



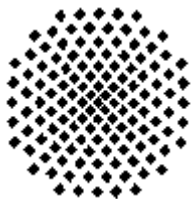
Монгол Улсын Их Сургууль  
Мэдээллийн Технологийн Сургууль  
Dipl.-Ing. Dipl.-Inf. Б.Батням



## Програм хангамжийн инженерчлэлийн үндэс

Хаврын улирал 2019

**ХБНГУ-ын Стүтгарт хотын Их Сургуулийн Автоматжуулалт, Програм хангамжийн технологийн институтын лекцийн материалыг зохиогчийн зөвшөөрөлтэйгээр ашиглав.**



**[www.ias.uni-stuttgart.de](http://www.ias.uni-stuttgart.de)**



## § 3 Объект хандлагат шинжилгээний статик ойлголт

3.1 Статик болон динамик ойлголтын харьцуулалт

3.2 Холбоос

**3.3 Бүрдмэл болон Нийлмэл-Composition**

3.4 Удамшил

3.5 Багц

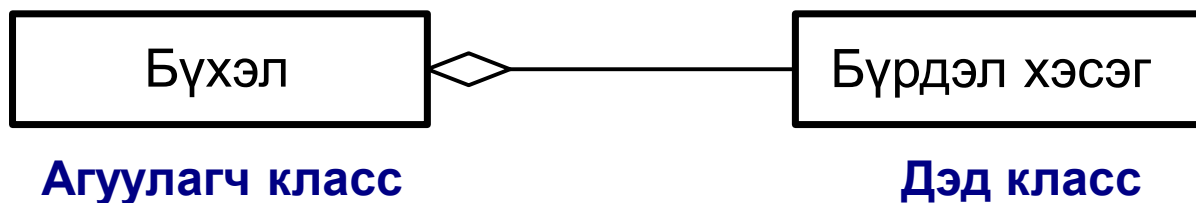
3.6 ЗНХ-ийг өргөтгөсөн механизм (Бие даалт)

3.7 Дүгнэлт



## Бүрдмэл

**Тодорхойлолт:** Бүрдмэл нь класс хоорондын “Бүрдэл-Бүхэл”- (whole-part) эсвэл “бүрдэл хэсэг нь”- (is-a part of) гэсэн харьцаа юм



- Бүрдэл нь холбоосын онцгой нэг хэлбэр юм
- **–ийн бүрэлдэхүүн хэсэг** ө.х. **–ээс тогтдог** (Бүхэл – хэсэг) гэж тодорхойлогдож болно
- Ромбоор бүхлийг тэмдэглэдэг

## Жишээ:



## Бүрдмэлийн онцлог шинж

### – Тэгш хэмт бус

хэрэв В нь А-гийн бүрдэл бол, А нь В-гийн бүрдэл байж болохгүй

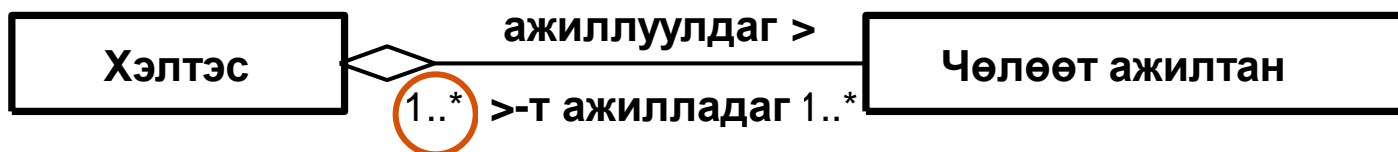
### – Дамжих чанар

хэрэв А нь В-гийн мөн В нь С-гийн бүрдэл бол, А нь С-гийн бүрдэл



### – Заавал тусгаарласан байх албагүй

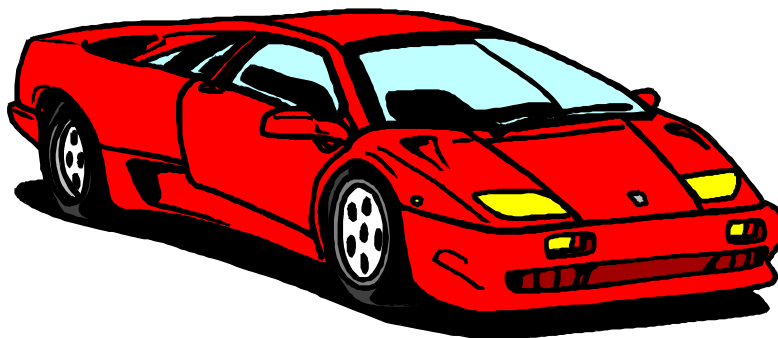
В нь нэгэн зэрэг А болон С-гийн бүрэлдэхүүн байж болно



### – Бүхэл нь бүрэлдэхүүн хэсгийнхээ даалгаварыг орлон авдаг

## Бүрдмэлийн шилжих шинж (Бичиж авах)

– Жишээ Суудлын тэрэг:

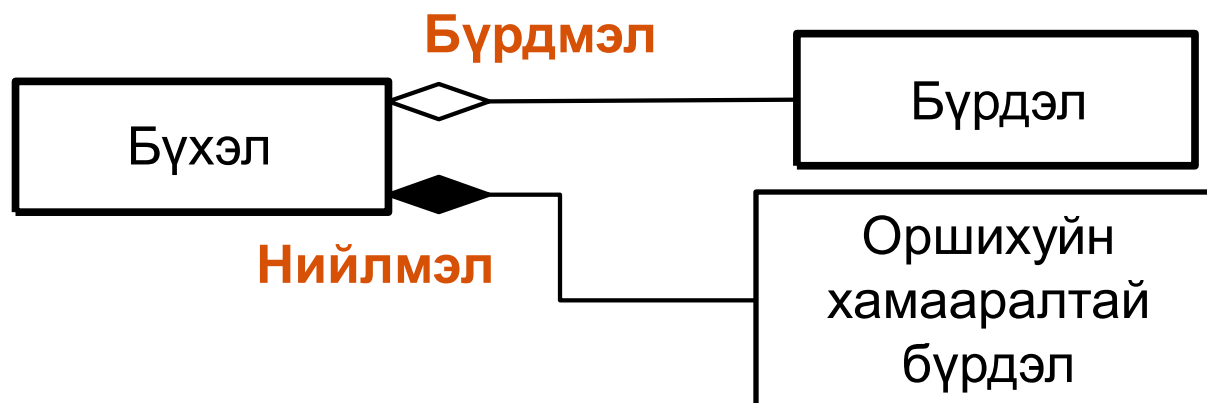


## Бүрдмэл нь хэзээ өгөгдсөн байх вэ?

- Дараах асуултад тийм гэж хариулагдвал бүрдмэл өгөгдсөн байна:
  1. ”-ний бүрэлдэхүүн” гэсэн тодорхойлолт таарч байна уу?
  2. “Бүхэл хэсэг”-т хийгдэх зарим нэг үйлдэл “Бүрэлдэхүүн” хэсэгт бас автоматаар хэрэглэгдэж байна уу?
  3. “Бүхэл хэсэг”-ийн зарим нэг шинж нь бүх эсвэл нэлээд хэдэн “Бүрэлдэхүүн” хэсэгт нөхөн үржиж байна уу?
  4. Холбоо нь “Бүрэлдэхүүн” хэсгүүд нь “Бүхэл хэсэг”-тээ захирагдсан, тэгш хэмт бус шинжээр илэрсэн байна уу?
- Бүрдмэл хамаарлын жишээ
  - Бүхэл болон түүний бүрэлдэхүүн  
**ж. нь: Тэрэг (Бүхэл) болон Мотор (Бүрдэл)**
  - Сав болон түүний агуулга  
**ж. нь: Кофений машин (Сав) мөн Кофений нунтаг (Доторх зүйл)**
  - Иж бүрдэл болон түүний гишүүд  
**ж. нь: Компани (Иж бүрдэл) болон Ажилчид (Гишүүн)**

