

ДААЛГАВАР 4

- ✓ Удамшил
- ✓ Дахин тодорхойлогдсон функц
- ✓ Дарж тодорхойлсон функц
- ✓ super, this түлхүүр үг

1. Дараах классуудыг тодорхойлж, класс тус бүр дээр объект үүсгэнэ үү. Бүх гишүүн өгөгдлүүд **private** хандалттай 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 харьяалагддаг 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.
        Жич: Person төрөлтэй массивт Sportsman, TennisPlayer болон Teacher классууд дээр
        үүсгэсэн объектыг олгох боломжтой. Өөрөөр хэлбэл удамшлын үед хүү класс дээр
        үүссэн объектын хувьд нэгээс олон төрөлтэй байдаг ба уг объектыг үүсгэхэд
```

хэрэглэгдсэн класс төрөлтэй байхаас гадна түүний эх классуудын төрөлтэй нийцтэй байдаг.

// Бүх объектын хэвлэх функцийг дуудаж, объект тус бүрийн бүх мэдээллийг хэвлэ.

}

}

2. Дараах классуудыг тодорхойлж, класс тус бүр дээр объект үүсгэнэ үү. **Бүх гишүүн өгөгдлүүд (тодорхой заасан гишүүдээс бусад нь) private хандалттай байна . Гишүүн функцүүд бүгд багцын хандалттай байна.**

class Person

{ *Гишүүн өгөгдлүүд (Инстэнс хувьсагчид):*

Нэр

Нас

Гишүүн функцүүд:

Байгуулагч

Бүх гишүүдийг хэвлэх display() функц

Хүний насыг өгөгдсөн утгаар өөрчлөх changeField() функц

Насыг буцаах функц

}

class Sportsman extends Person

{ *Гишүүн өгөгдлүүд (Инстэнс хувьсагчид):*

Тамирчны хичээллэдэг спортын нэр (Хувьсагчийн нэр **nameS** байх ба **багцын** хандалттай байна)

Биеийн жин (double төрлийн хувьсагч байна.)

Биеийн өндөр

Гишүүн функцүүд:

Байгуулагч

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

Биеийн жинг өгөгдсөн утгаар өөрчлөх changeField() функц. (Дахин тодорхойлогдсон функц)

Тамирчны зөвхөн биеийн жин болон насыг хэвлэх display() функц. (Дахин тодорхойлогдсон функц)

}

class TennisPlayer extends Sportsman

{ *Гишүүн өгөгдлүүд (Инстэнс хувьсагчид):*

Тамирчны харьяалагддаг клубын нэр (Хувьсагчийн нэр **nameS** байна.)

Тамирчны спортын зэрэг

Гишүүн функцүүд:

Байгуулагч

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

Тамирчны хичээллэдэг спортын нэр болон харьяалагддаг клубын нэрийг өгөгдсөн утгуудаар өөрчлөх changeField() функц. (Дахин тодорхойлогдсон функц)

Тамирчны зөвхөн хичээллэдэг спортын нэр, харьяалагддаг клубын нэр, мөн насыг хэвлэх display() функц (Дахин тодорхойлогдсон функц)

}

class Teacher extends Person

{ *Гишүүн өгөгдлүүд (Инстэнс хувьсагчид):*

Багшийн албан тушаал

Багшийн цалин

Гишүүн функцүүд:

Байгуулагч (Нэр, нас, албан тушаалын утгуудыг хүлээн авч, олгох)

Байгуулагч (Нэр, нас гэсэн 2 утгыг л хүлээн авах ба энэхүү байгуулагч нь Teacher классын өөр байгуулагчийг дуудах ба гурав дахь утгыг “Ахлах багш” гэсэн утга дамжуулна.)

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

Багшийн цалинг тодорхойлох функц. (Албан тушаал нь туслах багш, багш, ахлах багш гэсэн 3 янз байх ба үүнээс шалтгаалан ялгаатай байдлаар цалинг олгоно)

