ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ
МЭДЭЭЛЭЛ, ХОЛБООНЫ ТЕХНОЛОГИЙН СУРГУУЛЬ

F.CS202
ОБЪЕКТ ХАНДЛАГАТ ПРОГРАМЧЛАЛ

Лекц 7

Массив

док., дэд проф. Б.Батзолбоо
маг. Б.Мөнхбуян

2021 он

Агуулга

- Массив үүсгэх, массивын элементүүдтэй ажиллах
- Олон хэмжээст массив үүсгэж ашиглах
- ArrayList ашиглан объектын олонлогийг боловсруулах
- Жагсаалт, буулгалт, олонлог

Өмнөх лекцээр

- Объектуудын бүрдмэл харьцаа
  - Composition
  - Aggregation

Анхдагч өгөгдлийн төрлүүд ашиглан массив зарлах

- Зарлалт

```
<өгөгдлийн төрөл> [ ] <хувьсагч>          //хувилбар 1
<өгөгдлийн төрөл> <хувьсагч>[ ]           //хувилбар 2
```

- Үүсгэх

```
<хувьсагч> = new <өгөгдлийн төрөл> [ <хэмжээ> ]
```

- Жишээ

Хувилбар 1

```
double[ ] rainfall;
rainfall = new double[12];
```

Хувилбар 2

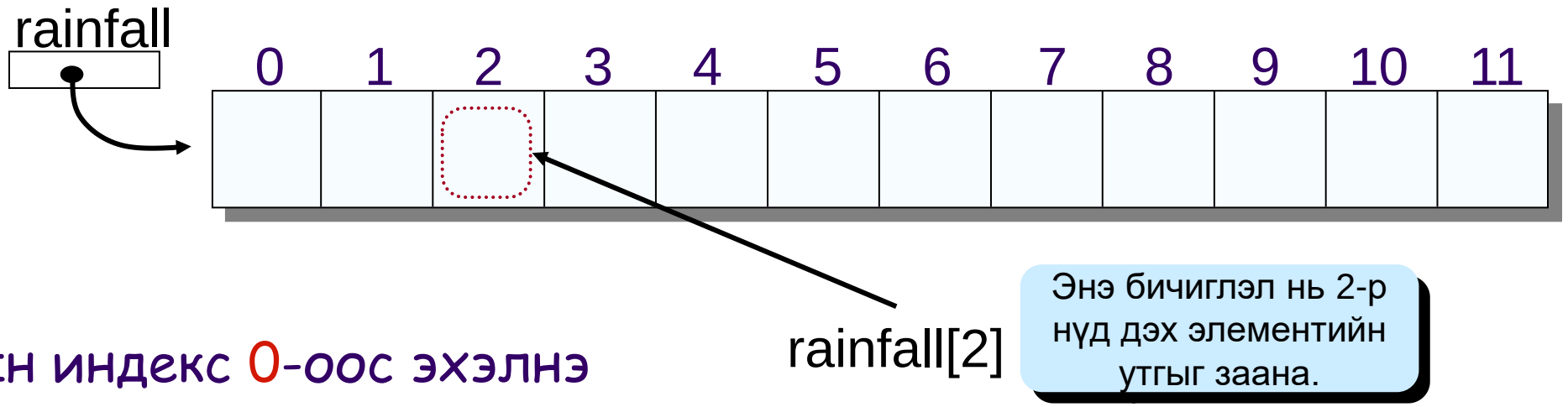
```
double rainfall [ ];
rainfall = new double[12];
```

массив нь объекттой ижил!

Массивын элементүүдэд хандах

- массив дахь элементэд хандахдаа индексээр хандана.

```
double[] rainfall = new double[12];
```



массивын индекс 0-оос эхэлнэ

Массив боловсруулах– Жишээ 1

```
Scanner scanner = new Scanner(System.in);
double[] rainfall = new double[12];

double annualAverage,
          sum = 0.0;

for (int i = 0; i < rainfall.length; i++) {

    System.out.print("Rainfall for month " + (i+1));
    rainfall[i] = scanner.nextDouble( );
    sum += rainfall[i];
}

annualAverage = sum / rainfall.length;
```

length тогтмол нь массивын хэмжээг буцаана.

