

## Лабораторийн ажил 4: Классын бүрэлдэхүүн

### Лабораторийн ажлын зорилго:

Классын бүрэлдэхүүнд багтах байгуулагч функц, байгуулагч функцийн олон хэлбэржилтийг хэрэгжүүлэх, функц, процедур бичих

### 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	Байгуулагч функц бичих	Ашиглах (Use)	Хэрэглээ	1, 7
2	Байгуулагчийн олон хэлбэржилтийг хэрэгжүүлэх	Ашиглах (Use)	Хэрэглээ	1, 7
3	Гишүүн функц, процедур бичих	Ашиглах (Use)	Хэрэглээ	6
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:

#### Классын гишүүн үйлдлүүд

Класс дараах төрлийн гишүүн үйлдлүүд байна. Үүнд:

- Асуух (query) үйлдэл,
- Өөрчлөх (mutator) үйлдэл,
- Байгуулагч функц,
- Устгагч функц.

#### Байгуулагч функц:

Байгуулагч бол классаас шинэ объект үүсэх үед ажилладаг онцгой үйлдэл юм. Байгуулагч функцийг Жава програмчлалын хэлэнд классын нэрээр тодорхойлдог. Байгуулагч функцийн үндсэн зорилго бол класс доторх private, protected харагдалттай гишүүдэд анхны утга оноож, объектыг үүсгэх юм. Ингэхдээ классын далд гишүүн өгөгдлүүдэд утга оноохын тулд байгуулагч функцийн параметруудийг ашигладаг.

### Байгуулагчийн төрлүүд:

- ✓ Өгөгдмөл/default байгуулагч
  - Хэрэв классд ямар нэг байгуулагч функц тодорхойлоогүй бол Жава компайляр нь ажиллахдаа кодонд өгөгдмөл байгуулагчийг нэмэх ба энэ нь кодонд харагдахгүй боловч .class файлд нэмэгдсэн байдаг.
  - Ямар нэг байгуулагч тодорхойлсон л бол өгөгдмөл байгуулагч ашиглагдахгүй болно.
- ✓ Аргументгүй/no-args байгуулагч
  - Аргументгүй тул өгөгдмөл байгуулагчтай адил мэт боловч ялгаа нь байгуулагчийн body хэсэгт код бичиж болно.
- ✓ Параметртэй байгуулагч
  - Классын гишүүн өгөгдлүүдэд утга олгох, үйлдэл хийхэд ашиглагдана.

Нэг класс хэд хэдэн байгуулагчтай (үүнийг байгуулагч функцийн олон хэлбэржилт гэнэ) байж болох бөгөөд бүгд ижилхэн нэр, буцаах утгатай боловч параметрийн жагсаалтаараа бусдаасаа ялгагдах ёстой. Параметргүй байгуулагчийг *үндсэн өгөгдмөл байгуулагч* гэж үздэг. Дараах жишээнд Student класс нэг байгуулагч функцтэй байна.

```
public class Student {  
    private String name;  
  
    public Student (String newName) {  
        name = newName;  
    }  
}
```

Энэ классаас объект үүсгэх жишээг харуулав.

```
Student st1 = new Student("Bold");
```

Байгуулагч функцийг олон хэлбэржилттэй тодорхойлж болно. Дараах жишээнд Person класс параметргүй, нэг параметртэй, хоёр параметртэй гэсэн гурван байгуулагч функцтэй байна.

```
class Person  
{  
    private String firstName;  
    private String lastName;  
    Person()  
    {  
        firstName = "";  
        lastName = "";  
    }  
    Person(String lname)  
    {  
        firstName = "";  
        lastName = lname;  
    }  
    Person(String fname, String lname)  
    {  
        firstName = fname;  
        lastName = lname;  
    }  
}
```

### Устгагч функц:

Объектыг санах ойгоос устгаж нөөцүүдийг чөлөөлөхөд устгагч функцийг ашиглана. Жишээлбэл:

- Санах ойд ачаалласан объектын өгөгдлийг устгаж, санах ойг чөлөөлөх,
- Сүлжээний холболт, өгөгдлийн сангийн холболтыг таслах,
- Санах ойд байгаа объектын өгөгдлийг өгөгдлийн санд хадгалан, өгөгдлийн сангийн холболтыг таслах,
- Санах ойд байгаа объектын өгөгдлийг файлд хадгалан, файлыг хаах.