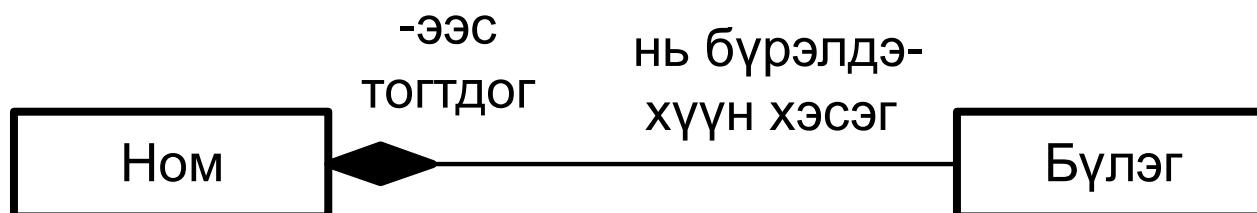
## Нийлмэл

**Тодорхойлолт:** Нягт, хүчтэй холбоо хэлхээтэй бүрдлийг (Оршихуйн хамааралтай) **Нийлмэл** гэдэг



– дүүргэсэн ромбон тэмдэглэгээ

**Жишээ:**



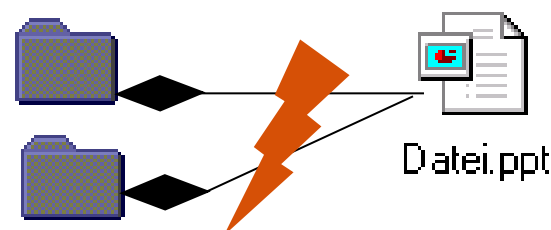
## Нийлмэлийн онцлог шинж

Бүрдэл дээр нэмэгдэн:

- Бүрдэл классын объект бүр нь хугацааны тодорхой нэг агшинд агуулагч классын объектын цорын ганц бүрэлдэхүүн байна
  - Агуулагч классын харьцааны тоо  $\leq 1$
  - Бүрдэл нь өөр нэг бүхэлд харьяалагдаж байж ч магадгүй (гэхдээ яг нэгэн зэрэг биш)
- Бүхлийн динамик агуулга зүй нь бүрэлдэхүүн хэсгүүдэд бас л хүчитэй (propagation semantics)

**Бүхлийг хуулбал, бүрдэл нь мөн хуулагдана**

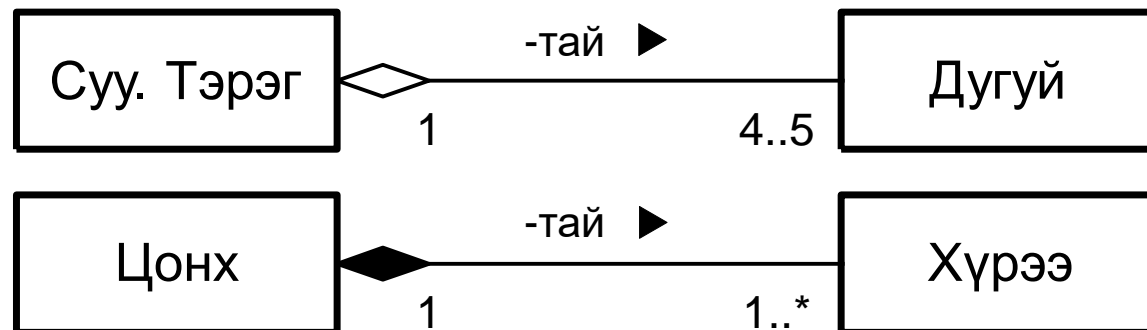
- Жишээ Хавтас болон файл:



**Нэгэн зэрэг боломжгүй!**



## Бүрдмэл, нийлмэлийн харьцуулалт



#### – Бүрдмэл: “Суудлын тэрэг дугуйтай”

- Дугуйнууд нь ямар нэгэн тэрэгт шаардлагатай учир хамаарна

#### Бүрдмэл

- Дугуйнууд нь харин өөрөө бие даасан мөн тэрэгний хооронд солиж болно гэж үзэж болно. **Нийлмэл биш**

#### – Нийлмэл: „Цонх хүрээтэй”

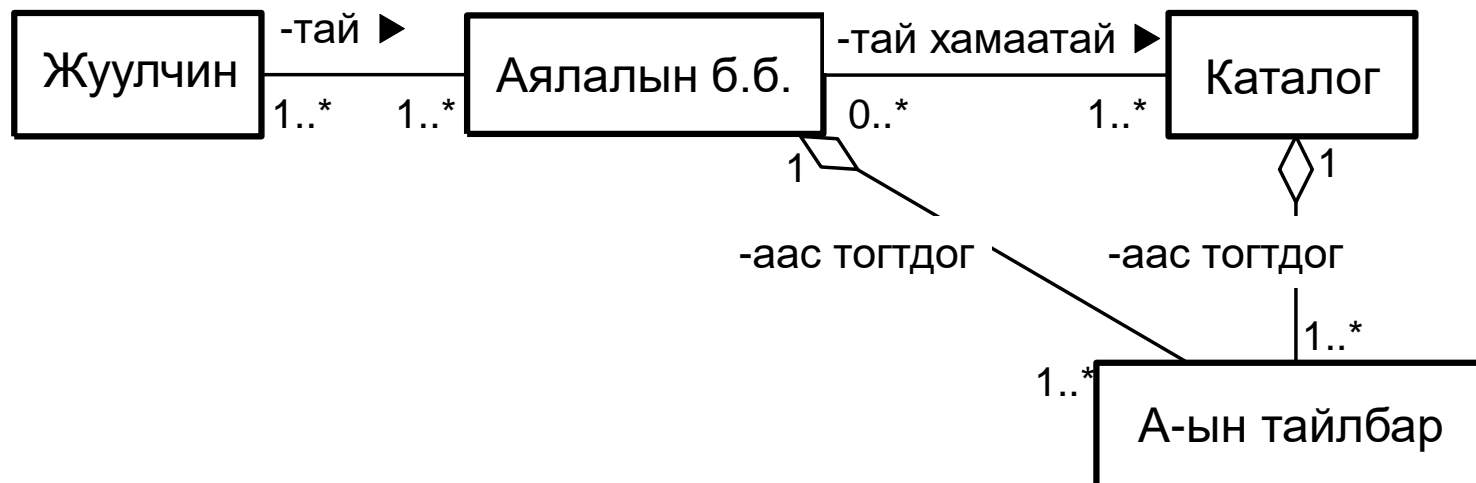
- Хэрэв цонх уствал, оршихуйн хамааралтай бүх дэд хэсгүүд цуг устана

#### Нийлмэл

## Жишээ Аялал жуулчлалын компани

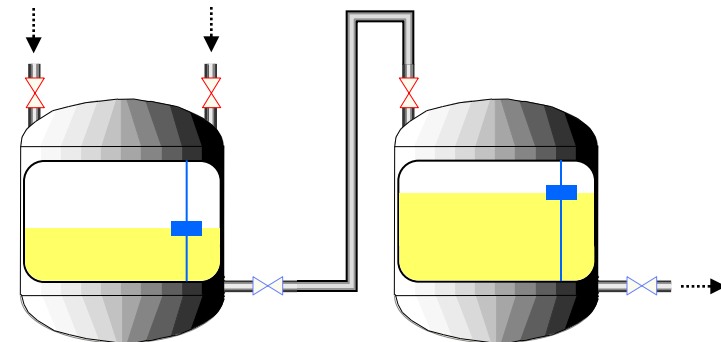
Аялал жуулчлалын компани өөрийн аялалын каталог, аялалын тайлбар болон аялалын бичиг баримтад зориулж классын загварыг боловсруулжээ. Мэдээж аялалд бүртгүүлсэн жуулчинг бас нэг классаар тодорхойлно.

- Аялалын каталог танилцуулга нь нэг эсвэл олон тооны аялалын тайлбарыг агуулна.
- Жуулчинд зориулсан аялалын бичиг баримт нь тус тусын аялалын тайлбар, магадгүй төрөл бүрийн каталогоос бүрдэнэ.
- Олон жуулчид аялалын бичиг баримтыг дундаа авч болно.



## Жишээ Торхны систем

- Торхны систем нь олон торх, оролт гаралтын крант, түвшинг хэмжигч болон дамжуулах хоолойноос тогтоно
- Торх бүр нэгээс хоёр оролтын, нэг гаралтын крант болон түвшин хэмжигч тэй.
- Ялгаатай төрлийн хоёр крант бүрийг дамжуулах хоолойгоор хооронд нь холбосон байж болно.



### 3.3-ийн асуулт

1. Өрөө болон байшин гэсэн класс өгөгджээ. Ямар төрлийн холбоосыг энэ хоёрын хооронд загварчлах вэ?

☐ “-тай холбоотой” гэх энгийн холбоос

☐ “-аас тогтоно” гэх бүрдмэл

☒ “-аас тогтоно” гэх нийлмэл

**Үндэслэл:** Байшин болон өрөө хоёрын дунд оршихуйн хамааралтай

2. Байшин болон хөлслөгч гэсэн класс өгөгджээ. Ямар төрлийн холбоосыг загварчлах вэ?