Массив боловсруулах– Жишээ 2

```
Scanner scanner = new Scanner(System.in);
double[] rainfall = new double[12];
String[] monthName = new String[12];
monthName[0] = "January";
monthName[1] = "February";
…
double annualAverage, sum = 0.0;

for (int i = 0; i < rainfall.length; i++) {
    System.out.print("Rainfall for " + monthName[i] + ": ");
    rainfall[i] = scanner.nextDouble();
    sum += rainfall[i];
}
annualAverage = sum / rainfall.length;
```

Дараагийн саруудыг ижил бичиглэлээр үүсгэнэ.

Тухайн индексд байрлах сарын нэрийг харуулна

Массив боловсруулах– Жишээ 3

- Улирал бүрийн хур тунадасын дундаж хэмжээг тооцоолох

```
//rainfall массивыг зарлаж анхны утга олгосон гэж үзье

double[] quarterAverage = new double[4];


for (int i = 0; i < 4; i++) {
    sum = 0;
    for (int j = 0; j < 3; j++) {
                              //улирлын дунджийг
       sum += rainfall[3*i + j];      //бодох
    }
    quarterAverage[i] = sum / 3.0; //Улирлын(i+1) дундаж
}
```

Массивт утга олгох

- Бусад өгөгдлийн төрлийн адил массивыг зарлах үедээ утга олгох боломжтой.

```
int[] number = { 2, 4, 6, 8 };

double[] samplingData = { 2.443, 8.99, 12.3, 45.009, 18.2,
                          9.00, 3.123, 22.084, 18.08 };

String[] monthName = {  "January", "February", "March",
                        "April", "May", "June", "July",
                        "August", "September", "October",
                        "November", "December" };
```

```
number.length       ⟶ 4
samplingData.length ⟶ 9
monthName.length    ⟶ 12
```

Массивын хэмжээг хувьсагчаар дамжуулж тодорхойлох

- Массивыг зарлахдаа хэмжээг тодорхойлохгүй байж болно.
- Доорх жишээнд массивын хэмжээг гараас авсан утгаар тодорхойлж байна:

```
Scanner scanner = new Scanner(System.in);
int size;
int[] number;

System.out.print("Size of an array:"));
size= scanner.nextInt( );

number = new int[size];
```

Объект хадгалах массив

- Жава хэлэнд өгөгдлийн анхдагч төрөлтэй массив зарлахаас гадна объект төрөлтэй массив зарлаж болно.
- Энэ нь улам хүчирхэг кодчиллыг бий болгодог.
- Объект агуулах массив үүсгэж ашиглах нь илүү цэвэр, логиктой програмын зохиомж гаргах боломжийг олгоно.

Person класс

- Объект агуулах массивын жишээ

```
Person latte;

latte = new Person( );
latte.setName("Ms. Latte");
latte.setAge(20);
latte.setGender('F');

System.out.println( "Name: " + latte.getName() );
System.out.println( "Age : " + latte.getAge() );
System.out.println( "Gender : " + latte.getGender() );
```

