Хамгаалах (defend)	Синтез	12
7	Англи хэл дээр холбогдох материал бусад эх үүсвэрээс унших	Унших (Read)	Ойлголт	13

### Ашиглах програм хангамж/техник хангамж, бусад хэрэглэгдэхүүнүүд:

Лабораторийн компьютер эсвэл өөрийн зөөврийн компьютерийг ашиглана.

### Онолын ойлголтууд

Объект хандлагат програмчлалд классуудын шаталсан харьцаа буюу програм хангамжийн дахин ашиглагдах(Reusability) чадварыг нэмэгдүүлдэг дараах 2 хэлбэр байдаг.

1. Удамшил харьцаа
2. Бүрдмэл харьцаа

Тус лабораторийн агуулгаар Бүрдмэл харьцааны тухай авч үзэх ба Удамшил харьцааны тухай дараагийн лабораториудад судална.

Бүрдмэл харьцааны хамгийн энгийн жишээ: Бүтцийн програмчлал дахь Програм В нь програм А-г дуудах

Жишээ 1:

```
public class A{
    int a=5;
    void b(){}
}

class B{
    B(){
        A a = new A();
        a.a=10;
    }
}
```

Классын гишүүн өгөгдлүүд нь дандаа өгөгдлийн үндсэн төрлийн хувьсагчид байсан. Тэгвэл заримдаа классын гишүүд нь өгөгдлийн хийсвэр төрлийн хувьсагч(классын объект) байж болно. Үүнийг классуудын хоорондын бүрдмэл харьцаа гэнэ. Удамшил харьцаа нь өгүүлбэрзүйн «... бол ... юм» холбоосоор холбогддог бол бүрдмэл харьцаа нь «... бол ..тай» холбоосоор холбогддог.

Composition means HAS A

Inheritance means IS A

Жишээлбэл: Компьютер бол дэлгэцтэй. Хүн гар хөлтэй. Гурвалжин оройнуудтай. Машин кузов, мотор, дугуйтай г.м.

Бүрдмэл харьцаа нь класс доторх объектын хэрэглээ гэж ойлгож болно.

Бүрдмэл харьцааны төрлүүд:

1. Бүтцийн бүрдмэл /Structural composition/ - Класс дотроо өөр классыг агуулах
2. Функцийн бүрдмэл / Behavioral composition/ - Класс нь өөр классын гишүүн функцийг агуулах
3. Холбоо хамаарлын бүрдмэл / Composition by relationship/ - Класс нь класс хоорондын холбоо хамаарлыг өөртөө агуулах

Жишээ 2: **Person** has a **Job**.

// Job классыг тодорхойлъя.

```
class Job {
    private String role;
    private long salary;
    private int id;
```

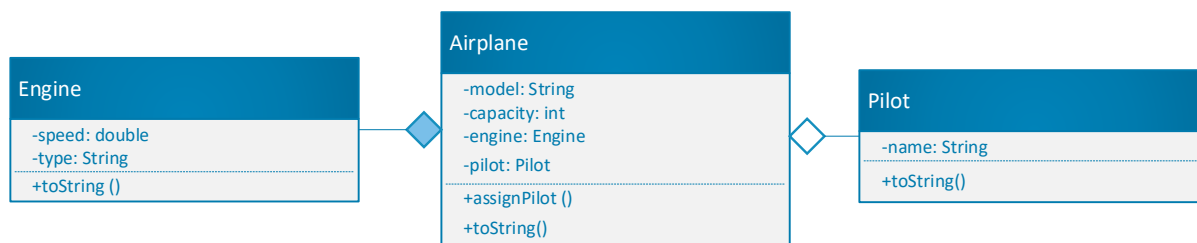
```

    public String getRole() {
        return role;
    }
    public void setRole(String role) {
        this.role = role;
    }
    public long getSalary() {
        return salary;
    }
    public void setSalary(long salary) {
        this.salary = salary;
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
}
// Person классыг тодорхойлъя.
class Person {
    //composition has-a relationship
    private Job job;
    public Person(){
        this.job=new Job();
        job.setSalary(1000L);
    }
    public long getSalary() {
        return job.getSalary();
    }
}
//TestPerson класс
public class TestPerson {
    public static void main(String[] args)    {
        Person person = new Person();
        long salary = person.getSalary();
    }
}

```

```
}
```

Жишээ 3: Онгоц класс нь Engine класстай бүтцийн бүрдмэл(composition) харьцаагаар, Pilot класстай нийлмэл(Aggregation) харьцаагаар тус тус холбогдсон байна.