☒ “тай” гэх энгийн холбоос

☐ “-аас бүрэлдэн тогтоно” гэх бүрдмэл

☐ “-аас тогтоно” гэх нийлмэл

**Үндэслэл :** Байшин болон хөлслөгч хоорондын холбоос тийм ч гүн биш. Бүрдэл бас байж болох юм.



## § 3      **Объект хандлагат шинжилгээний статик ойлголт**

3.1 Статик болон динамик ойлголтын харьцуулалт

3.2 Холбоос

3.3 Бүрдмэл болон Нийлмэл-Composition

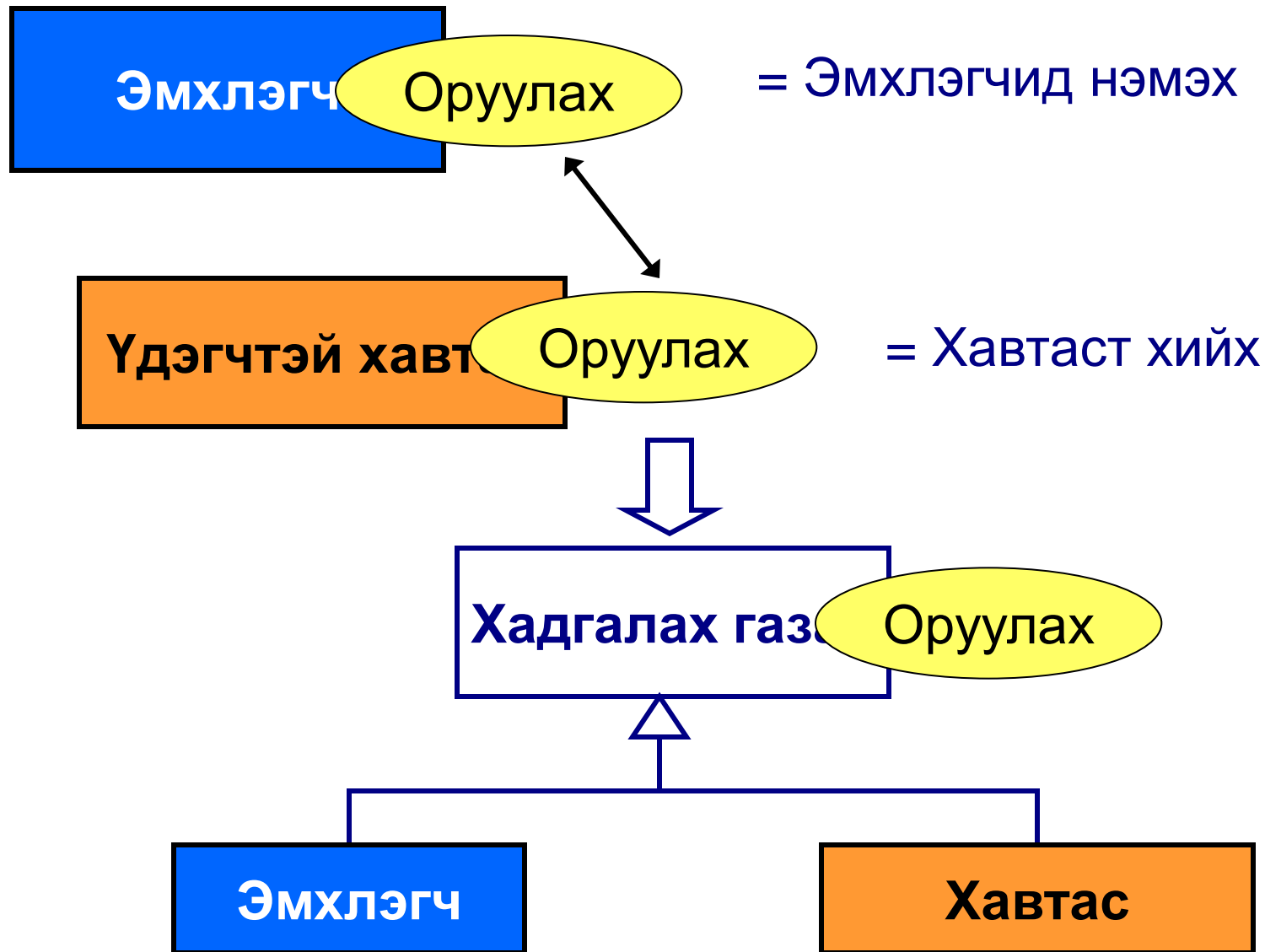
**3.4 Удамшил**

3.5 Багц

3.6 ЗНХ-ийг өргөтгөсөн механизм (Бие даалт)

3.7 Дүгнэлт



**Объектын төстэй зан төлөв (Бичиж авах)**

### Удамшил гэж юу вэ?

**Тодорхойлолт:** **Удамшил** (*generalization*) нь ерөнхий класс болон нарийвчилсан (задаргаа хийгдсэн) класс хоорондын харьцаа юм.

### Удамшил нь зөвхөн хоёр классын хооронд тохиолддог!

- Удамшил нь шаталсан эрэмбэтэй бүтцэд оруулах хийсвэрлэлтийн зарчим юм (Классыг шаталсан бүтцэд оруулан эрэмбэлсэн)
- Зорилго: Нийтлэг онцлох шинж болон зан төлөвийг нэгтгэн нийлүүлэх
- Нарийвчилсан класс нь ерөнхий класстайгаа бүрэн нэг хэвтэй, харин нэмэгдэл мэдээлэлтэй (Шинж, Үйлдэл, Холбоос) байж болно

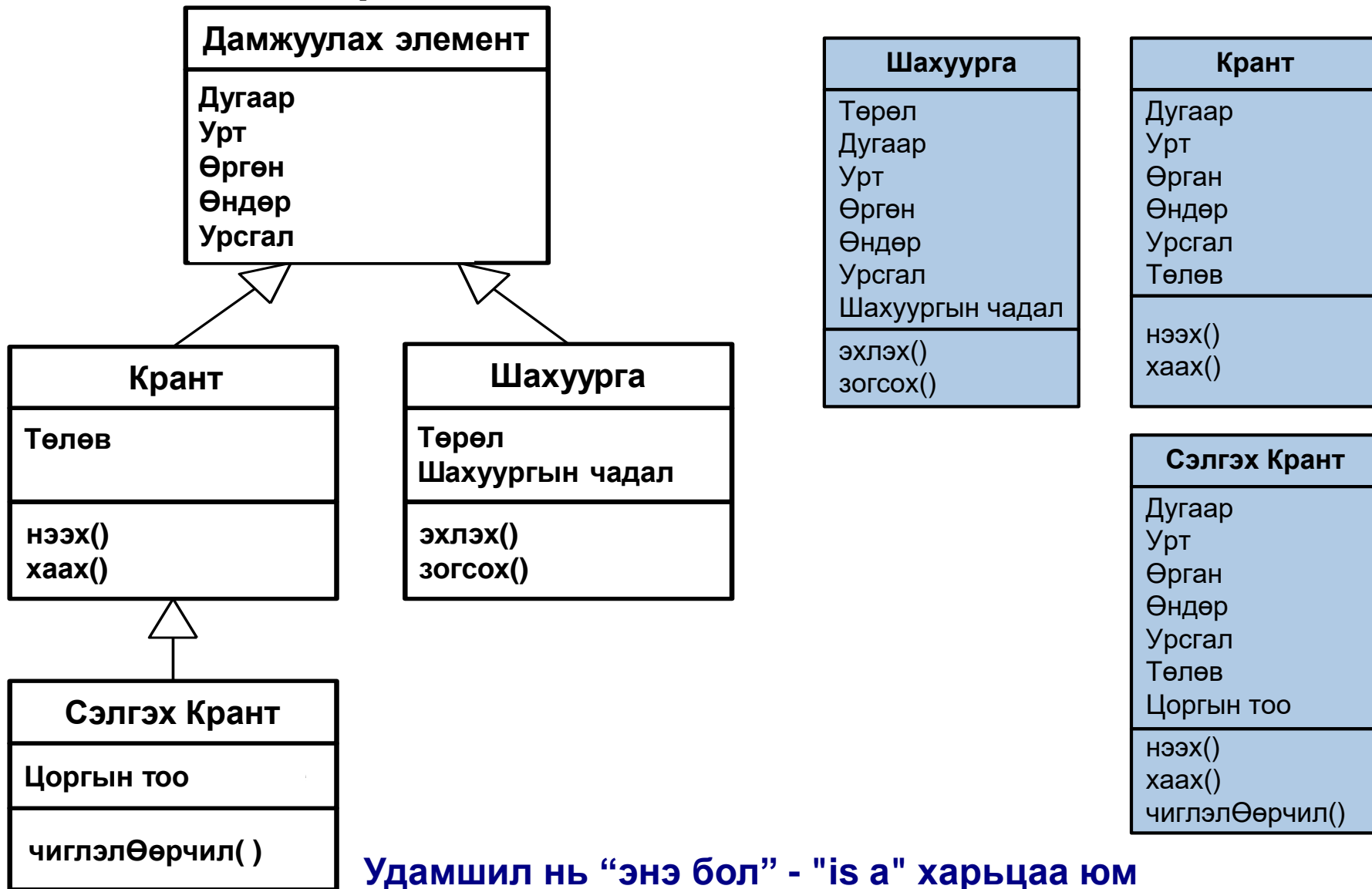
- Ерөнхий класс = Эх класс (*super class*)

- Наривчилсан класс = Дэд класс (*sub class*)



**Орлуулах зарчим:** Эх классын объект зөвшөөрөгдсөн байрлал **бүрт** дэд классын объект хэрэглэгдэж болно, харин эсрэгээрээ биш!