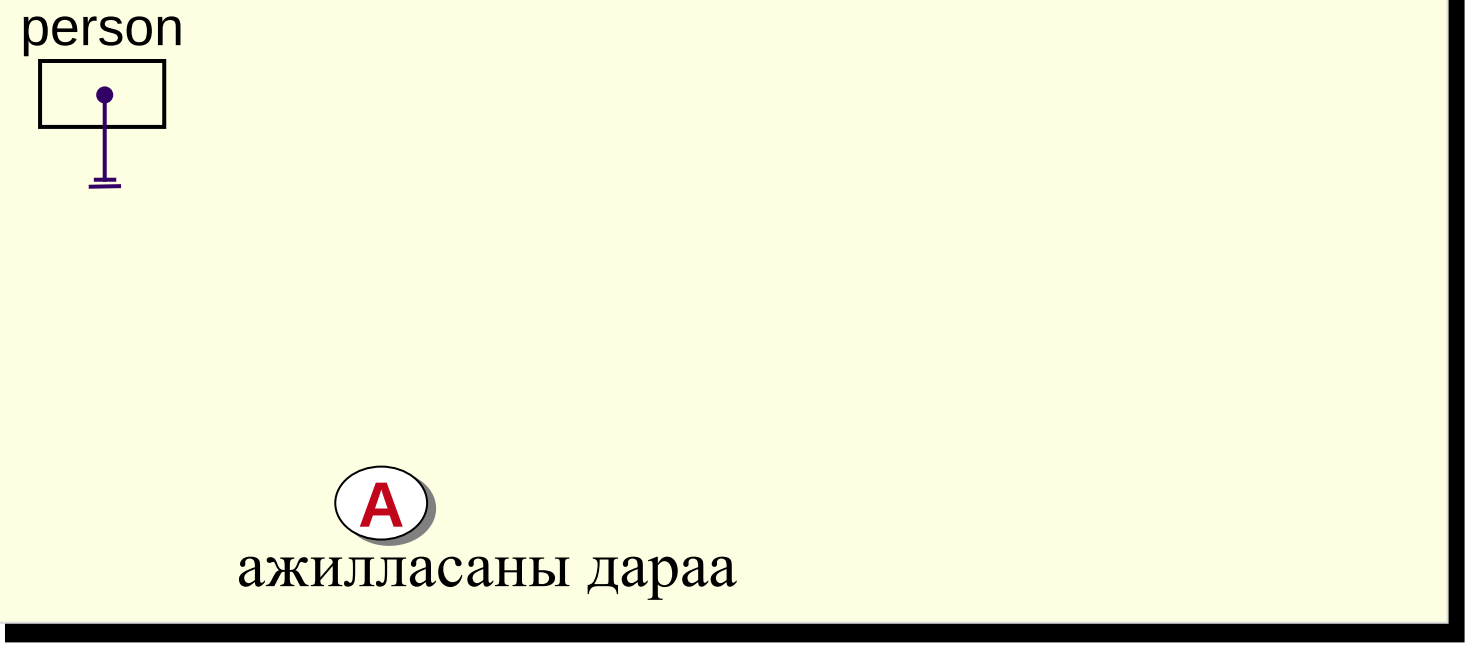
Person класс нь set, get методуудтай.

Объект массив үүсгэх - 1


```
Person[ ] person;

person = new Person[20];

person[0] = new Person( );
```

Зөвхөн Person гэсэн нэр зарлагдаж байна. Санах ойд ямар ч массив ачаалагдахгүй.

Санах ойн төлөв



Объект массив үүсгэх - 2

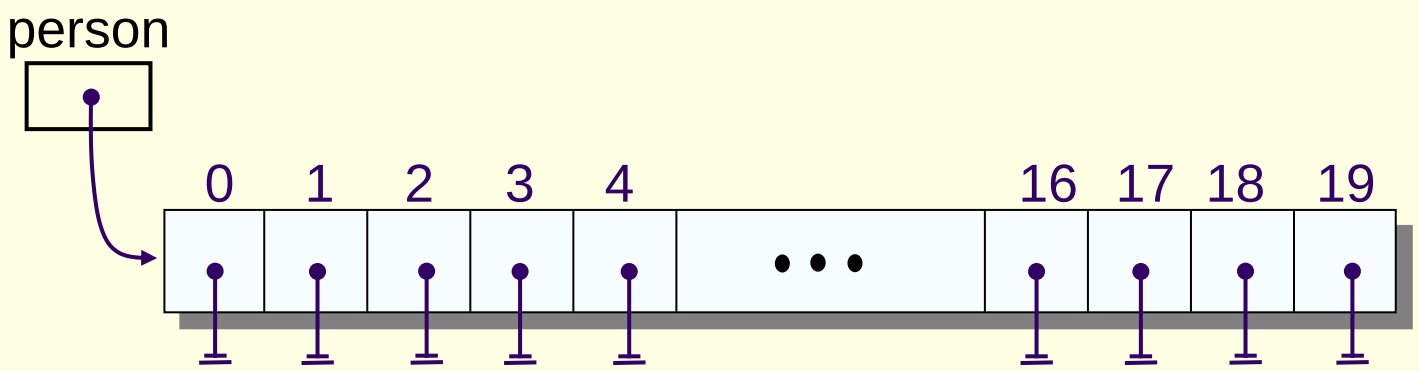
Код

```
Person[ ] person;
person = new Person[20];
person[0] = new Person( );
```