Нэг классд зөвхөн ганц устгагч функц байна. Объект хандлагат технологид объектыг хоёр замаар устгадаг. Үүнд:

- Ил - ө.х тусгай үйлдэл эсвэл оператор дуудаж (C++)
- Далд - объект програмд хэрэггүй болох үед устгагдах (Java,C#)

### Ил үед

- Ашиглаж байгаа объектыг устгаж болно.
- Объектын устгалт хийгээгүйгээс санах ойн дутагдал үүсч болно.

### Далд үед

- Garbage collector (хог цуглуулагч) нь хэрэггүй болсон объектуудыг автоматаар устгадаг.
- Хог цуглуулагч ажиллаж эхэлсэн үед програмын ажиллагааны хувьд нэмэлт ачаалал өгдөг.

Java хэлэнд объект устгалт нь хог цуглуулагчаар дамжин далд гүйцэтгэгдэнэ. System.gc(); гэж ил байдлаар хог цуглуулагчийг идэвхижүүлж болно. Хог цуглуулагч нь объектыг устгахын өмнө Finalize гэх объектын тусгай үйлдлийг дууддаг. Энэ үйлдлийг өөрийн классдаа тодорхойлж болно. Энэ үйлдэл нь аргумент болон буцаах утгагүй байдаг. Java хэлэнд автоматаар хийгддэг тул устгагч функц тодорхойлохгүй, хог цуглуулагчийг дуудахгүй програмчилж болно.

## **Нэг үйлдлийн олон хэлбэржилт – overloading**

Нэг классд ижилхэн нэртэй үйлдэл байж болно. Гэхдээ эдгээр нь параметрууд болон буцаах утгаараа ялгаатай байх ёстой. Үүнийг параметрийн *олон хэлбэржилт* гэдэг. Байгуулагч функц нь олон хэлбэржилттэй байж болно.

**Жишээ :** Хүний мэдээлэл бүртгэдэг PersonRegistrar класс байвал хүнийг зөвхөн нэрээр, эсвэл овог нэрээр, эсвэл овог нэр болон төрсөн он сар өдрөөр бүртгэх гэсэн гурван үйлдлийг олон хэлбэржилтээр илэрхийлж болно. Үүнд:

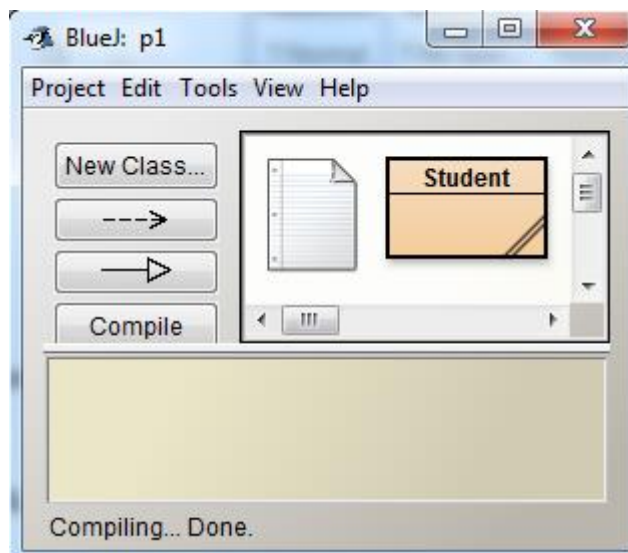
```
class PersonRegistrar {  
    public void registerPerson (String fname) {}  
    public void registerPerson (String lname, String fname) {}  
    public void registerPerson (String lname, String fname, Date bdate)  
    {}  
}
```

Дээрх жишээнд registerPerson үйлдлийг нэг тэмдэгт мөр параметртэй дуудвал нэг параметртэй эхний үйлдэл дуудагдана. Хэрэв хоёр тэмдэгт мөр параметртэй дуудвал хоёр параметртэй дундах үйлдэл дуудагдана. Харин хоёр тэмдэгт мөр, нэг Date төрөлтэй параметртэй дуудвал сүүлийн үйлдэл дуудагдана.