## Удамшилын бүтцийн жишээ

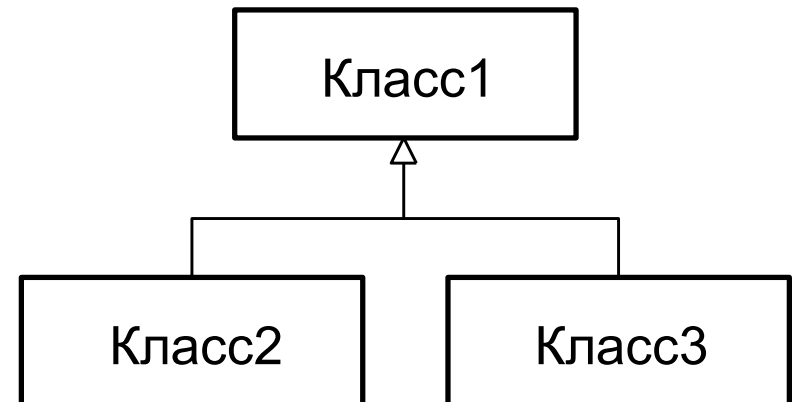
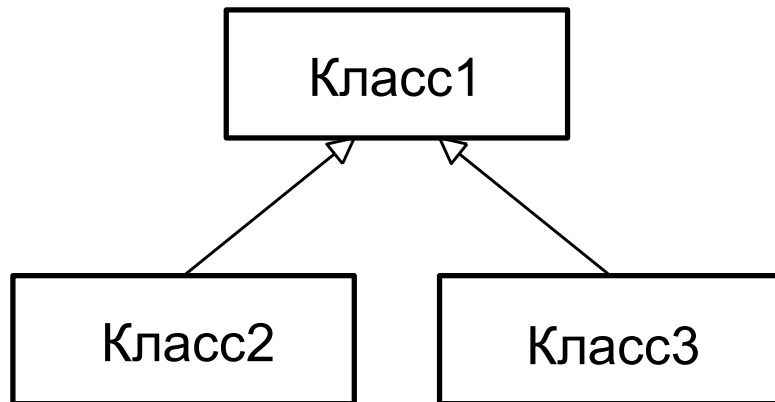


Удамшил нь “энэ бол” - "is a" харьцаа юм



## Удамшилын ЗНХ-ийн тэмдэглэгээ

- Цагаан ө.х. тунгалаг гурвалжин, эх класс талд

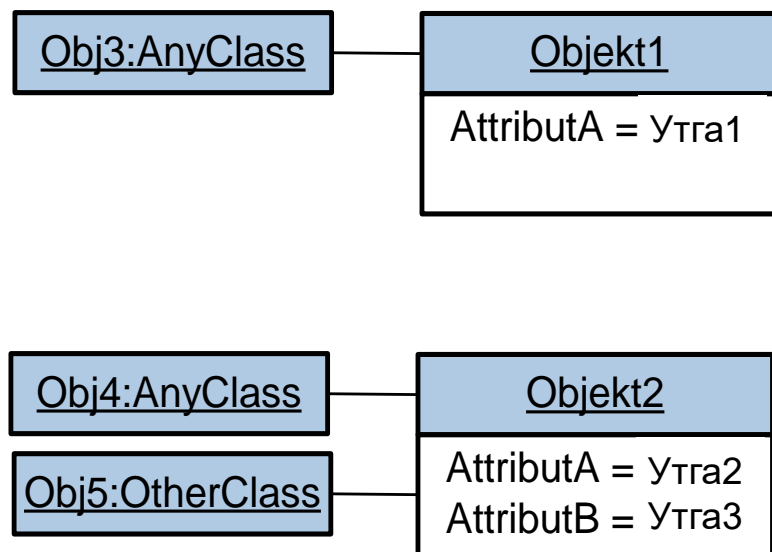


**Дүрлэлийн хэлбэр нь адил утгатай → Хувилбарыг хэрэглэж болно**

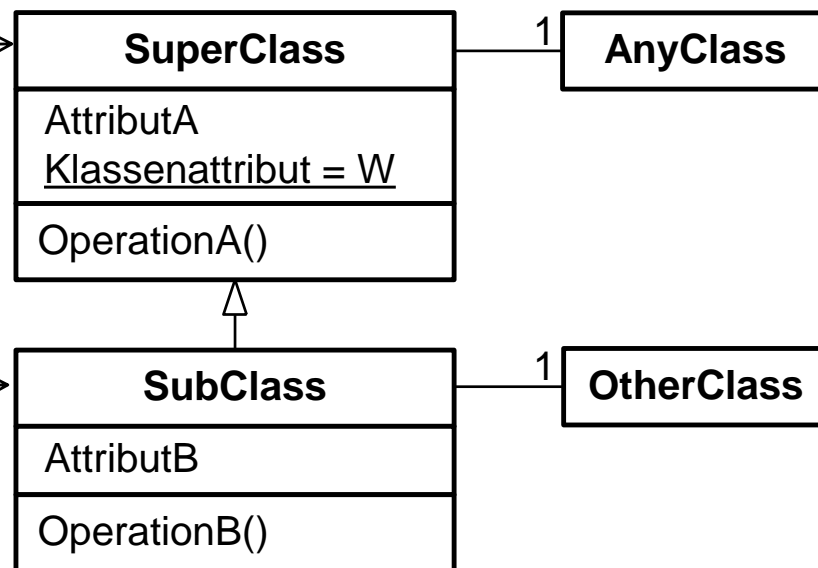
## Юу удамшдаг вэ?

- Үйлдэл, шинж болон холбоос

**Objectdiagramm**



**Classdiagramm**



1. Superclass –ийн шинж A нь Subclass –д удамшсан
2. Superclass –ийн OperationenA() –г мөн Subclass –д хэрэглэж болно
3. Superclass-ийн W гэсэн утгатай классын шинж Subclass –д
4. Superclass болон AnyClass хоорондын холбоос нь Subclass –д

## Үйлдлийг дарж бичих

- Дэд класс нь өөрийн эх классын зан төлөвийг сайжруулж, дахин тодорхойлж ө.х. дарж бичиж болно (*redefine, override*)
- Үйлдэл нь дэд классад дарагдан бичиж болно, харин хасагдахгүй

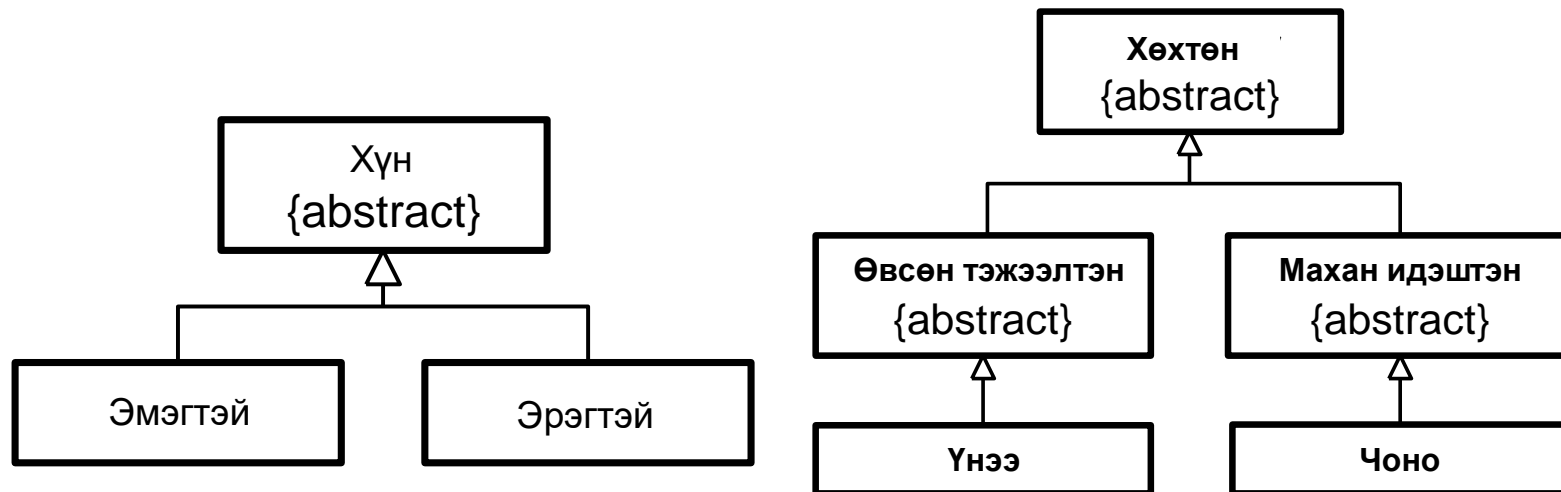
### Үйлдлийн ижил сигнатур шаардлагатай!