В

20 Person объектыг агуулах массив үүснэ. Гэхдээ Person объектууд нь үүсээгүй байна.

Санах ойн төлөв



В ажилласаны дараа

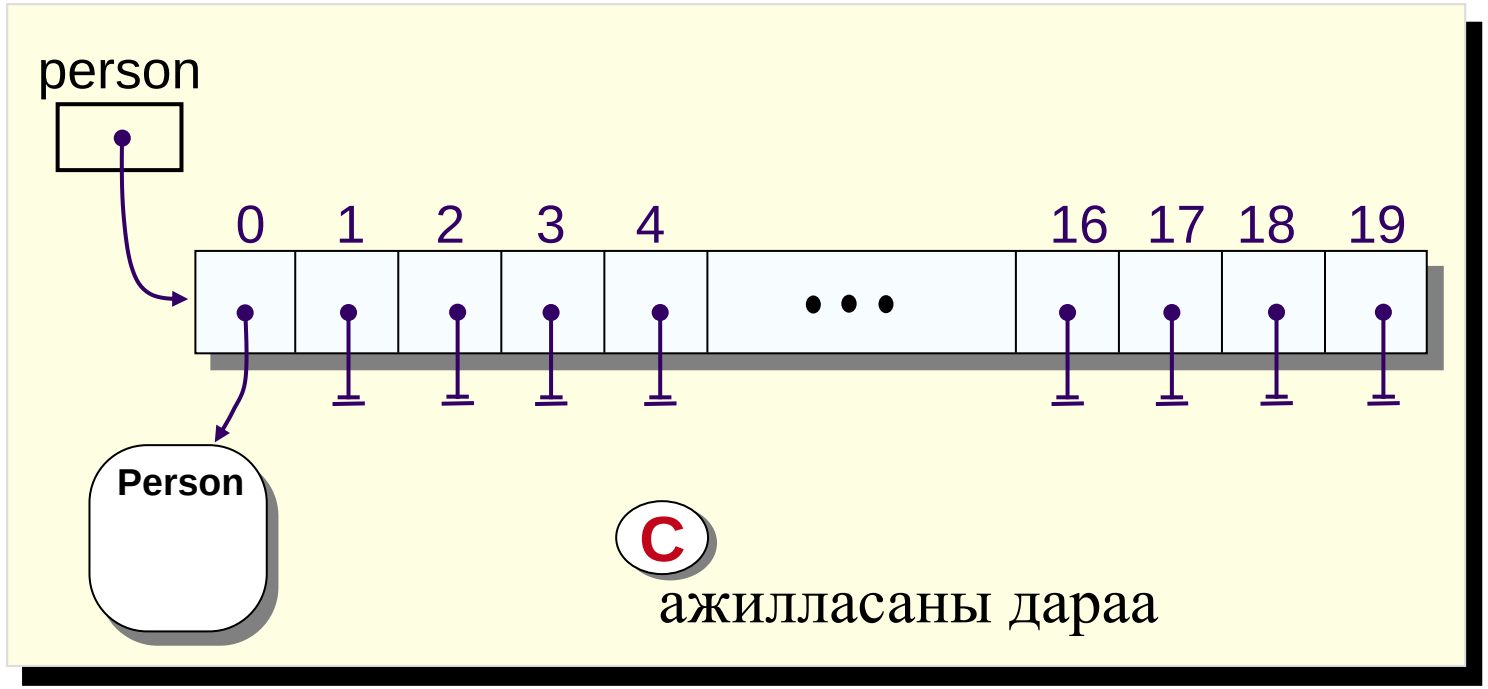
Объект массив үүсгэх - 3

Код

```
Person[ ] person;

person = new Person[20];

person[0] = new Person( );
```

Нэг Person объект үүсгэж, уг объект руу холбогдох холбоос 0 индекс дээр байрлана.

Санах ойн төлөв



Person массивтай ажиллах – Жишээ 1

- Person объектыг үүсгэж, person массивт хадгалах

```
String name, inpStr; int age; char gender;
Scanner scanner = new Scanner(System.in);

for (int i = 0; i < person.length; i++) {
    System.out.print("Enter name:"); name = scanner.next ( );
    System.out.print("Enter age:"); age = scanner.nextInt( );
    System.out.print("Enter gender:"); inpStr = scanner.next( );
