

## Лабораторийн ажил 13-14: Классын хийсвэрлэлийн төрлүүд

### Лабораторийн ажлын зорилго:

Интерфейс класс, хийсвэр класс, бодит класс, удамшсан классын ялгааг ойлгож, 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	Интерфейс класс, классыг ашиглах	Ашиглах (Use)	Хэрэглээ	4, 10
2	Хийсвэр классыг ашиглах	Ашиглах (Use)	Хэрэглээ	4, 10
3	Бодит класс, удамшсан классыг ашиглах	Ашиглах (Use)	Хэрэглээ	4, 10
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 програмчлал ба Хийсвэрлэл

Классыг “abstract” түлхүүр үгийг ашиглан зарлаж болдог ба үүнийг хийсвэр класс гэнэ. Хийсвэр класс нь хийсвэр аргуудыг агуулсан байж болно. Агуулаагүй ч байж болно. Мөн хийсвэр классууд нь тодорхой (их биетэй) аргуудтай байж болно. Бүр эсвэл ямар нэгэн хэрэгжүүлэлт хийгдэггүй (угалзан хаалт байхгүй шууд цэгтэй таслалаар төгссөн) аргуудтай байж болно. Хийсвэр класс нь ямар нэгэн тодорхой зүйлийг хийдэггүй. Тиймээс хийсвэр классын объект үүсгэж болохгүй.

#### Хийсвэр классыг зарлах

Класс зарлах үедээ өмнө нь “abstract” түлхүүр үгийг тодорхойлж өгдөг. Өөрөөр хэлбэл “class” түлхүүр үгийн өмнө байрлах “abstract” түлхүүр үг нь хийсвэр класс гэдгийг илэрхийлдэг. Дараах жишээ кодоос хийсвэр классын бүтэц болон зарлалтыг харна уу.

```
// Хийсвэр классыг зарлахдаа abstract түлхүүр үгийг ашиглана
abstract class AbstractDemo{
```

```
// Тодорхой арга: Их бие болон хаалттай (угалзан)
public void myMethod() {
    //Энд аргын хийх үйлдэл бичигдэнэ
}

// Хийсвэр арга: Их бие болон хаалт (угалзан) байхгүй
abstract public void anotherMethod();
}
```

Хийсвэр класс нь тодорхой аргыг агуулж болдог болохоор 100% хийсвэрлэлийг бий болгодоггүй. Өөрөөр хэлбэл хийсвэр класс нь хэсэгчлэн хийсвэрлэх боломжийг олгодог.

Объект хандлагат технологид хийсвэрлэлийг хэрэгжүүлэх хэд хэдэн аргууд байдаг. Тэдгээрийн нэг нь интерфэйс юм. Интерфэйсийг ашиглах юм бол 100% хийсвэрлэлийг (full abstraction) бий болгоно.

#### Мэдэхэд илүүдэхгүй 2 дүрэм:

1. Хэрэв класс нь цөөн хийсвэр арга болон цөөн тодорхой аргатай бол хийсвэр классаар зарлана.
2. Хэрвээ класс нь зөвхөн хийсвэр аргуудтай бол интерфэйсээр зарлана.

#### Анхааруулга – Хийсвэр классын объект үүсгэж болохгүй

Дээр хийсвэр классын объект үүсгэж болохгүй гэж дурьдаж байсан. Дараах жишээ кодонд хийсвэр классын объект үүсгэж болохгүй гэдгийг харуулж байна. Өөрөөр хэлбэл классын объект үүсгэх кодын мөрөн дээр алдаа заана.

```
abstract public class AbstractDemo{
    public void myMethod() {
        System.out.println("Hello");
    }
    abstract public void anotherMethod();
}

public class ConcreteDemo{

    public void anotherMethod() {
        System.out.print("Abstract method");
    }

    public static void main(String args[])
    {
        //Хийсвэр классын объект үүсгэж болохгүй - Алдаа!
        AbstractDemo obj = new AbstractDemo();
    }
}
```

```
obj.display();  
}  
}
```

#### Гаралт:

Unresolved compilation problem: Cannot instantiate the type  
AbstractEx1

Хийсвэр классыг класст удамшуулахад түүний бүх хийсвэр аргуудыг хэрэгжүүлэх боломжтой байдаг. Эсвэл хийсвэр аргуудыг тухайн класст даран тодорхойлох боломжтой байдаг.

#### Яагаад хийсвэр класс хэрэгтэй вэ?

Нэгэн жишээг тайлбарлая. Animal класс болон хоорондоо төстэй Cat, Dog, Horse гэсэн классууд байсан гэж үзье. Эдгээр классууд нь Animal классыг өргөтгөж нийтлэг шинжүүдийг нь авна. Өөрөөр хэлбэл Animal классаас удамшина. Дээрх жишээг сайн бодоод үзвэл Animal классын объектыг үүсгэх нь ач холбогдолгүй гэдэг нь харагдана. Animal класс нь амьтны ерөнхий шинжийг дүрслэх болохоос бүх амьтныг тодорхойлж дүрслэж чадахгүй. Иймээс энэ жишээнд ерөнхий хийсвэр класс үүсгээд дараа нь тодорхой (Cat, Dog, Horse) классуудыг өргөтгөж үүсгэнэ. Өргөтгөсөн классууд нь хийсвэрлэлийн дүрмийн дагуу аргуудыг даран тодорхойлно. Мөн өөрийн буюу өөрт хэрэгтэй аргуудыг нэмж тодорхойлно.

#### Хийсвэр класс болон Тодорхой классын ялгаа

Хийсвэр биш классыг тодорхой класс (**Concrete class**) гэдэг. Тодорхой классыг өөрөөр бодит класс гэж хэлж болно. Дээрх үзүүлсэн жишээгээр Animal класс нь хийсвэр класс, Cat, Dog, Horse зэрэг классууд нь тодорхой класс.