## Хийсвэр класс

- Хийсвэр болон бодит классыг хооронд ялгах
- Хийсвэр классаас объект үүсгэж болохгүй
- Өөрийн мэдээллийг нарийвчилсан класст удамшуулахад хэрэглэдэг
- Тэмдэглэгээ: *налуу* нэр эсвэл **{abstract}** шинж тэмдэг
- Хийсвэр класс хийсвэр үйлдлийг агуулж болно. Хийсвэр үйлдлийг дэд класст хэрэгжүүлэх ёстой.

### Удамшилгүй хийсвэр класс нь утга учиргүй



## Полиморфизм

### Тодорхойлолт:

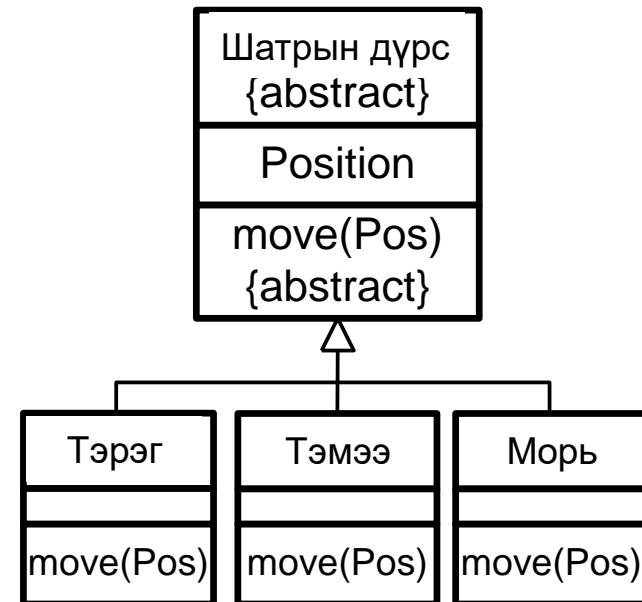
Хэрэв зурвас нь өөр өөр утга зүйтэй аргыг ялгаатай объектын тусламжтайгаар идэвхжүүлбэл, зурвас нь **полиморф** байна.

- Полиморф гэдэг нь ижил зурваст янз бүрийн объектын гаргах ялгаатай зан төлөвийг хэлнэ

### Жишээ

- move(Position) аргыг өөр өөр шатрын дүрсийн хувьд ялгаатай хэрэгжүүлсэн
  - Тэрэг: Чиг шулуун
  - Тэмээ: Зөвхөн ташуу
  - Морь: 1 нэг нүд чигээрээ, 1 нүд ташуу

⇒ Ижил зурвас move(Position) ялгаатай хариу үйлдлийг өдөөдөг



## Полиморфизмийн ажиллах зарчим (Бичиж авах)

- Жишээ Шатар



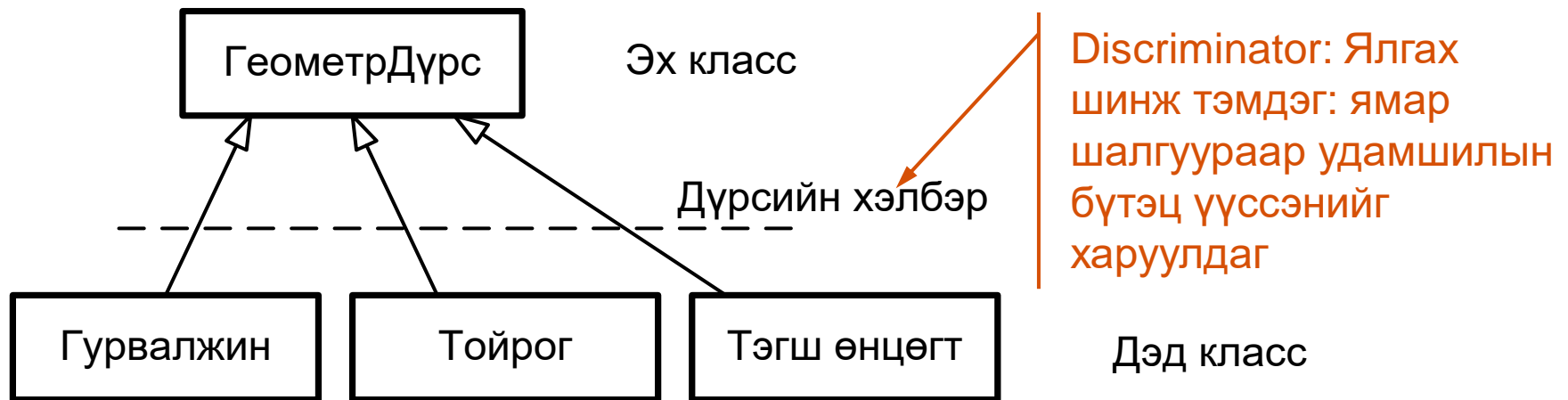
минийДүрс = (:Тэрэг, :Хүү, :Морь, :Тэмээ, :Тэрэг, ...)

move(Pos)

- Арга move(Pos) → ялгаатай хариу үйлдэл
- ж. нь минийДүрс[2].move(а3)

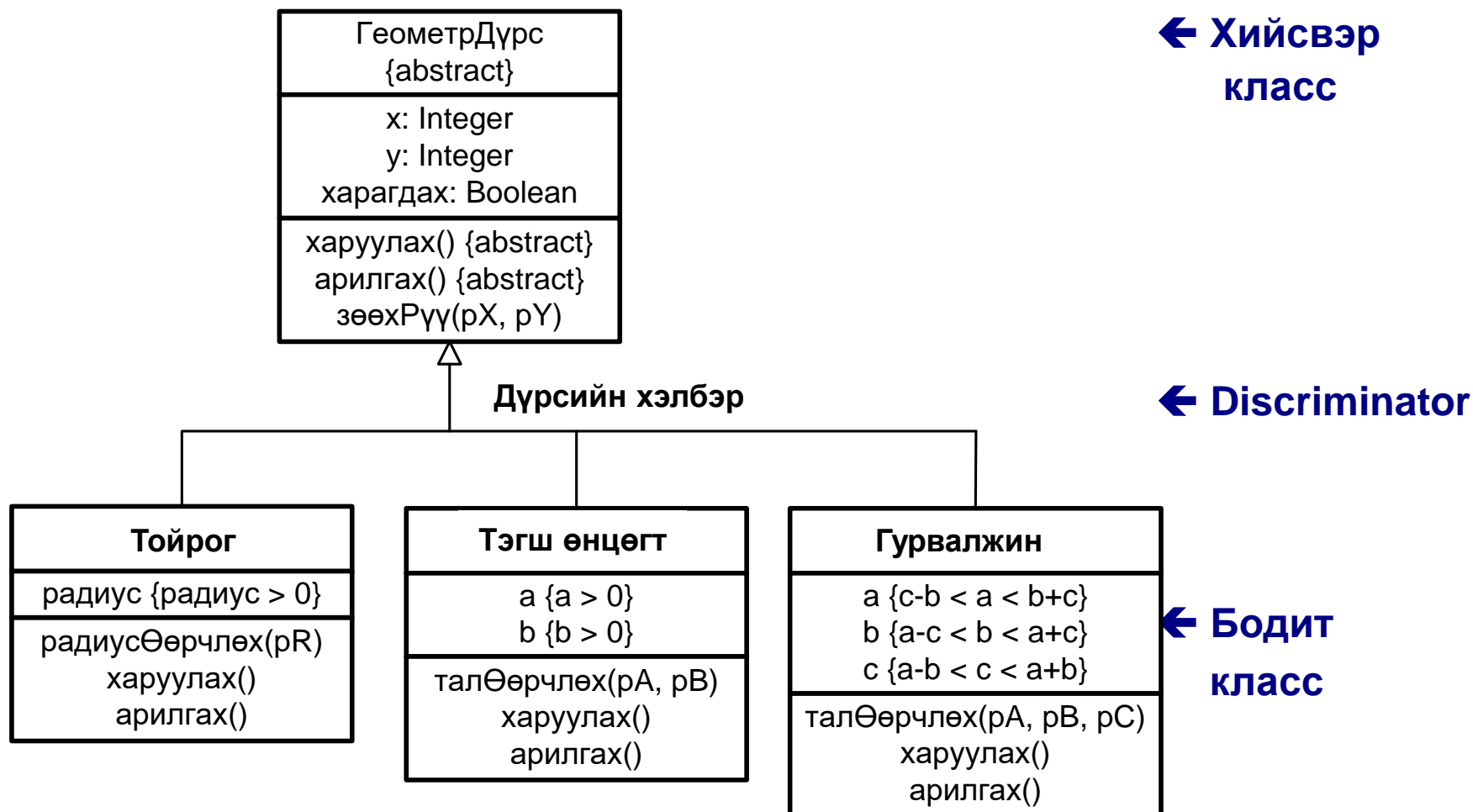
## Удамшилын жишээ(1)