    gender = inpStr.charAt(0);

    person[i] = new Person( ); //шинээр Person үүсгэж утгуудыг олгоё
    person[i].setName ( name );
    person[i].setAge ( age );
    person[i].setGender( gender );
}
```

Person массивтай ажиллах – Жишээ 2

- Хамгийн залуу болон хамгийн настай хүнийг олох.

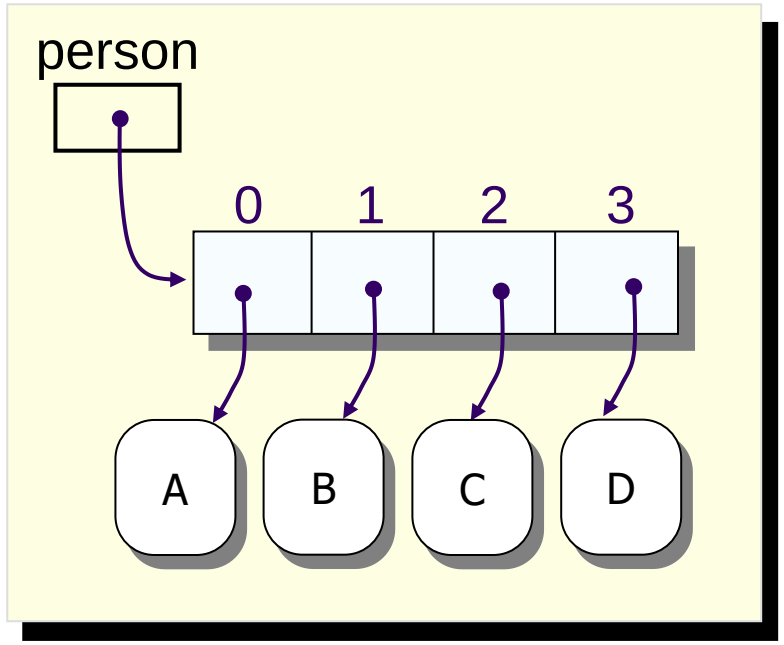
```
int minIdx = 0; //хамгийн залуу хүний индексийг хадгалах хувьсагч
int maxIdx = 0;   //хамгийн настай хүний индексийг хадгалах

for (int i = 1; i < person.length; i++) {

    if ( person[i].getAge() < person[minIdx].getAge() ) {
        minIdx = i;   //хамгийн залуу хүнийг олох

 } else if (person[i].getAge() > person[maxIdx].getAge() ) {

        maxIdx = i;   //хамгийн настай хүний олох
    }
}
//person[minIdx] нь хамгийн залуу, person[maxIdx] хамгийн настай нь
```

