



# Өгөгдлийн бүтэц



Өгөгдлийн объект бол тохиолдол/утгын цуглуулга буюу олонлог юм. Жишээ нь:

$\text{integer} = \{0, +1, -1, +2, -2, +3, -3, \dots\}$

$\text{daysOfWeek} = \{S, M, T, W, Th, F, Sa\}$

# Өгөгдлийн объект

Тохиолдлууд хамааралтай, хамааралгүй байж болно

`myDataObject = {apple, chair, 2, 5.2, red, green, Jack}`



# Өгөглийн бүтэц



Өгөгдлийн объектын холбоо нь нэг  
буюу хэд хэдэн тохиолдол дээр  
хэрэгжих үйлдлээр илэрхийлэгддэг

add, subtract, predecessor, multiply

# Шугаман (Дараалсан) Жагсаалт

тохиолдлууд дараах хэлбэртэй байна:

$(e_0, e_1, e_2, \dots, e_{n-1})$

үүнд  $e_i$  жагсаалтын элемент

$n \geq 0$  – жагсаалт төгсгөлтэй

түүний хэмжээ нь  $n$



# Шугаман жагсаалт



$$L = (e_0, e_1, e_2, e_3, \dots, e_{n-1})$$

Холбоо:

$e_0$  – 0 дэх (эхний) элемент

$e_{n-1}$  – сүүлийн элемент

$e_i$  - элемент  $e_{i+1}$  элементийн өмнө орно

# Жишээ/Тохиолдол

CS204 –ийн оюутнууд =

(Бат, Гэрэл, Дулмаа, Оюун, ..., Ганбат)

Days of Week = (S, M, T, W, Th, F, Sa)

Months = (Jan, Feb, Mar, Apr, ..., Nov, Dec)

Шугаман жагсаалтын үйлдэл—size()

