

## Лаборатори 8:

### 1. Удамшил ямар харьцаа үүсгэдэг вэ?

- Тэр бол тэр ( is a )
- Тийм төрлийнх (is a kind of) гэсэн 2 харьцаа үүсдэг.

Тэр бол тэр харьцаа нь хүүхэд бол хүн байна гэсэн утгатай юм. Харин тийм төрлийнх гэдэг харьцаа нь хүүхэд класс нь эх классын төрөл байна гэсэн харьцаа буюу жишээлбэл сандал бол тавилга төрлийн гэсэн үг.

### 2. Бүрдэл харьцаа гэж юу вэ? Объект хандлагат программчлалд хэрхэн хэрэгжүүлдэг вэ?

Нарийн бүтэц бүхий классыг бүрдүүлэхэд бүрдэл харьцааг ашигладаг бөгөөд энэ нь тийм юмтай **“has-a”** байх гэдэг харьцааг үүсгэдэг.

**Жишээ нь:** Машин гэсэн класс байвал энэ нь мотортой болон дугуйтай байх бөгөөд нэг классын гишүүн өгөгдөл нь өөр классын объект байж болно.

```
class Engine {
private:
    char *name;
    int v;
};
class Car {
private:
    char *name;
    Engine e;
};
```

### 3. Бүрдэл харьцааны хориглолт (constrain) гэж юу вэ?

Классууд хоорондоо тодорхой зааг буюу хориглолттойгоор хоорондоо харьцааг үүсгэдэг. Үүнд:

- **1..1:** харьцаа нь нэг класс заавал нөгөө классын объектыг агуулна гэсэн үг.
- **0..1:** харьцаа нь агуулж буй классад байхгүй эсвэл зөвхөн ганц байна гэсэн үг.
- **0..n:** харьцаа нь агуулж буй классад нөгөө классын хэдэн ч объект хадгалагдаж болно гэсэн утгатай.

- **1..n:** харьцаа нь агуулж буй классад нөгөө классын нэгээс цөөнгүй объектыг хадгална гэсэн утгатай .

```
#include <iostream>
#include <vector>
using namespace std;

struct Date
{
    int year;
    int month;
    int day;
};

class Person
{
private:
    string name;
    string ssNum;
    int age;

public:
    Person()
    {
        name = "ner";
        ssNum = "123";
        age = 16;
    }
    Person(string n, string ss, int a)
    {
        name = n;
        ssNum = ss;
        age = a;
    }
    string getName()
    {
        return name;
    }
    string getSSNum()
    {
        return ssNum;
    }
    int getAge()
    {
        return age;
    }
    void setName(string n)
    {
        name = n;
    }
    void setSSNum(string ss)
```

```

    {
        ssNum = ss;
    }
    void setAge(int a)
    {
        age = a;
    }
};

class Spouse : public Person
{
private:
    Date anniversaryDate;

public:
    Spouse()
    {
        anniversaryDate.year = 2022;
        anniversaryDate.month = 11;
        anniversaryDate.day = 1;
    }
    Spouse(int y, int m, int d)
    {
        anniversaryDate.year = y;
        anniversaryDate.month = m;
        anniversaryDate.day = d;
    }
    Date getAnniversaryDate()
    {
        return anniversaryDate;
    }
    void setAnniversaryDate(Date d)
    {
        anniversaryDate = d;
    }
};

class Child : public Person
{
private:
    string favoriteToy;

public:
    Child()
    {
        favoriteToy = "";
    }
    Child(string t)
    {
        favoriteToy = t;
    }
    string getFavoriteToy()

```

```

    {
        return favoriteToy;
    }
    void setFavoriteToy(string toy)
    {
        favoriteToy = toy;
    }
};

class Division
{
private:
    string divisionName;

public:
    Division()
    {
        divisionName = "";
    }
    string getDivisionName()
    {
        return divisionName;
    }
    void setDivisionName(string division)
    {
        divisionName = division;
    }
};

class JobDescription
{
private:
    string jobDescription;

public:
    JobDescription()
    {
        jobDescription = "";
    }
    string getJobDescription()
    {
        return jobDescription;
    }
    void setJobDescription(string job)
    {
        jobDescription = job;
    }
};

class Employee : public Person
{
private:

```

```

    string companyId;
    string title;
    Date startDate;

public:
    vector<Spouse> spouse;
    vector<Child> childs;
    Division div;
    vector<JobDescription> jDescription;
    Employee()
    {
        companyId = "";
        title = "";
        startDate.year = 2022;
        startDate.month = 11;
        startDate.day = 1;
    }
    Employee(string id, string t, int y, int m, int d)
    {
        companyId = id;
        title = t;
        startDate.year = y;
        startDate.month = m;
        startDate.day = d;
    }
    string getCompanyId()
    {
        return companyId;
    }
    string getTitle()
    {
        return title;
    }
    Date getStartDate()
    {
        return startDate;
    }
    void setCompanyId(string id)
    {
        companyId = id;
    }
    void setTitle(string t)
    {
        title = t;
    }
    void setStartDate(Date d)
    {
        startDate = d;
    }
};

int main()

```

```

{
    Employee emp1("N1", "MCS", 2016, 10, 1), emp2("N2", "Unitel", 2020, 10,
1);
    emp1.setName("BoId");
    emp1.setSSNum("UY21332421");
    emp1.setAge(32);

    Division div1, div2;
    JobDescription j1, j2;
    Child child1("car"), child2("doll");
    Spouse s1(2010, 12, 23), s2(2012, 4, 10);
    s1.setName("HuLan");
    s1.setSSNum("UI2142343");
    s1.setAge(32);
    child1.setName("Bat");
    child1.setSSNum("UI2142343");
    child1.setAge(5);
    child2.setName("Anu");
    child2.setSSNum("UI2142343");
    child2.setAge(2);

    emp1.spouse.push_back(s1);
    emp1.chlds.push_back(child1);
    emp1.chlds.push_back(child2);

    emp1.spouse.push_back(s2);

    cout << "Division name: ";
    cin >> n;
    emp1.div.setDivisionName(n);
    emp2.div.setDivisionName(n);

    cout << "Job description: ";
    cin >> n;
    j1.setJobDescription(n);

```

```

emp1.jDescription.push_back(j1);
emp2.jDescription.push_back(j1);

cout << "Emp1 ner: " << emp1.getName() << endl;
cout << "Emp1 register dugaar: " << emp1.getSSNum() << endl;
cout << "Emp1 nas: " << emp1.getAge() << endl;

cout << "Emp1 spouse name: " << emp1.spouse[0].getName() << endl;
cout << "Emp1 spouse age: " << emp1.spouse[0].getAge() << endl;

cout << "Emp1 company id: " << emp1.getCompanyId() << endl;
cout << "Emp1 company title: " << emp1.getTitle() << endl;
a = emp1.getStartDate();
cout << "Emp1 job started date: "
    << "year: " << a.year << ", "
    << "month: " << a.month << ", "
    << "day: " << a.day << ", " << endl;

a = emp1.spouse[0].getAnniversaryDate();
cout << "Emp1 annivesary date: "
    << "year: " << a.year << ", "
    << "month: " << a.month << ", "
    << "day: " << a.day << ", " << endl;

for (int i = 0; i < emp1.childds.size(); i++)
{
    cout << "Emp1 child name: \n";
    cout << emp1.childds[i].getName() << endl;
    cout << "Emp1 child age: \n";
    cout << emp1.childds[i].getAge() << endl;
    cout << "Emp1 child favorite toy: \n";
    cout << emp1.childds[i].getFavoriteToy() << endl;
}

return 0;

```

```
}

```