## BlueJ ашиглан классын объектоой ажиллах

Жишээлбэл, дараах Student классыг BlueJ хэрэгсэл дээр үүсгэсэн гэж үзье. Энэ класс нь гурван хэлбэрээр бичигдсэн байгуулагч функцийн олон хэлбэржилттэй, setName үйлдэл нь мөн олон хэлбэржилттэй, getName гэсэн асуух үйлдэлтэй.

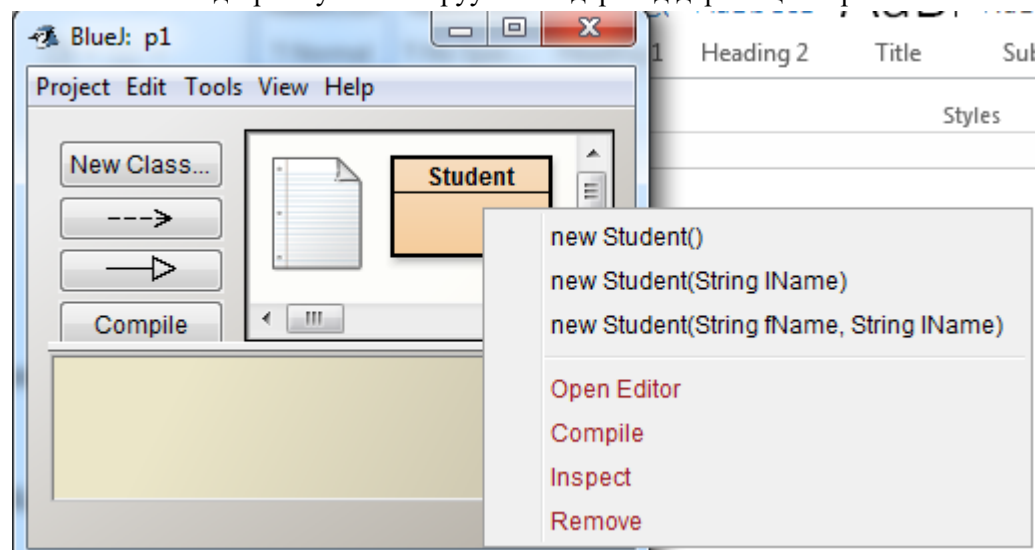
```
public class Student
{
    private String firstName;
    private String lastName;
    Student()
    {
        firstName = "";
        lastName = "";
    }
    Student(String lName)
    {
        firstName = "";
        lastName = lName;
    }
    Student(String fName, String lName)
    {
        firstName = fName;
        lastName = lName;
    }
    public void setName(String fName)
    {
        firstName = fName;
    }
    public void setName(String fName, String lName)
    {
        firstName = fName;
        lastName = lName;
    }
    public String getName()
    {
        return firstName + "-" +lastName;
    }
}
```



Зураг 4.1 Класс диаграм

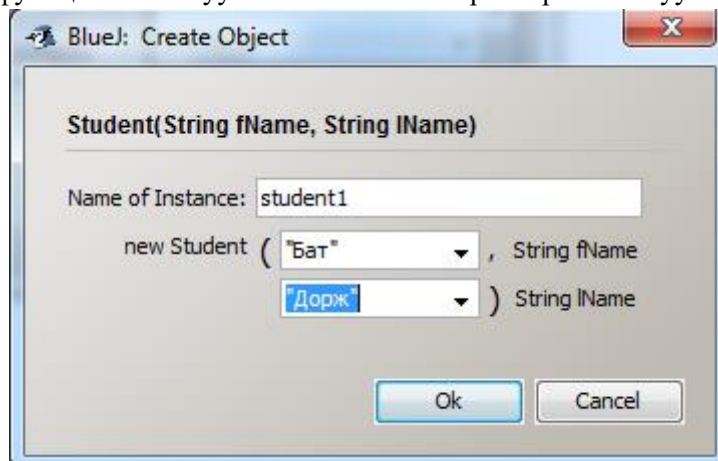
### Объект үүсгэх:

1. Student класс дээрээ хулганы баруун товч дарахад дараах цэс гарна.



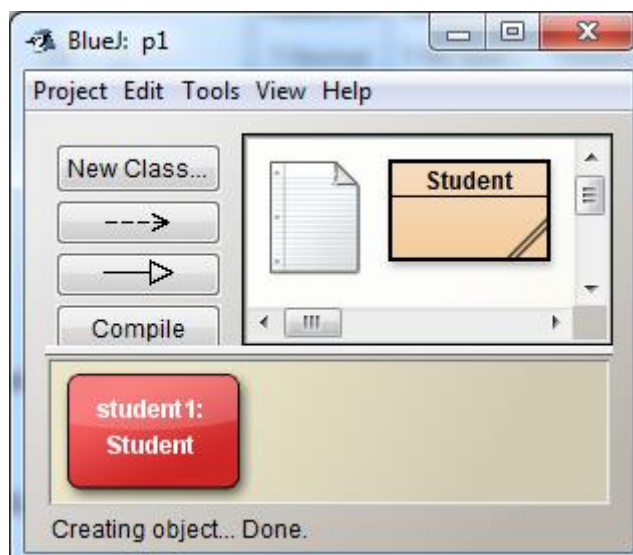
Зураг 4.2 Байгуулагч функц дуудах

2. Аль байгуулагч функцийг ажиллуулахаа сонгоно. 2 параметртэй байгуулагч функцийг сонгов.



Зураг 4.3 Параметрт утга олгох

3. Байгуулагч функцэд параметр дамжуулах. Ингэхдээ String төрөлтэй параметрт дамжуулахдаа “” хашилтанд хийнэ. Ok товч дарахад дараах байдлаар student1 нэртэй объект санах ойд үүссэн байна.



Зураг 4.4 Объект

### Функцийн тухай

Бодлогыг бодохын тулд жижиг асуудлуудад хувааж, жижиг асуудлуудаа дахин шийдвэрлэх боломжтой болтол нь жижиг асуудлуудад хуваадаг. Ингэж хуваахдаа хуваалтын хэсгүүдэд тодорхой үүргийг хариуцуулна. Жишээлбэл, энэ хэсэг нь энэ бодлогын ийм үүргийг хариуцна гэх мэт. Ажлын тодорхой үүргүүдийг хуваарилах үйл ажиллагаа нь хүн төрөлхтний хамгийн түгээмэл арга байсаар ирсэн. Энэ аргыг *хуваагаад нэгтгэх (divide and conquer) арга* гэнэ. Програмчлалын хувьд үүрэг хариуцлагуудыг функцуудад хуваарилсанаар програм зохиогчид дараах давуу талууд бий болно. Үүнд:

- кодын давхцалаас зайлсхийнэ
- програмын дахин ашиглагдах чанарыг нэмэгдүүлнэ

- програмын хийсвэрлэлтийг нэмэгдүүлнэ.

Функц нь програмын тодорхой үүргийг гүйцэтгэх бөгөөд утга буцаадаг бол функц, утга буцаадаггүй бол процедур гэж нэрлэгдэнэ. Функц нь үүргээ гүйцэтгэхэд шаардлагатай мэдээллийг параметр хэлбэрээр авна. Параметр нь дуудаж байгаа функц болон дуудагдаж байгаа функц хооронд өгөгдөл солилцоход ашиглагдана. Жава хэлэнд функцийг дараах хэлбэрээр тодорхойлно.

### Функцийн тодорхойломж

```
public [Буцах утгын төрөл][Функцийн нэр] ([параметрууд])
{
    [Функцийн бие]
    return [буцах илэрхийлэл];
}
```

Тодорхойлсон функцээ дуудаж ашиглахдаа дараах хэлбэрээр бичнэ.

### Функц дуудалт

```
[Хувьсагч] = [Функцийн нэр] ([параметрууд]);
```

### Процедурын тодорхойломж

Жава хэлэнд процедурыг дараах хэлбэрээр тодорхойлно.

```
public void [Процедурын нэр] ([параметрууд])
{
    [Функцийн бие]
}
```

Тодорхойлсон процедураа дуудаж ашиглахдаа дараах хэлбэрээр бичнэ.

### Процедур дуудалт

```
[Функцийн нэр] ([параметрууд]);
```

Функцийн нэр жижиг үсгээр эхэлнэ. Жишээ нь: `maxOfList`. Функц дотор зарласан хувьсагчийг локаль хувьсагч гэнэ. Зөвхөн функц дотор ашиглана. Параметртэй ижил нэртэй байж болохгүй.

Анхаарах зүйлс:

- Функцийн буцааж байгаа утга буцаах өгөгдлийн төрөлтэй ижил байх ёстой.