Жагсаалтын хэмжээг олох

$$L = (a, b, c, d, e)$$

$$\text{ХЭМЖЭЭ} = 5$$

# Шугаман жагсаалтын үйлдэл—`get(theIndex)`

Өгөгдсөн индексийн элементийг авах

$$L = (a, b, c, d, e)$$

$$\text{get}(0) = a$$

$$\text{get}(2) = c$$

$$\text{get}(4) = e$$

$$\text{get}(-1) = \text{error}$$

$$\text{get}(9) = \text{error}$$



# Шугаман жагсаалтын үйлдэл — `indexOf(theElement)`

Элементийн индексийг олох

$$L = (a, b, d, b, a)$$


$$\text{indexOf}(d) = 2$$

$$\text{indexOf}(a) = 0$$

$$\text{indexOf}(z) = -1$$

# Шугаман жагсаалтын үйлдэл — `remove(theIndex)`

Өгөгдсөн индексийн элементийг устгаж,  
буцаах


$$L = (a, b, \underline{c}, d, e, f, g)$$

*remove(2)* -с *c* буцах ба

$L = (a, b, d, e, f, g)$  болно

$d, e, f, g$  –ийн индекс 1 –ээр багасна

# Шугаман жагсаалтын үйлдэл — `remove(theIndex)`

Өгөгдсөн индексийн элементийг устгаж,  
буцаах

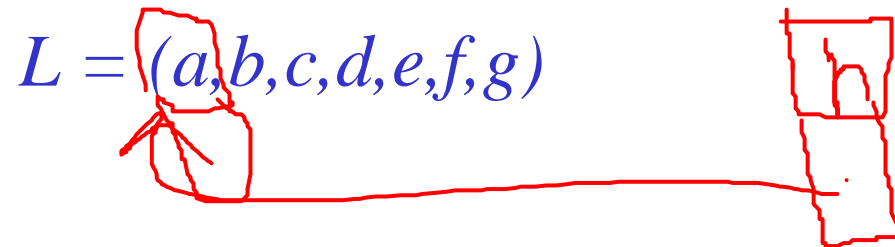
$L = (a, b, c, d, e, f, g)$

`remove(-1)` => error

`remove(20)` => error

# Шугаман жагсаалтын үйлдэл — $\text{add}(\text{theIndex}, \text{theElement})$

Өгөгдсөн индексээр шинэ элемент  
НЭМЭХ



$\text{add}(0, h) \Rightarrow L = (h, a, b, c, d, e, f, g)$

$a, b, c, d, e, f, g$  –ийн индекс 1-ээр  
НЭМЭГДЭНЭ

# Шугаман жагсаалтын үйлдэл — `add(theIndex, theElement)`

$$L = (a, b, c, d, e, f, g)$$

$$\text{add}(2, h) \Rightarrow L = (a, b, h, c, d, e, f, g)$$

$c, d, e, f, g$  –ийн индекс 1-ээр нэмэгдэнэ

$$\text{add}(10, h) \Rightarrow \text{error}$$

$$\text{add}(-6, h) \Rightarrow \text{error}$$

# Өгөгдлийн бүтцийн тодорхойлолт

## □ Хэлнээс хамаарахгүй

- Өгөгдлийн хийсвэр төрөл(Abstract Data Type)

## □ Java

- Интерфейс(Interface)
- Хийсвэр класс(Abstract Class)

# Шугаман жагсаалтын хийсвэр өгөгдлийн төрөл

ХийсвэрӨгөгдлийнТөрөл *LinearList*

{

ТОХИОЛДОЛ

дараалсан элентийн цуглуулга(хоосон байж болно)

ҮЙЛДЛҮҮД

*isEmpty()*: жагсаалт хоосон бол true, үгүй бол false буцна

*size()*: хэмжээг буцаах (ө.х. жагсаалтын элементийн тоо)

*get(index)*: *index* дэх элементийг буцаах

*indexOf(x)*: эхэлж олдсон *x* –ийн индексийг буцаана.

байхгүй бол -1 буцна

*remove(index)*: *index* дэх элементийг устгаж, буцаана

устгагдсанаас хойшхи элементийн индекс 1-ээр багасна

*add(theIndex, x)*: *x* –ийг *index* дэх элемент болгож оруулах

оруулснаас хойшхи элементийн индекс 1-ээр ихэснэ

*output()*: элементүүдийг зүүнээс баруун тийш гаргах

}

# Шугаман жагсаалтын Java интерфейс

Интерфейст тогтмолууд болон хийсвэр аргууд(хэрэгжилт нь байхгүй арга) орж болно



# Шугаман жагсаалтын Java интерфейс

```
public interface LinearList
{
    public boolean isEmpty();
    public int size();
    public Object get(int index);
    public int indexOf(Object elem);
    public Object remove(int index);
    public void add(int index, Object obj);
    public String toString();
}
```

# Интерфейсийг хэрэгжүүлэх

```
public class ArrayLinearList implements LinearList
{
    // бүх аргууд энд бичигдэнэ
}
```

# Шугаман жагсаалтын хийсвэр класс

Хийсвэр класст тогтмолууд,  
хувьсагчид, хийсвэр болон хийсвэр  
бус аргууд байж болно

# Шугаман жагсаалтын хийсвэр класс

```
public abstract class LinearListAsAbstractClass
{
    public abstract boolean isEmpty();
    public abstract int size();
    public abstract Object get(int index);
    public abstract int indexOf(Object theElement);
    public abstract Object remove(int index);
    public abstract void add(int index,
                             Object theElement);
    public abstract String toString();
}
```

# Java классыг өргөтгөх

```
public class ArrayLinearList
    extends LinearListAsAbstractClass
{
    // бүх хийсвэр классууд энд бичигдэнэ
}
```

# Олон интерфэйсийг хэрэгжүүлэх

```
public class MyInteger implements Operable, Zero,  
                                   CloneableObject  
{  
    // Operable, Zero, CloneableObject –ийн  
    // бүх аргууд энд бичигдэнэ  
}
```



# Олон классыг өргөтгөх



## **JAVA –д үүнийг зөвшөөрөхгүй**

Класс олон интерфейсийг хэрэгжүүлж  
болох боловч зөвхөн 1 классыг өргөтгөж  
болно.

# Өгөгдлийг дүрслэх аргууд



массив --- Бүлэг 5

холбоос --- Бүлэг 6

заагч --- Бүлэг 7



# Шугаман жагсаалтын массив дүрслэл

Нэг хэмжээст массив - `element[]`

a	b	c	d	e										
0	1	2	3	4	5	6								

$L = (a, b, c, d, e)$

$i$  дэх элементийг `element[i]` –д хадгална.