Объект устгах– Арга 1

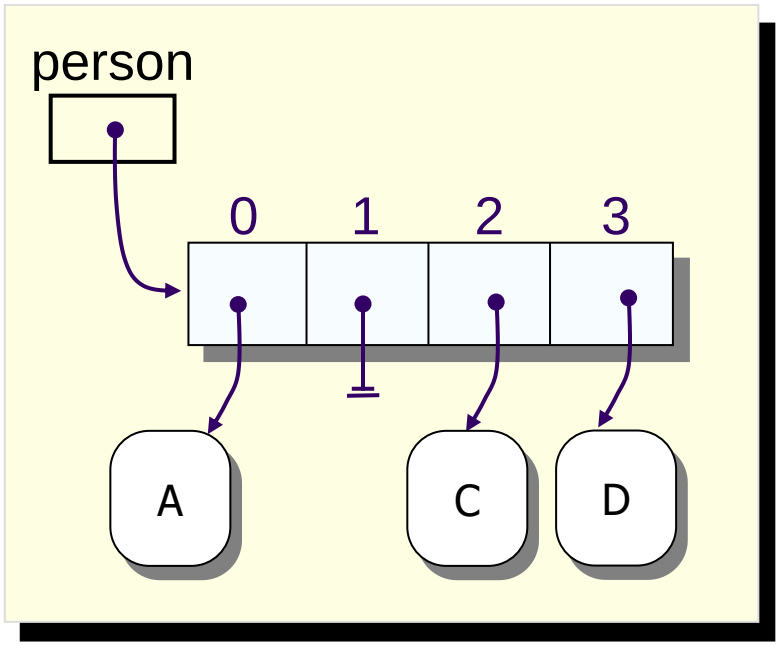
```
int delIdx = 1;
person[delIdx] = null;
```

A

Person B-г устгах бол 1-р байрлал дахь холбоосыг null болгоно.