Зураг 6.1 Класс диаграм

### Pilot классын тодорхойлолт

```
public class Pilot {
    private String name;
    public Pilot(String name){
        this.name = name;
    }
    public String toString() {
        return "Pilot{" + "name=" + name + '}';
    }
}
```

### Engine классын тодорхойлолт

```
public class Engine {
    private double speed;
    private String type;
    public Engine(double speed, String type){
        this.speed = speed;
        this.type = type;
    }
    public String toString() {
        return "Engine{" + "speed=" + speed + ", type=" + type + '}';
    }
}
```

### Airplane классын тодорхойлолт

```
public class Airplane {
    private String model;
    private int capacity;
    //гишүүн өгөгдөл хэлбэрээр
    private Engine engine ;
    private Pilot pilot;
```

```

// байгуулагч функц дотор engine объектыг үүсгэнэ
public Airplane(String model, int capacity)
{
    Engine engine = new Engine(2400.0, "IO-360");
    this.model = model;
    this.capacity = capacity;
    this.engine = engine;
}

//параметр хэлбэрээр ашиглах
public void assignPilot(Pilot pilot){
    this.pilot = pilot;
}

//буцаах утга хэлбэрээр ашиглах
public Pilot getPilot(){
    return pilot;
}

public String toString() {
    return "Airplane{" + "model=" + model + ", capacity=" +
    capacity + ", engine=" + engine + ", pilot=" + pilot + '}';
}
}

```

### **MainProgram классын тодорхойлолт**

```

public class MainProgram {
    public static void main(String args[]){
        Pilot man1 = new Pilot("Bold");
        Pilot man2 = new Pilot("Bayar");

        Airplane airplane = new Airplane("Boeing 747", 350);

        // airplane классын гишүүн үйлдэл рүү объект дамжуулж байна
        airplane.assignPilot(man1);

        //Airplane объектын мэдээллийг хэвлэх ба toString() үйлдэл
        дуудагдана
        System.out.println(airplane);
        // үйлдлийн үр дүнд man1 объектыг буцаах ба man2 объектод уг
        объектын хаягийг олгож байна.
        man2 = airplane.getPilot();
        System.out.println(airplane);
        System.out.println(man2);
    }
}

```

Үр дүн:

```
Airplane{model=Boeing 747, capacity=350, engine=Engine{speed=2400.0, type=IO-360}, pilot=Pilot{name=Bold}}
Pilot{name=Bold}
```

### **Ажил гүйцэтгэх дараалал:**

Өмнөх лаборатори дээр тодорхойлсон классуудыг өргөтгөж өөр бүрдмэл классууд үүсгэнэ. Ингэхдээ:

- A. Дотор нь заавал байх бүтэцлэг бүрдмэл харьцаа үүсгэх классуудыг тодорхойлно.
- B. Дотор нь зарласан боловч бие даасан байдалтай ажиллах боломжтой нийлмэл классууд тодорхойлно.
- C. Зөвхөн нэг үйлдлийг гүйцэтгэхэд оролцож болох нийлмэл классууд тодорхойлно.
- D. Эдгээр классуудыг агуулсан класс диаграмуудыг зурна. Мөн main үйлдэл доторх объектуудаар объект диаграм зурна.

### **Суралцахуйн үр дүнг үнэлэх даалгаврууд:**

Өмнөх лабораторийн ажил дээр үүсгэсэн класс дээрээ нэмж дараах төрлийн бүрдмэл харьцааг хэрэгжүүлнэ. Үүнд:

A. Бүтэцлэг бүрдмэл харьцаа хэрэгжүүлэх. Ингэхдээ дараах хэлбэрүүдийг ашиглана.

- гишүүн өгөгдөл хэлбэр
- параметр хэлбэр
- буцаах төрөл хэлбэр

B. Нийлмэл (has a) бүрдмэл харьцааг хэрэгжүүлэх

- гишүүн өгөгдөл хэлбэр
- параметр хэлбэр
- буцаах төрөл хэлбэр

C. Нийлмэл (uses) бүрдмэл харьцааг хэрэгжүүлэх

- параметр хэлбэрээр ашиглах
- үйлдэл дотор объект үүсгэх хэлбэрээр ашиглах

### **Ашиглах материал:**

- Ю.Намсрай, Т.Гантөр, Д.Ганцоож, Програмчлалын Жава хэл, 2015 он.
- Д.Энхжаргал, Java 2 Объект хандлагат програмчлал, 2009 он.
- Б.Батзолбоо, UML 2: Объект хандлагат шинжилгээ ба түүний хэрэглээ, 2015 он.