# Баруунаас зүүн тийш байршуулах



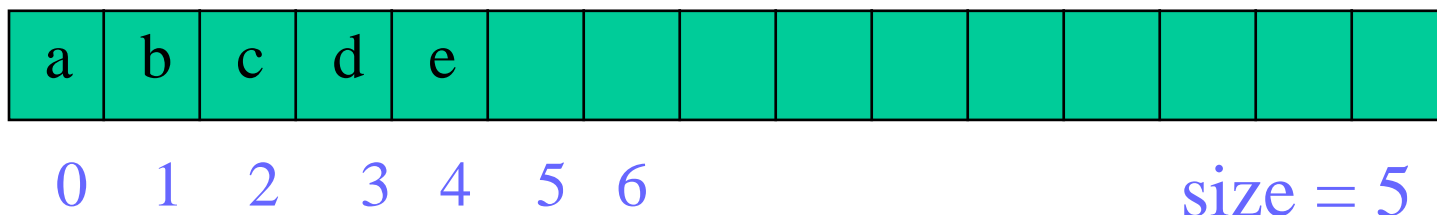
# Байршил алгасаж байршуулах



# Тойргоор байршуулах



# Бидний ашиглах дүрслэл



$i$  дэх элементийг `element[i]` –д хадгална

`size` элементийн тоог илэрхийлэхэд энэ хувьсагчийг ашиглана

# Элемент нэмэх/хасах

size = 5

a	b	c	d	e										
---	---	---	---	---	--	--	--	--	--	--	--	--	--	--

add(1,g)

size = 6

a	g	b	c	d	e									
---	---	---	---	---	---	--	--	--	--	--	--	--	--	--

# element[] массивын өгөгдлийн төрөл

Жагсаалтын өгөгдлийн төрөл тодорхойгүй.

element[] -өгөгдлийн төрлийг **Object** гэж  
тодорхойлье.

Манай жагсаалтанд энгийн төрлүүд (**int**,  
**float**, **double**, **char**, г.м.) орохгүй.

# element[] массивын урт

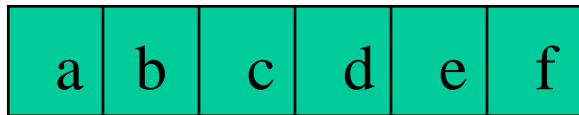
Жагсаалтанд хэдэн элемент байхыг мэдэхгүй.

Анхны уртыг нь зааж өгөөд хэрэгцээгээрээ  
нэмье.



# Массивын уртыг нэмэх

`element[]` массивын урт 6.



Эхлээд илүү урттай шинэ массив үүсгэнэ

`newArray = new Object[15];`



# Массивын уртыг нэмэх

Элементүүдийг хуучнаас шинэ массивт хуулна.

a	b	c	d	e	f
---	---	---	---	---	---

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

a	b	c	d	e	f									
---	---	---	---	---	---	--	--	--	--	--	--	--	--	--

# Массивын уртыг нэмэх

Эцэст нь шинэ массивын нэрийг  
өөрчилнө.

```
element = newArray;
```

element[0]



```
element.length = 15
```

# Нэг дор бичвэл

// зөв төрөл, урттай шинэ массив үүсгэх

```
Object [] newArray = new Object [newLength];
```

// элементүүдийг хуулах

```
System.arraycopy(element, 0, newArray, 0,  
    element.length);
```

// нэрийг өөрчлөх

```
element = newArray;
```

```
public static Object [] changeLength(Object [] a,  
                                     int newLength)  
{  
    Object [] newArray = new Object [newLength];  
    System.arraycopy(...);  
    return newArray;  
}
```

// erroneous

# ArrayList - класс



- Шугаман жагсаалтын нийтлэг зорилготой хэрэгжүүлэлт.
- Жагсаалтын элементийн тоо тодорхойгүй.