- Дэлгэцэнд тойрог, тэгш өнцөгт болон гурвалжинг харуулж, зөөж мөн устгахаар болов.
- Тойрог, тэгш өнцөгт болон гурвалжин гэх ухагдхууныг ерөнхийлж мөн нийтэд нь геометрийн дүрс гэж нэрлэв.
- Тойрог, тэгш өнцөгт болон гурвалжин нь энэ дагуу GeomFigur гэх нэг эх классын нарийвчлалт юм.



## Удамшилын жишээ(2)

- Классын диаграм





## Удамшилын жишээ(3)

– Java-Програмын код (хэсэгчилсэн)

```
abstract class GeomFigur
{
    int x, y;
    boolean isVisible;

    public abstract void show();
    public abstract void delete();
    public void moveTo(int pX,pY)
    {
        if (isVisible)
        {
            delete();
            this.x = pX;
            this.y = pY;
            show();
        } else
        {
            this.x = pX;
            this.y = pY;
        }
    }
}
```

```
class Rectangle extends GeomFigur
{
    int a, b;

    public void show()
    {...}

    public void delete()
    {...}

    public void setEdge(int pA, pB)
    {
        if ((a > 0) && (b > 0))
        {
            this.a = pA;
            this.b = pB;
        }
    }
}
```

### Удамшилын давуу болон сул тал

#### + Өөрчилж болохуйц байдлыг дэмждэг

- Эх классын шинжийн өөрчлөлт нь автоматаар бүх дэд класст нөлөөлдөг

#### + Код хэмнэлт

#### + Дахин ашиглалтыг дэмждэг

#### — Нууцлалын зарчмыг зөрчдөг

- Дэд класс нь эх классын өөрчлөлтөөс хамааралтай
- Дэд классыг ойлгохын тулд эх классыг мөн авч үзэх хэрэгтэй

**Удамшил нь хүчирхэг, харин бас төвөгтэй ойлголт юм**



## 3.4-ийн асуулт

Объект нь шинж болон аргыг өвлөн авч болох уу?

### Хариулт

- ☐ Тийм
- ☒ Үгүй
- ☐ Тийм, классын шинж болон үйлдлээс бусдыг нь

### Үндэслэл

- Класс нь өөр классаас шинж болон аргыг өвлөн авч болно
- Объект нь өөрийн классын шинж болон аргыг үүсэж байх үедээ тэр л хэв загвараар авдаг



## **§ 3      Объект хандлагат шинжилгээний статик ойлголт**

3.1 Статик болон динамик ойлголтын харьцуулалт

3.2 Холбоос

3.3 Бүрдмэл болон Нийлмэл-Composition

3.4 Удамшил

### **3.5 Багц**

3.6 ЗНХ-ийг өргөтгөсөн механизм (Бие даалт)

3.7 Дүгнэлт



## Багц гэж юу вэ?

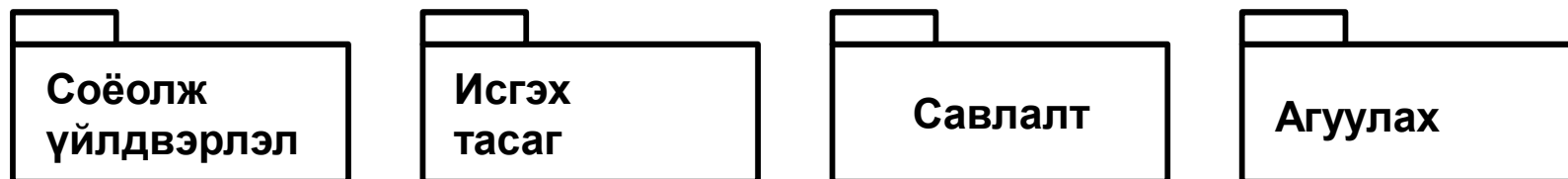
### Тодорхойлолт:

**Багц** (*package*) загварын элементыг (ж. нь класс) нэгтгэн хураангуйлдаг

- Том хэмжээний загварыг илүү тоймтой, ойлгомжтой болгодог
- Багц нь өөрөө багцыг агуулж болно
- Системийн бүтцийн өндөр хийсвэрлэлтийн түвшиний тайлбар

**Системийг бүхэлд нь нэг том багц болгож авч үзэж болно**

### Жишээ: Исэнгэ нэрэх үйлдвэр



## Багцын ЗНХ-ийн тэмдэглэгээ

- Багцын нэр нь нийт системийн хувьд тов тодорхой байх ёстой
- Багцийн диаграм: Багц болон тэдгээрийн хамаарлыг агуулдаг



**Багц 1 болон Багц 2 нь Багц 3-аас хамаарна**

- Класс бүр (Загварын элемент бүр) хамгийн ихдээ нэг багцад хамаарна
- Өөр олон багцад түүнийг зааж өгч болно
- Багц нь бүх агуулагдсан классын нэрийн орон зайг тодорхойлж өгдөг
- Багцын гаднаас заах
  - Багц::класс эсвэл багц1::багц11::багц111::класс

### **Багц дахь удамшил**

- Эх багцын нийтийн (public) болон хамгаалагдсан (protected) элементыг удамшсан багц нь өвлөдөг
- Элемент нь удамшсан багц дотор дарж бичигдэж болно

### **Багцад хуваасны давуу тал**

- + Дэд систем ө.х. системийг хэсэгчилэн авч үзсэнээр илүү ойлгомжтой
- + Нэрийн зөрчилээс зайлсхийх
- + Багц дахь элементын хандалтын хяналт, Битүүмжлэл
- + Багцийг тусгаарлан шалгаснаар тестийн үе шатыг хялбарчилсан



## 3.5-ийн асуулт

Багцийг үүсгэх ямар дүрэм нь зөв вэ?

### Хариулт

- ☒ Харьцаа ихтэй классыг нэгтгэх
- f** ☐ Систем бүрт дор хаяж гурван багц байлгах ёстой
- f** ☐ Багцын нэрийг агуулагдаж буй классын нэрээс нь үүсгэх
- ☒ Хоорондоо хамааралтай ажлын явцаас олж таних
- ☒ Бүрдмэл, нийлмэл болон удамшилын бүтцийг нэг ижил багцад нэгтгэх





## 3.5-ийн асуулт

Аялалын баримт бичиг, аялалын танилцуулга, гэрээ, жуулчин болон каталог гэсэн классыг аялал болон үйлчлүүлэгчийн мэдээлэл гэсэн багцад хуваа.

### Хариулт



## **§ 3      Объект хандлагат шинжилгээний статик ойлголт**

3.1 Статик болон динамик ойлголтын харьцуулалт

3.2 Холбоос

3.3 Бүрдмэл болон Нийлмэл-Composition

3.4 Удамшил

3.5 Багц

**3.6 ЗНХ-ийг өргөтгөсөн механизм (Бие даалт)**

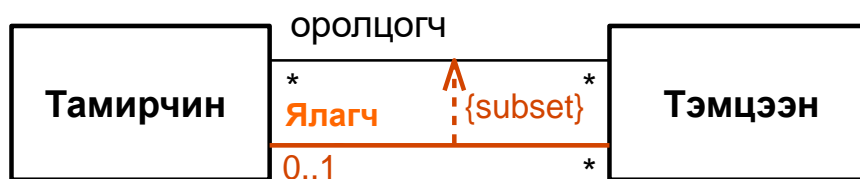
3.7 Дүгнэлт



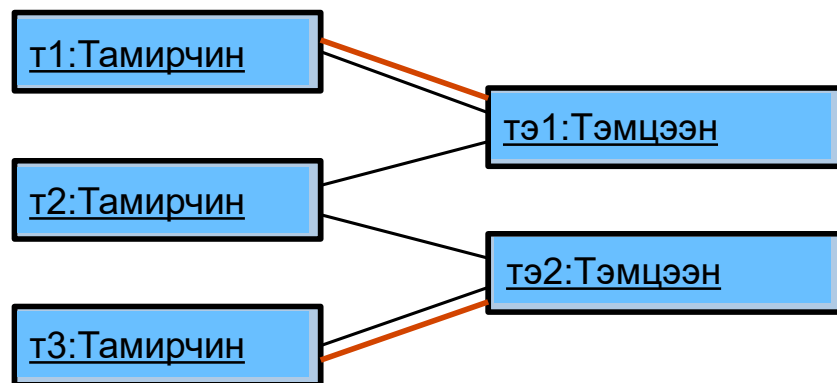
## Холбоос дахь хязгаарлалтын бусад төрлүүд