#### Гол ойлголт:

1. Хийсвэр классаас ямар нэгэн класс удамшуулаагүй л бол тухайн хийсвэр класс ямар ч ашиглалтгүй байна.
2. Хэрвээ хийсвэр арга зарласан бол хийсвэр класс зарлах ёстой. Хийсвэр арга хийсвэр биш класст байж болохгүй. Гэхдээ хийсвэр класс нь хийсвэр аргагүй байж болно.
3. Хийсвэр класс нь хийсвэр биш (concrete) аргуудтай байж болно.

#### Хийсвэр арга

Хийсвэр аргын талаар дээр дурьдсан учраас одоо хэрхэн зарлах, ямар ямар гол ойлголтууд байдаг талаар дурьдъя.

#### Хийсвэр аргыг зарлах (синтакс):

```
public abstract void display();
```

#### Хийсвэр аргын талаарх санаамж:

1. Хийсвэр арга нь их биегүй байна.
2. Хийсвэр аргын зарлагаа нь үргэлж цэгтэй таслалаар төгсөнө.

3. Хийсвэр аргыг даран тодорхойлох ёстой.
4. Хийсвэр арга нь хийсвэр класст байх ёстой.

Жич: Хийсвэрийг классыг өргөтгөх үед бүх хийсвэр аргуудыг даран тодорхойлох ёстой.

#### Хийсвэр класс болон хийсвэр аргын жишээ програм

```
abstract class Demo1{
    public void disp1(){
        System.out.println("Concrete method of abstract class");
    }
    abstract public void disp2();
}

class Demo2 extends Demo1{
    /*
        Demo1 класст их биегүй хийсвэр арга тодорхойлж өгсөн. Хэрвээ их
        биегүй аргыг Demo2 класст даран тодорхойлж өгөхгүй бол хөрвүүлэх үед
        алдаа заана.
    */
    public void disp2()
    {
        System.out.println("Энэ бол даран тодорхойлогдсон арга.");
    }
    public static void main(String args[]){
        Demo2 obj = new Demo2();
        obj.disp2();
    }
}
```

#### Гаралт:

Энэ бол даран тодорхойлогдсон арга.

#### Ажил гүйцэтгэх дараалал:

1. Интерфейс үүсгэх,
2. Интерфейсээс хийсвэр класс удамших,
3. Хийсвэр класс үүсгэх,
4. Хийсвэр классаас бодит класс удамших,
5. Бодит классаас удамшсан класс удамших.

## Суралцахуйн үр дүнг үнэлэх даалгаврууд:

1. Хүн (Person) гэдэг класс үүсгэнэ. Person класс нь хүний нэр (name), нас (age) гэх мэт гишүүн өгөгдлүүдтэй байна. Гишүүн өгөгдлүүдээ байгуулагч аргаараа дамжуулан авдаг байна. Нэр, нас гэсэн гишүүн өгөгдлүүдэд утга олгох болон тэдгээрийг буцаах гишүүн аргуудтай байна. Нэмэлт гишүүн өгөгдөл болон аргууд тодорхойлж болно.

2. Person классаас Student, Teacher, Professor гэсэн классууд удамшуулна.

3. Дараах классуудыг тодорхойлж, класс тус бүр дээр объект үүсгэнэ үү. Бүх гишүүн өгөгдлүүд private харагдалттай байна .

```
class Person
```

```
{    Гишүүн өгөгдлүүд (Инстэнс хувьсагчид):
```

```
    Нэр Нас
```

```
    Гишүүн функцүүд:
```

```
    Байгуулагч
```

```
    Бүх гишүүдийг хэвлэх display() функц
```

```
}
```

```
class Sportsman extends Person
```

```
{    Гишүүн өгөгдлүүд (Инстэнс хувьсагчид):
```

```
    Биеийн жин Биеийн өндөр
```

```
    Гишүүн функцүүд:
```

```
    Байгуулагч
```

```
    Хэвлэх display() функц (биеийн жин, биеийн өндрийг хэвлэх ба эх классын display  
    функцийг дарж тодорхойлно.)
```

```
}
```

```
class TennisPlayer extends Sportsman
```

```
{    Гишүүн өгөгдлүүд (Инстэнс хувьсагчид): Тамирчны харъяалагддаг клубын нэр
```

```
    Тамирчны спортын зэрэг
```

```
    Гишүүн функцүүд:
```

```
    Байгуулагч
```

```
    Хэвлэх display() функц (клубын нэр, спортын зэргийг хэвлэх ба эх классын display  
    функцийг дарж тодорхойлно.)
```

```
}
```

```
class Teacher extends Person
```

```
{    Гишүүн өгөгдлүүд (Инстэнс хувьсагчид):
```

```
    Багшийн албан тушаал Багшийн цалин
```

```
    Гишүүн функцүүд:
```

```
    Байгуулагч
```

Хэвлэх display() функц (Ажилтны албан тушаал, цалингийн утгыг хэвлэх ба эх классын display функцийг дарж тодорхойлно.)

```
}  
class Demo  
{  
    public static void main(String []args)  
    {  
        // Person төрөлтэй 4 объект хадгалах боломжтой массив үүсгэ.  
  
        // Массивын элемент тус бүрт нэг хүн, нэг тамирчин, нэг теннисний тамирчин,  
        нэг багш үүсгэн олгоно уу.  
  
  
        // Бүх объектын хэвлэх функцийг дуудаж, объект тус бүрийн бүх мэдээллийг  
        хэвлэ.  
    }  
}
```

#### **Ашиглах материал:**

- Д.Энхжаргал, Java 2 Объект хандлагат програмчлал, 2009 он.