

# Удамшлын харьцаа болон бүрдэл харьцаа (Лаборатори №9)

Ү.Мэндсайхан

МУИС, ХШУИС, Компьютерын ухаан, mendsaikhan@gurvanshiidel.mn

## 1. ОРШИЛ

Удамшлын үүсгэх харьцаа болон бүрдэл харьцааны талаар дэлэнгрэнгүй судлаж объект хандлагат програмчлалд хэрхэн хэрэгжүүлж болох талаар судалсан. Мөн бүрдэл харьцаан дотроо хориглолт буюу constraint гэж юу болох талаар судалан хэрэгжүүлсэн.

## 2. ЗОРИЛГО

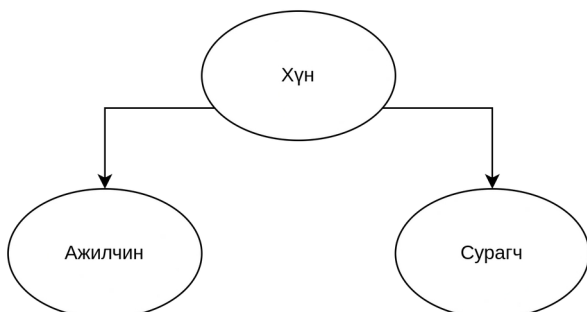
Хүн гэх классаас шаталсан хэлбэрээр удамшуулан Эхнэр, Ажилчин, Хүүхэд гэх классуудыг шинээр тодорхойлсон ба Эхнэр, Хүүхэд, АжлынТухай, Салбар зэрэг классуудыг Ажилчин классын нэгэн хэсэг байхаар классуудыг тодорхойлох. Үүний тулд дараах дараах ойлголтуудыг судлах.

- 2.1 Удамшил ямар харьцаа үүсгэдэг вэ?
- 2.2 Бүрдэл харьцаа гэж юу вэ?
- 2.3 Aggregation харьцааны тухай ойлголт
- 2.4 Бүрдэл харьцааны хориглолт (constrain) гэж юу вэ?

## 3. ОНОЛЫН СУДАЛГАА

### 3.1 Удамшил ямар харьцаа үүсгэдэг вэ?

Удамшил нь “тэр бол(is a)” болон “тийм төрлийнх(is a kind of)” гэх харьцаа үүсдэг бөгөөд ямарваа нэгэн эх класс нь хүүхэд классынхаа ерөнхий шинж чанар нь болж өгдөг. Жишээ нь доорх диаграмаас бид ажилчин бол хүн, сурагч бол хүн гэх удамшлын хамаарлыг хэлж чадна.



### 3.2 Бүрдэл харьцаа гэж юу вэ?

Бүрдэл харьцаа нь ямарваа нэгэн класс тодорхойлохын тулд тухайн класс өөр нэгэн класс бүхий объектыг гишүүн өгөгдөл хэлбэрээр агуулах байдлыг хэлнэ. Бүрдэл харьцаа нь “түүнийг агуулсан” буюу “has a” гэх хамаарлыг төлөөлдөг. Жишээ нь машин хөдөлгүүртэй болон дугуйнуудтай гэж үзье. Тэгвэл машин гэдэг класс нь дугуй класс болон хөдөлгүүр классын объектуудыг агуулна гэж ойлгож болно.

### 3.3 Aggregation харьцааны тухай ойлголт

Aggregation буюу Бүрдмэл холбоо нь олон объект нийлж нэг объект үүсгэхийг хэлнэ. Уг харьцаа нь нарийн бүтэц бүхий класс бүрдүүлэхэд хэрэглэгддэг. Класс хоорондын харьцаа, объект хоорондын харьцаа нь классыг ондоо классаас удамшуулж үүсгэх эсэх, объектон өгөгдөл байх эсэх зэрэг асуудлыг шийдэхэд хэрэглэнэ.