```
}  
class Demo  
{  
  
    public static void main(String []args)  
    {  
        // Person класс дээр нэг объект буюу нэг хүн үүсгэж, мэдээллүүдийг нь хэвлэж хар.  
        // Sportsman класс дээр нэг тамирчин үүсгэн, бүх мэдээллүүдийг нь хэвлэж хар.  
        // Тамирчны биеийн жин болон насыг өөрчилж, зөвхөн эдгээр өөрчлөгдсөн утгуудыг  
        буюу биеийн жин, насыг дахин хэвлэн хар.  
        // TennisPlayer класс дээр нэг теннисний тамирчин үүсгэн, бүх мэдээллүүдийг хэвлэж  
        хар.  
        // Теннисний тамирчны биеийн жин, хичээллэдэг спортын нэр, харъяалагддаг  
        клубын нэр, насыг өөрчил.  
        // Теннисний тамирчны зөвхөн өөрчлөгдсөн мэдээллүүдийг хэвлэж, хар.  
        // Teacher класс дээр хоёр багш үүсгэх ба эхний объектыг үүсгэхдээ эхний  
        байгуулагчийг, хоёр дахь объектыг үүсгэхдээ хоёр дахь байгуулагчийг ашиглан  
        үүсгэнэ.  
        // Хоёр багшийн цалинг тооцоолж олно.  
        // Аль нэг багшийн насыг өөрчил.  
        // Багш бүрийн бүх мэдээллийг хэвлэж хар.  
    }  
}
```

ДААЛГАВАР 5

- ✓ “Has-a” харилцаа холбоо
- ✓ Хийсвэр класс
- ✓ Интерфейс

1. Дараах интерфейс болон классуудыг тодорхойлно уу. Гишүүн өгөгдлүүд бүгд private хандалттай байна.

interface A

```
{  
    //Данснаас мөнгө хасах функц  
    void methodA(int a);  
}
```

interface B extends A

```
{  
    //Дансанд мөнгө нэмэх функц  
    void methodB(int b);  
}
```

interface C

```
{  
    //Хэвлэх функц  
    void display();  
}
```

abstract class Account implements B, C

```
{  
    Гишүүн өгөгдлүүд (Инстэнс хувьсагчид):  
        Дансны дугаар  
        Дансны үлдэгдэл  
        Хүүгийн хэмжээ  
    Гишүүн функцүүд:  
        Байгуулагч (Дансны дугаар, үлдэгдлийг олгох )  
        Дансны үлдэгдлийг буцаах функц  
        Данс дахь үлдэгдлээс сар бүрийн хүүг тооцож, дансны үлдэгдлийг нэмэгдүүлэх  
        хийсвэр функц (Уг функцийг дарж тодорхойлох классууд хэрэгжүүлэлтийг ялгаатай  
        байдлаар хийнэ. )  
        Хүүгийн утгыг буцаах функц  
        .....  
}
```

class CheckingAccount extends Account implements C

```
{  
    Шалгах дансанд таны өөрийн мөнгө байрлах ба бүх төрлийн гүйлгээ хийх, дэлгүүрээс бараа  
    авах зэрэгт хэрэглэж болно.  
    Гишүүн өгөгдлүүд (Инстэнс хувьсагчид):  
        Хүүгийн хэмжээ  
  
    Гишүүн функцүүд:  
        Байгуулагч (Дансны дугаар, үлдэгдэл, хүүгийн хэмжээг олгох)  
        Хүү тооцох функц (Данс дахь үлдэгдэл 1000000 төгрөгнөөс бага бол Account класст  
        тодорхойлогдсон хүүгийн утгаар, эсрэг тохиолдолд CheckingAccount класст  
        тодорхойлогдсон хүүгийн утгаар тухайн сарын хүүг үлдэгдэл мөнгөний хэмжээнээс  
        тооцож бодно. Бодсон хүүгийн утгаар дансны үлдэгдлийг нэмэгдүүл. )  
        .....  
}
```

class CreditAccount extends Account