A ажиллахын өмнө



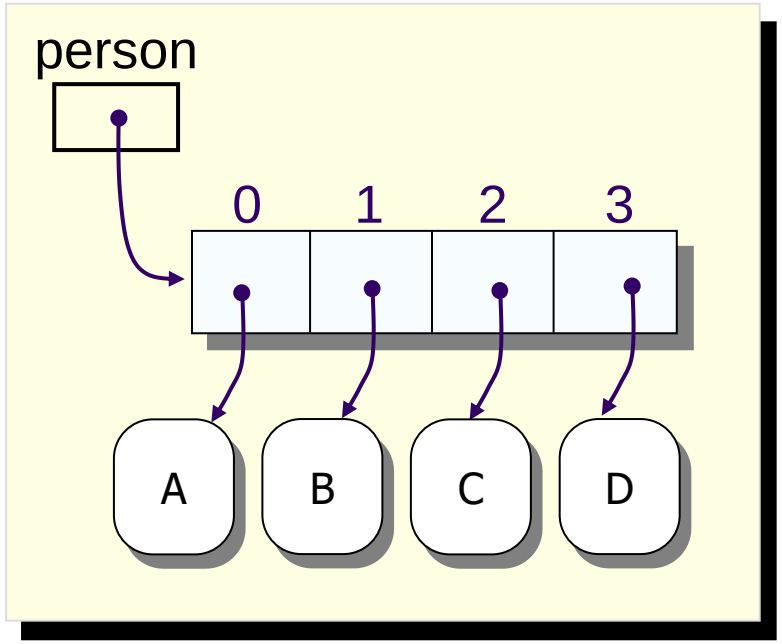
A ажилласаны дараа

Объект устгах– Арга 2

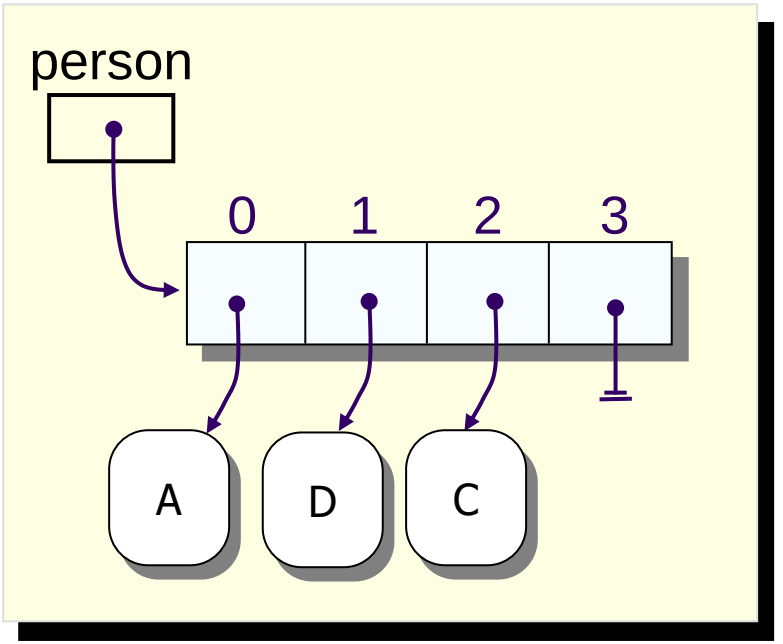
A

```
int delIdx = 1, last = 3;
person[delIndex] = person[last];
person[last] = null;
```

Person В-г устгахдаа 1-р байрлал дахь холбоосыг сүүлчийн байрлал руу заах.



A ажиллахын өмнө



A ажилласаны дараа

Person массивтай ажиллах – Жишээ 3

- Өгөгдсөн нэртэй хүнийг хайх.

```
int i = 0;

while ( person[i] != null && !person[i].getName().equals("Latte") ) {
    i++;
}
if ( person[i] == null ) {
    //өгөгдсөн нэртэй хүн олдоогүй бол
    System.out.println("Ms. Latte нэртэй хүн байхгүй");
} else {
    //өгөгдсөн нэртэй хүн олдсон бол
     System.out.println("Ms. Latte " + i + "-р байрлалд олдлоо");
}
```

For-Each давталт

- Энэ давталтыг Java 5.0-ээс эхлэн ашиглах болсон
- for-each давталт нь коллекц дахь элементүүдтэй ажиллахад хялбар болгосон
- массив дахь элементүүдэд хандах жишээ:

```
int sum = 0;


for (int i = 0; i < number.length; i++)
{
    sum = sum + number[i];
}
```

стандарт for давталт

```
int sum = 0;


for (int value : number) {
    sum = sum + value;
}
```

for-each давталт

For-Each давталт ашиглан массив дахь объекттой ажиллах

```
Person[] person = new Person[100];
//person[0] -ээс person[99] хүртэлх Person үүсгэнэ
```

```
for (int i = 0; i < person.length; i++) {
    System.out.println(person[i].getName());
}
```

стандарт for давталт

```
for (Person p : person) {
    System.out.println(p.getName());
}
```

for-each давталт

For-Each: Анхаарах зүйлс

- for-each давталт нь зөвхөн унших үйлдлийг дэмждэг. Элементүүдийн утгыг өөрчлөх боломжгүй.
- Нэг for-each давталтаар нэг л массивт хандана. Иймд нэг for-each давталт дотор хэд хэдэн массивт зэрэг хандах боломжгүй.
- for-each давталт нь массивын бүх элементүүдэд эхнээс нь дуустал нэг бүрчлэн ханддаг. Иймд элемент алгасах боломжгүй.