### 3.4 Бүрдэл харьцааны хориглолт (constrain) гэж юу вэ?

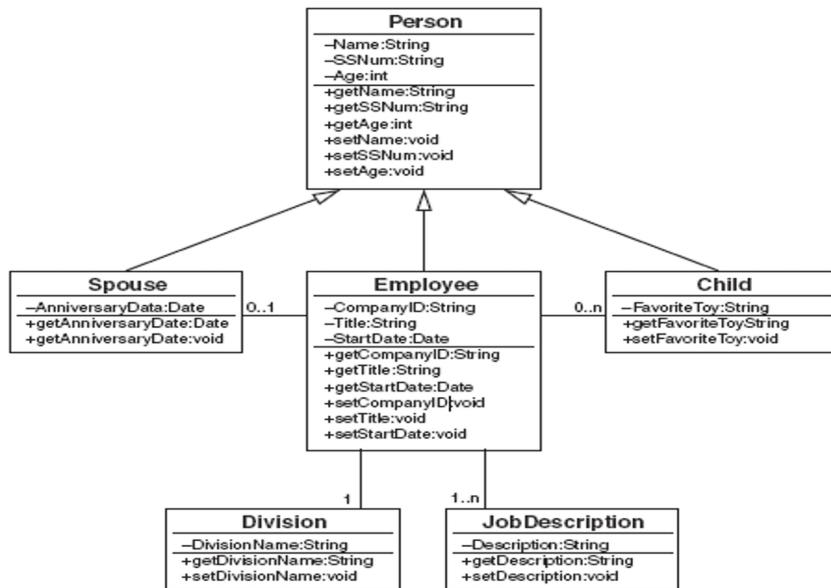
Бүрдэл харьцаан хориглолт гэдэг нь классууд өөр бусад классын объектыг агуулахад хориглолт хийж байгааг хэлнэ. Жишээ нь:

- 1)  $0 \dots 1$  (Хамгийн багадаа 0 буюу байхгүй, хамгийн ихдээ 1 ширхэгийг агуулна)
- 2)  $0 \dots n$  (Хамгийн багадаа 0 буюу байхгүй, хамгийн ихдээ хэд ч байж болно)
- 3)  $1 \dots 1$  (Хамгийн багадаа 1, хамгийн их дээ 1 буюу ямар ч үед 1ийг агуулна)
- 4)  $1 \dots n$  (Хамгийн багадаа 1, хамгийн ихдээ хэд ч байж болно)

Дээрх бүрдэл харьцааны хориглолт нь ямар нэгэн классд хамгийн багадаа болон хамгийн ихдээ хэдэн өөр классын объект агуулагдаж болохыг заана гэсэн үг.

## 4. ХЭРЭГЖҮҮЛЭЛТ

Дараах зурагт харуулсан класс диаграммын дагуу классуудад байгуул



Cardinality in a UML diagram.

### 1. Person class

```
class Person {
private:
    char *name;
    char *ssnum;
    int age;

public:
    char *getName() {
        return name;
    }
    char *getSSNum() {
        return ssnum;
    }
    int getAge() {
        return age;
    }
    void setName(char *ner) {
        delete name;
        name = new char(strlen(ner) + 1);
        strcpy(name, ner);
    }
    void setSSNum(char *registerDugaar) {
        delete ssnum;
        ssnum = new char(strlen(registerDugaar) + 1);
        strcpy(ssnum, registerDugaar);
    }
    void setAge(int a) {
        age = a;
    }
};
```

## 2. Spouse class

```
class Spouse: public Person{
private:
    int AnniversaryDate;
public:
    int getAnniversaryDate(){
        return AnniversaryDate;
    }
    void setAnniversaryDate(int a){
        AnniversaryDate = a;
    }
};
```

## 3. Child class

```
class Child : public Person {
protected:
    char *favoriteToy;

public:
    char getFavoriteToy() {
        return *favoriteToy;
    }
    void setFavoriteToy(char *togloom) {
        delete[] favoriteToy;
        favoriteToy = new char(strlen(togloom) + 1);
        strcpy(favoriteToy, togloom);
    }
    void showChildData() {
        showPersonData();
    }
};
```