### – subset-Хязгаарлалт

- **subset**- Хязгаарлалт нь дэд олонлогийг үүсгэдэг
- Зөвхөн, хэрэв үндсэн холбоос оршин буй бол, оршино

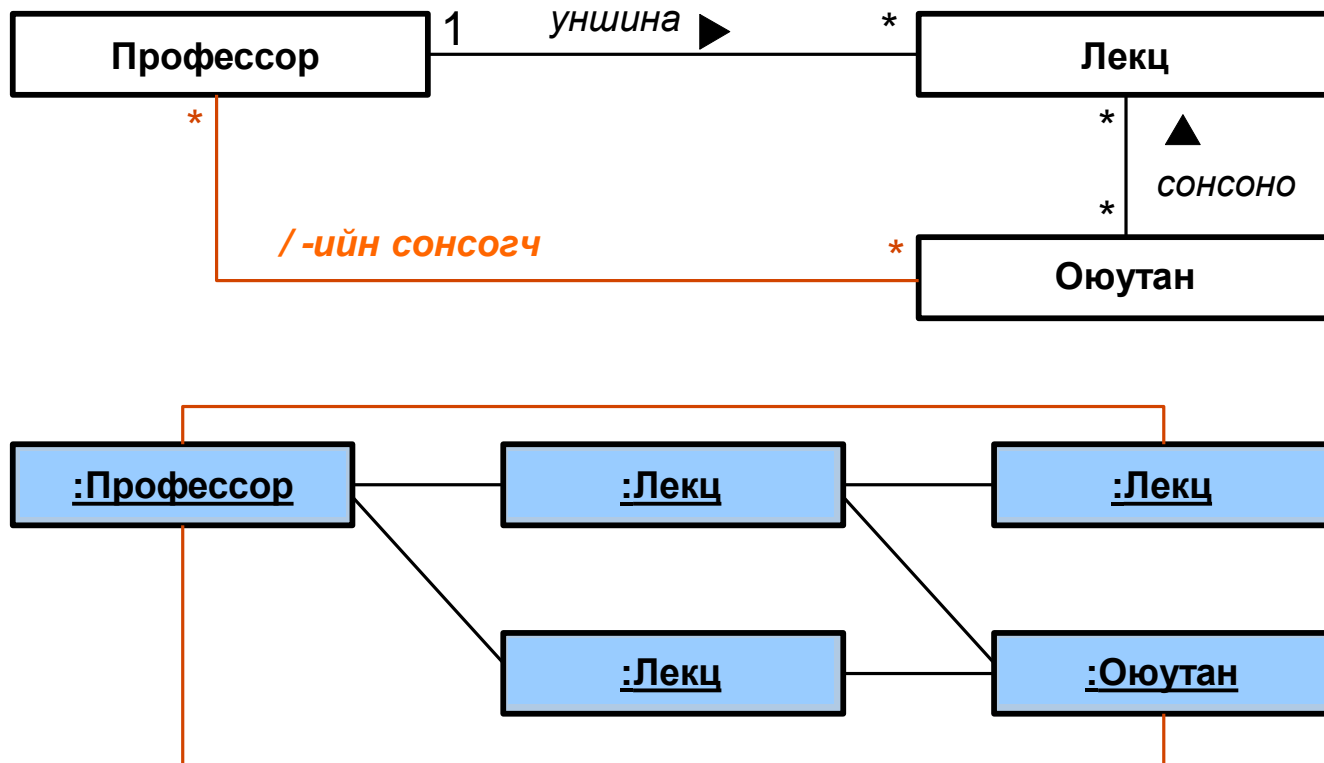


Тэмцээний ялагч нь оролцогч  
л байх ёстой



## Үүсмэл холбоос(derived association)

- Өөр бусад объект болон объектын харьцааны утгаас дурын үед үүсгэх объектын харьцаа ( $\Rightarrow$  redundant)
- Тэмдэглэгээ: өмнө нь „/“
- Хэрэв шаардлагатай бол үүсгэх дүрмийг хязгаарлалтаар тэмдэглэнэ

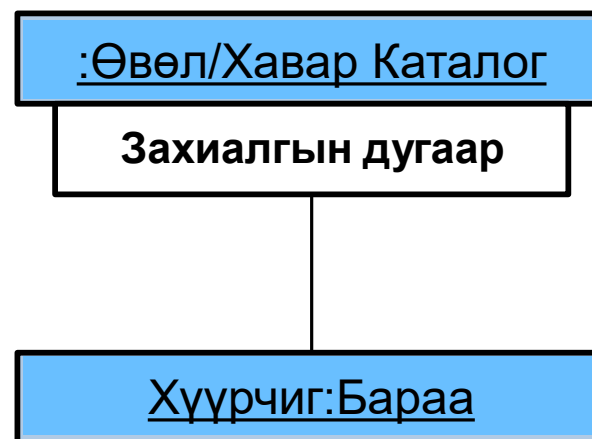


## Объект диаграмд холбоосыг хэрэглэх

- Объект диаграммын илэрхийлэх чадварыг өсгөх
  - Үүргийн нэр

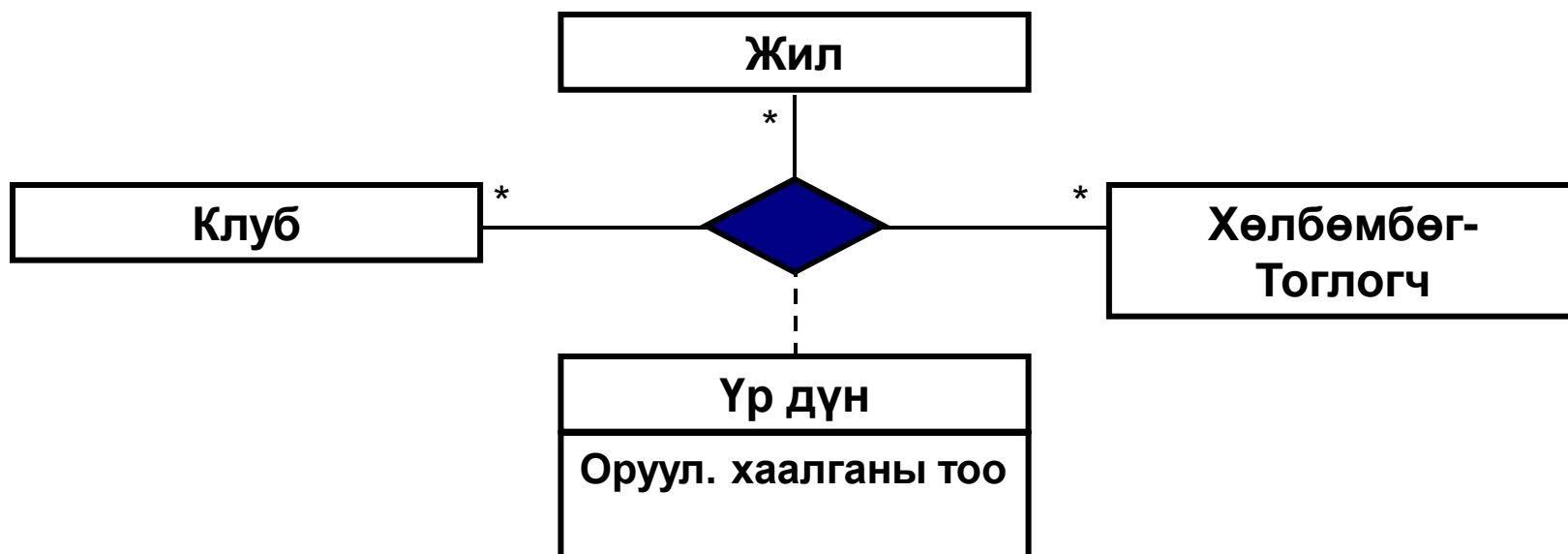
**Объектын холбоосын холбоосын нэрийг орхиж болно, доогуур нь зурах ёстой!**

- Бүлэглэлтийг өгөх
- Бүрдмэл ө.х. нийлмэлийн тэмдэгийг (Бүлэг 3.3-ийг үз)