# Хоосон жагсаалтыг үүсгэх

```
ArrayLinearList a = new ArrayLinearList(100),  
                b = new ArrayLinearList(),  
                c;  
LinearList d = new ArrayLinearList(1000),  
            e = new ArrayLinearList(),  
            f;
```

# Шугаман жагсаалтыг ашиглах

```
System.out.println(a.size());
```

```
a.add(0, Integer(2));
```

```
b.add(0, Integer(4));
```

```
System.out.println(a);
```

```
b.remove(0);
```

```
if (a.isEmpty())
```

```
    a.add(0, Integer(5));
```



# Шугаман жагсаалтын массив

```
LinkedList [] x = new LinkedList [4];
```

```
x[0] = new ArrayLinkedList(20);
```

```
x[1] = new Chain();
```

```
x[2] = new Chain();
```

```
x[3] = new ArrayLinkedList();
```

```
for (int i = 0; i < 4; i++)
```

```
    x[i].add(0, new Integer(i));
```

# ArrayLinearList класс

```
/** LinearList –ийн хэрэгжүүлэлт */  
package dataStructures;  
import java.util.*; // Iterator интерфейс  
import utilities.*; // массивын хэмжээг өөрчлөх класс  
  
public class ArrayLinearList implements LinearList  
{  
    // өгөгдөл гишүүд  
    protected Object [] element; // массив  
    protected int size; // элементийн тоо  
    // байгуулагч болон бусад аргууд  
}
```

# Байгуулагч



```
/** initialCapacity хэмжээтэй жагсаалт үүсгэх  
 * @throws initialCapacity < 1 бол  
 * IllegalArgumentException унана  
 */  
public ArrayLinearList(int initialCapacity)  
{  
    if (initialCapacity < 1)  
        throw new IllegalArgumentException  
            ("initialCapacity must be >= 1");  
    // хэмжээний анхны утга 0  
    element = new Object [initialCapacity];  
}
```

# Өөр байгуулагч



*/\*\* 10 –ын урттай жагсаалт \*/*

```
public ArrayLinearList()  
{  
    this(10);  
}
```

isEmpty арга



```
/** @return хоосон бол true */  
public boolean isEmpty()  
{return size == 0;}
```

size() арга

```
/** @return жагсаалтын ХЭМЖЭЭ */
```

```
public int size()  
{ return size; }
```

## checkIndex арга

```
/** @throws index хэмжээнээс хальбал  
 * IndexOutOfBoundsException */  
void checkIndex(int index)  
{  
    if (index < 0 || index >= size)  
        throw new IndexOutOfBoundsException  
            ("index = " + index + " size = " + size);  
}
```

## get арга

```
/** @return index –ийн элемент  
 * @throws index хэмжээнээс халььбал  
 * IndexOutOfBoundsException унана */  
public Object get(int index)  
{  
    checkIndex(index);  
    return element[index];  
}
```



# indexOf арга

```
/** @return theElement элементийн индекс,  
    * байхгүй бол -1 */  
public int indexOf(Object theElement)  
{  
    // theElement –г element[] –аас хайх  
    for (int i = 0; i < size; i++)  
        if (element[i].equals(theElement))  
            return i;  
  
    // theElement олдсонгүй  
    return -1;  
}
```

## remove арга

```
public Object remove(int index)
{
    checkIndex(index);

    // индекс зөв бол элементүүдийг шилжүүлэх
    Object removedElement = element[index];
    for (int i = index + 1; i < size; i++)
        element[i-1] = element[i];

    element[--size] = null; // хаягдалд өгөх
    return removedElement;
}
```

add apra

```
public void add(int index, Object theElement)
{
    if (index < 0 || index > size)
        // байршил буруу
        throw new IndexOutOfBoundsException
            ("index = " + index + " size = " + size);

    // индекс зөв, сул зай бий юү?
    if (size == element.length)
        // байхгүй бол 2 дахин ихэсгэх
        element = ChangeArrayLength.changeLength1D(element, 2 *
            size);
}
```

## add арга

// элементүүдийг баруун тийш гүйлгэх

```
for (int i = size - 1; i >= index; i--)
```

```
    element[i + 1] = element[i];
```

```
element[index] = theElement;
```

```
size++;
```

```
}
```

## Баруун тийш хурдан гүйлгэх арга

```
System.arraycopy(element, index, element,  
                  index + 1, size - index);
```

# Хэлхээст хөрвүүлэх

```
public String toString()
{
    StringBuffer s = new StringBuffer("[");
    // буферт элементийг хийх
    for (int i = 0; i < size; i++)
        if (element[i] == null) s.append("null, ");
        else s.append(element[i].toString() + ", ");
    if (size > 0) s.delete(s.length() - 2, s.length());
    // сүүлийн ", " -г устгах
    s.append("]");
    // эквивалент хэлхээс үүсгэх
    return new String(s);
}
```