#### 4. Employee class

```
class Employee : public Person {
protected:
    char *companyID;
    char *title;
    int startDate;
    Spouse ehner;
    vector<JobDescription> ajil;
    Division bairlal;
    vector<Child> child;

public:
    Employee() : Person() {
        companyID = new char(8);
        strcpy(companyID, "cmpnyId");
        title = new char(6);
        strcpy(title, "title");
        startDate = 0;
    }

    char getCompanyID() {
        return *companyID;
    }
    char getTitle() {
        return *title;
    }
    int getStartDate() {
        return startDate;
    }
    void setCompanyID(char *id) {
        delete[] companyID;
        companyID = new char(strlen(id) + 1);
        strcpy(companyID, id);
    }
    void setTitle(char *garchig) {
        delete[] title;
        title = new char(strlen(garchig) + 1);
        strcpy(title, garchig);
    }
    void setStartDate(int a) {
        startDate = a;
    }

    ~Employee() {
        delete[] companyID;
        delete[] title;
    }
};
```

#### 5. JobDescription & Division класс

```
class JobDescription {
private:
    char *description;

public:
    char *getDescription() {
        return description;
    }
    void setDescription(char *desc) {
        delete description;
        description = new char(strlen(desc) + 1);
        strcpy(description, desc);
    }
};
```

```
class Division {
private:
    char *divisionName;

public:
    char *getDivisionName() {
        return divisionName;
    }
    void setDivisionName(char *divisionNer) {
        delete[] divisionName;
        divisionName = new char(strlen(divisionNer) + 1);
        strcpy(divisionName, divisionNer);
    }
};
```

Main функц дотроо класс болгон дээр хэд хэдэн объект үүсгэж Employee классын объектуудад Child, Spouse, JobDescription, Division классуудын объектуудыг тохируулж өгсөн. Эцэст нь Employee классын объектуудын бүх мэдээллийг хэвлэж харуулсан.

```
mendeeislife@money-maker: ~/Classes/object/lab9
mendeeislife@money-maker:~/Classes/object/lab9$ ./a.out
*****Huviin-medeelel*****
Name: Bold
Register: ri03252211
Nas: 18
-----Ajil-alba-----
Company ID: Com1
Title: MCS
Start date: 20010501
Division 1st
Job: manager
Job: supervisor
-----Gergii-----
Name: Dulmaa
Register: ri01210112
Age: 21
Gerlesen jil: 20010501
-----Huuhed-----
Name: Bat
Register: ri12312312
Age: 10
Name: Dorj
Register: af12312312
Age: 9
*****Huviin-medeelel*****
Name: Bayr
Register: ri01210114
Nas: 20
-----Ajil-alba-----
Company ID: Com2
Title: TavanBogd
Start date: 20010101
Division 2nd
Job: ceo
Job: supervisor
-----Gergii-----
Name: Zaya
Register: fb99210112
Age: 21
Gerlesen jil: 20010501
-----Huuhed-----
Name: Tsetseg
Register: gg11112222
Age: 5
*****Huviin-medeelel*****
Name: Tselmeg
Register: ri00251114
Nas: 21
-----Ajil-alba-----
Company ID: Com2
Title: TavanBogd
Start date: 20020511
Division 3rd
Job: accountant
-----Gergii-----
```

## 5. ДҮГНЭЛТ

Энэхүү лабораторын ажлаар бүрдэл харьцаа гэж юу болох түүнийг классуудад хэрхэн хэрэгжүүлж ашиглах талаар мэдсэн ба давуу тал нь бүрдэл холбоог ашигласнаар классууд хоорондоо илүү уян хатан уялдаатай ажиллаж байна. Мөн динамик хүснэгт буюу vector-ийн тусламжтайгаар олон объектыг нэгэн мөрөнд хадгалж болох бөгөөд аливаа хүснэгтийн багтаамж дүүрэх зэрэг асуудлуудаас зайлсхийх боломжтой юм.

## 6. АШИГЛАСАН МАТЕРИАЛ

1. Объект хандлагат технологийн C++ програмчлал, Ж.Пүрэв, 2008, Улаанбаатар

## 7. ХАВСРАЛТ

Github линк, [Lab 9](#)