## Өндөр эрэмбийн холбоос

- Зарчимын хувьд гурав болон олон классын хоорондын холбоос боломжтой
- Нэршил: n-тоот холбоос (n-ary association)
- Гуравтын болон дээших холбоос нь бүрдмэл эсвэл нийлмэлийг үүсгэх боломжгүй



---

## § 3      **Объект хандлагат шинжилгээний статик ойлголт**

3.1 Статик болон динамик ойлголтын харьцуулалт

3.2 Холбоос

3.3 Бүрдмэл болон Нийлмэл-Composition

3.4 Удамшил

3.5 Багц

3.6 ЗНХ-ийг өргөтгөсөн механизм (Бие даалт)

**3.7 Дүгнэлт**

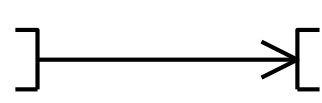


## § 3-ийн дүгнэлт

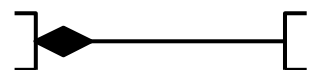
- **Холбоос** нь нэг эсвэл олон классын объектын хоорондох холбоо хэлхээг загварчилдаг
- Холбоосын онцгой тохиолдол нь **бүрдмэл** болон **нийлмэл**
- **Удамшил** нь ерөнхий класс болон нарийвчилсан класс хоорондын харьцааг тайлбарлан бичдэг
- **Багц** нь загварын элементийг бүлэглэдэг мөн ПХ-ийн системийг хийсвэрлэлтийн өндөр түвшинд дүрслэн харуулах боломжийг олгодог

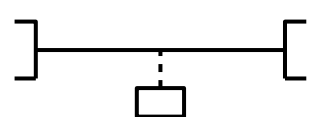
 Удамшил

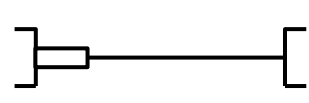
 Холбоос

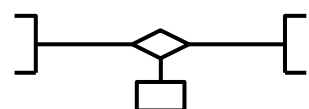
 Чиглэлтэй холбоос

 Бүрдмэл

 Нийлмэл

 Холбоос класс

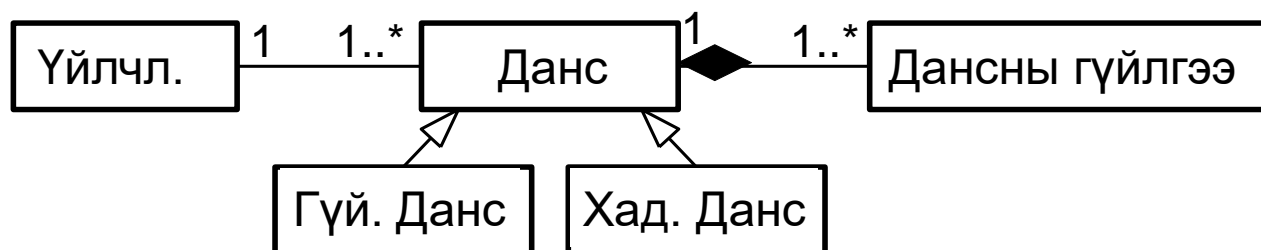
 Бүлэглэсэн холбоос

 Өндөр эрэмбэт холбоос



## § 3 –ийн асуулт

**Дансыг захиран зохицуулах** классын диаграм дахь класс хоорондын харьцааг тайлбарла!



- **Үйлчлүүлэгч** болон **Данс** хооронд энгийн холбоос байна, учир нь үйлчлүүлэгч нь дансны бүрэлдэхүүн биш, эсрэгээрээ ч гэсэн.
- **Данс** болон **дансны гүйлгээ** хооронд **Нийлмэл** байна, учир нь whole-part-харьцаа байна: Дансны динамик семантик нь бүх гүйлгээний хувьд үргэлж биелэнэ
- Данс нээх үед тэр дор нь мөнгө хийх учир, дансны гүйлгээний хувьд харьцааны тоон утга 1..\* гэж бичигдэнэ
- Данснаас дансны гүйлгээ дэх **нийлмэл** нь дэд класст **удамшина**, тодруулбал гүйлгээний болон хадгаламжийн данс-объект нь дансны гүйлгээг агуулна
- **Үйлчлүүлэгч** дэх холбоосын **удамшил**–ийн утга нь, үйлчлүүлэгч болон гүйлгээний эсвэл хадгаламжийн данс хооронд харьцаа оршино гэсэн үг.

## § 3-ийн шалгалтанд бэтгэх асуулт

### Асуулт 1: Багцын тухай аль өгүүлбэр зөв вэ?

#### Хариулт

- ☐ Нэг интерфейс нь олон багцад агуулагдсан байж болно.
- ☐ Хоёр багцад ижил нэртэй класс оршин байвал, багцын нэрийг онцгойлон бичиж өгөх ёстой.
- ☐ Багц нь том хэмжээний загварыг хуваахад бүтэцлэх хэрэгсэл болж хэрэглэгдэнэ.
- ☐ ЗНХ-д багцыг нэг нэгэнд нь оруулж хайрцаглаж болохгүй.
- ☐ Багц нь өөр хоорондоо үргэлж хамааралгүй байх ёстой.



## § 3-ийн шалгалтанд бэтгэх асуулт

**Асуулт 2: Холбоос, Бүрдмэл болон Нийлмэлтэй холбоотой өгүүлбэрээс аль нь зөв вэ?**

### Хариулт

- ☐ Холбоос нь оролцож буй классын объект хоорондын харьцааг тайлбарлан бичдэг.
- ☐ Бүрдмэл болон Нийлмэл нь холбоос юм.
- ☐ Нийлмэлд объект нь нэгэн зэрэг олон бүхэл объектын бүрэлдэхүүн хэсэг нь байж болно.
- ☐ Бүрдмэлд бүрэлдэхүүн класс нь бүхэл классаас өвлөдөг.
- ☐ Програмчлалын хэл Java нь холбоос, бүрдмэл болон нийлмэлийг хооронд нь ялгадаггүй.



## § 3-ийн шалгалтанд бэлтгэх асуулт

**Асуулт 3: Удамшилтай холбоотой дараах өгүүлбэрээс аль нь зөв вэ?**

### Хариулт

- ☐ Бодит болон хийсвэр классын аль альнаас нь объект үүсгэх боломжтой.
- ☐ Дэд класс нь өөрийн эх классын нарийчлалыг илэрхийлдэг.
- ☐ Нэг ерөнхий классаас олон тооны дэд класс өвлөж байвал, үүнийг нийлмэл удамшил гэдэг.
- ☐ Эх класс нь өөрийн дэд класстайгаа бүрэн нэг хэв шинжтэй, харин өөр нэмэгдэл мэдээллийг дээр нь агуулдаг.



## § 3-ийн шалгалтанд бэлтгэх асуулт

**Асуулт 4: ОХШ-тэй холбоотой дараах өгүүлбэрээс аль нь зөв вэ?**

### Хариулт

- ☐ Шинжилгээний үед систем юу хийхийг мөн хэрхэн түүнийг хэрэгжүүлэхийг тогтоодог.
- ☐ Шинжилгээ нь хэрэгжүүлэлттэй нягт холбогдсон.
- ☐ Хийх гэж буй системийн бизнес ойлголтыг шинжилгээний үед гаргадаг.
- ☐ ОХШ-ний загвар нь зөвхөн статик загвараас тогтдог.
- ☐ Шинжилгээ нь хэрэглэгчийн өнцөгөөс харан асуудлын талбарыг тайлбарлан бичдэг.



## § 4 Объект хандлагат шинжилгээний динамик ойлголт

### Сурах зорилт

- Ажлын явц гэж юуг тайлбарлаж чадах
- Зурвас явц гэж юуг тайлбарлаж чадах
- Сценари гэж юуг тайлбарлаж чадах
- Төлөвийн автомат гэж юуг болон динамик загварт ямар үүрэгтэйг тайлбарлаж чадах
- Идэвхжилтийн диаграм гэж юуг тайлбарлаж чадах
- Классын диаграм болон динамик загварын диаграмууд хэрхэн нэгдэн ажиллахыг тайлбарлаж чадах
- Ажлын явцыг загварчилж чадах
- Дарааллын болон Харилцааны диаграмыг үүсгэж чадах
- Төлөвийн диаграмыг үүсгэж чадах



## **§ 4      Объект хандлагат шинжилгээний динамик ойлголт**

### **4.1 Ажлын явц**

4.2 Зурвас

4.3 Сценари

4.4 Төлөвийн автомат

4.5 Үйл идэвхжилт

4.6 Объект хандлагат ойлголтын хураангуй (бие даалт)

4.7 Дүгнэлт