- Функцэд дамжуулж байгаа параметруудийн тоо, дэс дараалал, тэдгээрийн өгөгдлийн төрлүүд функцийн хүлээж авах параметруудийн тоо, дэс дараалал, өгөгдлийн төрлүүдтэй ижил байх ёстой.

## Статик функц

Зарлалтандаа static түлхүүр үгийг ашигладаг ба классын объектыг үүсгэлгүйгээр классын нэрийг ашиглан дуудаж хэрэглэх боломжтой функцийг статик функц гэнэ. Статик функц нь объектод харъяалагддаггүй, класс-д харъяалагддаг. Зөвхөн статик хувьсагч болон функцүүд руу хандаж чадах ба main функц нь статик функц тул түүн дотроос статик функцийг л дуудаж болдог. Статик биш функц дотроос статик биш функц дуудна.

Жишээлбэл 10, 20 тооны ихийг нь олж хэвлэх жишээ дараах хэлбэртэй байна.

```
public class Program
{
    public static void main()
    {
        int k = max(10, 20);
        printer(k);
    }
    public static int max(int a, int b)
    {
        if (a > b)
            return a;
        else
            return b;
    }
    public static void printer(int s)
    {
        System.out.println(s);
    }
}
```

## Локаль хувьсагч

Функц, процедур дотор зарлаж, зөвхөн тухайн функц, процедурт ашиглаж байгаа хувьсагчдыг *локаль хувьсагч* гэнэ. Параметр нь дуудаж байгаа функц, процедураас оролт авахад ашиглагдах боловч нөгөө талаас тухайн функц, процедурын локаль хувьсагч юм. Иймээс параметрийн нэр нь локаль хувьсагчдын нэртэй ижил байж болохгүй. Жишээ нь, намар болон хаврын сургалтын кредит болон нэг кредитийн төлбөрийг параметр хэлбэрээр авч оюутны жилийн төлбөрийг тооцоолдог yearCost функцийг дараах хэлбэрээр тодорхойлов.

```

public class Program1
{
    public static int yearCost(int namar,int havar,int creditCost)
    {
        int yearCredit = namar + havar;
        int sumCost = yearCredit * yearCredit;
        return sumCost;
    }
    public static void main()
    {
        int sumCost = yearCost(14,17,30000);
    }
}

```

Энэ жишээнд **yearCredit** болон **sumCost** нь локаль хувьсагчид юм. Мөн **sumCost** гэсэн ижил нэртэй хоёр хувьсагч **main** болон **yearCost** функцэд байна. Гэвч эдгээр нь хоёр тусдаа функц тул үүнийг зөвшөөрдөг.

### Ажил гүйцэтгэх дараалал:

1. Нэг бодит классын өгөгдөл, үйлдлүүдийг тодорхойлж өгүүлбэрээр бичнэ.
2. Тухайн классын байгуулагчийг Жава хэл дээр бичнэ.
3. Байгуулагчийг олон хэлбэржүүлж бичнэ.
4. BlueJ хэрэгслээ ашиглан классаас объект үүсгэж, үйлдлүүдийг дуудна.
5. Ажиллуулагч класс ашиглан бичсэн классаас объект үүсгэж, үйлдлүүдийг дуудна.
6. Функц, процедур бичих сурах
7. Параметр, буцаах утгыг ашиглах
8. Статик функц бичиж, дуудах

### Суралцахуйн үр дүнг үнэлэх даалгаврууд:

1. Өмнөх лабораторид үүсгэсэн класст
  - a. олон хэлбэржилт бүхий байгуулагч функц 3-г бич.
2. Объект үүсгэх,
  - a. BlueJ ашиглан гараар олон хэлбэржилт бүхий байгуулагч функцүүдийг дуудах,
  - b. main функц бүхий класс бичиж, 2а-д дуудсан үйлдлүүдийн кодыг бич.
3. Өмнөх 3-р лабораторид үүсгэсэн класст:
  - a. функц, процедур, статик функц нэм.
  - b. олон хэлбэржилт бүхий функц нэм.

### Ашиглах материал:

- Ю.Намсрай, Т.Гантөр, Д.Ганцоож, Програмчлалын Жава хэл, 2015 он.
- Д.Энхжаргал, Java 2 Объект хандлагат програмчлал, 2009 он.