```
{  
    Кредит дансыг ашиглан дэлгүүрээс бараа худалдан авч, төлбөр төлөх үед ашиглагдах данс.  
    Гэхдээ энэ нь өөрийн мөнгө биш, банкны мөнгө ба ингэснээр та өртэй болох буюу зээл авч буй  
    хэрэг юм. Худалдан авалт хийх бүрт энэ дансны үлдэгдэл таны гаргасан зардлын хэмжээгээр  
    нэмэгдэх ба энэ нь таны хувьд зээл юм. Та өөрийн дансанд байрлах утгын хэмжээгээр банкинд  
    мөнгө шилжүүлснээр таны энэ дансны үлдэгдэл тэглэгдэх буюу зээл эргэн төлөгдөнө.  
  
    Гишүүн функцүүд:  
    Байгуулагч (Дансны дугаар, үлдэгдэл, хүүгийн хэмжээг олгох)  
    Хүү тооцох функц (Данс дахь үлдэгдэл ямар байхаас үл хамааран Account класст  
    тодорхойлогдсон хүүгийн утгаар тухайн сарын хүүг үлдэгдэл мөнгөний хэмжээнээс  
    тооцож бодно. Бодсон хүүгийн утгаар дансны үлдэгдлийг нэмэгдүүл.)  
}
```

class Employee implements C

```
{  
    Гишүүн өгөгдлүүд (Инстэнс хувьсагчид):  
    Ажилтны нэр  
    Ажилтны орлого  
    CheckingAccount account1;  
    CreditAccount account2;  
  
    Гишүүн функцүүд:  
    Байгуулагч (Ажилтны өгөгдүүдийн анхны утгыг олгох)  
    Орлого нэмэгдүүлэх функц  
    Орлогын тодорхой хэсгийг account1 данс руу шилжүүлэх функц (уг функцэд хэдэн  
    хувийг шилжүүлэхийг илэрхийлэх тоо ирнэ. Жишээ нь 80, энэ тохиолдолд та  
    орлогын 80%-г бодож, данс руу шилжүүлнэ.)  
    Худалдаа хийх функц (Уг функцэд худалдан авалтын зардлын утга ирнэ. Хэрэглэгч  
    худалдан авалт хийх үед account1 дансыг ашиглах ба данс дахь үлдэгдэл  
    хүрэлцэхгүй үед account2 дансыг ашиглан худалдан авалт хийнэ.)  
    Зээл төлөх функц буюу account2 дансны утгыг тэглэх функц  
    Данс тус бүрийн хүүг тооцох функц  
    Дарж тодорхойлох функц  
}  
  
class DemoEmployee  
{  
    public static void main(String []args)  
    {  
        // Нэг ажилтан үүсгэ. Ажилтны мэдээллийг гараас ав.  
        // Орлогын өгөгдсөн хувийг ажилтны account1 данс руу шилжүүл.  
        // Ажилтны account1 дансны үлдэгдлээс хүү бодож, дансны үлдэгдлийг шинэчил.  
        // Ажилтны данс бүрийн үлдэгдлийг дэлгэцэнд хэвлэ.  
        // Тухайн ажилтны хувьд хэд хэдэн удаа худалдан авалт хийнэ үү. Хоёр данс хоёулаа  
        ашиглагдах ёстой.  
        // Ажилтны данс бүрийн үлдэгдлийг дэлгэцэнд хэвлэ.  
        // Ажилтан дараа сарын цалин буутал account2 данс дахь зээлээ эргэн төлөөгүй ба  
        энэ хугацаанд хүү бодогдож, таны эргэн төлөх мөнгө нэмэгдэх ёстой.  
        // Ажилтны сарын цалин буух ба орлогыг нэмэгдүүл.  
        // Ажилтан зээлээ төлж, account2 дансны үлдэгдлийг тэглэнэ.  
        // Ажилтны данс бүрийн үлдэгдлийг дэлгэцэнд хэвлэ.  
    }  
}
```