Метод руу объект дамжуулах – 1 –р алхам

Код

A

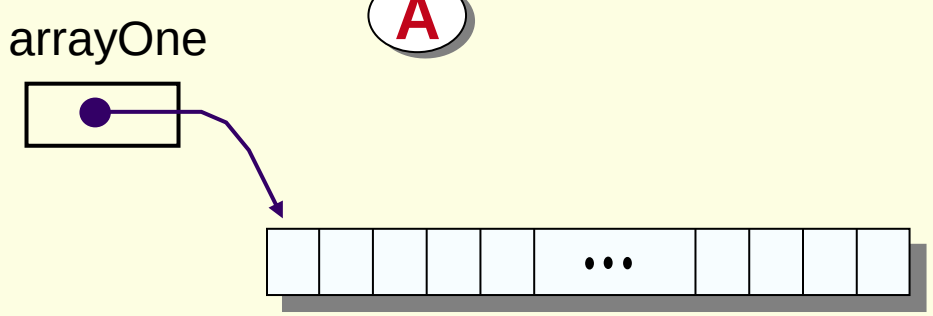
```
minOne = searchMinimum(arrayOne);
```

```
public int searchMinimum(float[] number))
{
    …
}
```

Санах ойн төлөв

searchMinimum ажиллахаас өмнө буюу цэг дэх

A



A. number локаль хувьсагч нь методыг ажиллуулахаас өмнө үүсээгүй байна.

Метод руу объект дамжуулах - 2 –р алхам

Код

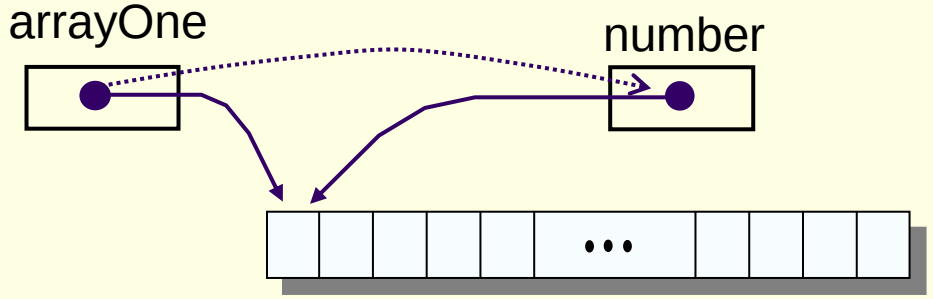
```
minOne = searchMinimum(arrayOne);
```

```
public int searchMinimum(float[] number))
{
    …
}
```

B

Санах ойн төлөв

B Цэгт массивын хаяг хуулагдана



B. Аргументийн утга буюу массивын хаяг нь параметрт олгогдоно(хуулагдана).

Метод руу объект дамжуулах - 3 –р алхам

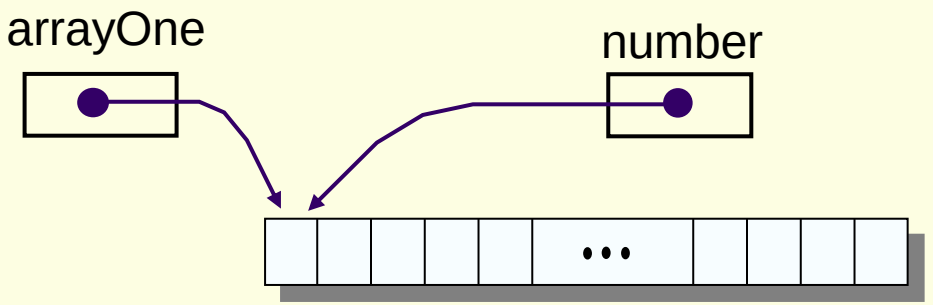
Код

```
minOne = searchMinimum(arrayOne);
```

```
public int searchMinimum(float[] number))
{
    …
}
```

C

C Метод дотор



C. Number параметрээр дамжуулан метод дотроос массивт хандана.

Санах ойн төлөв

Метод руу объект дамжуулах - 4 –р алхам

Код

```
minOne = searchMinimum(arrayOne);
```

D

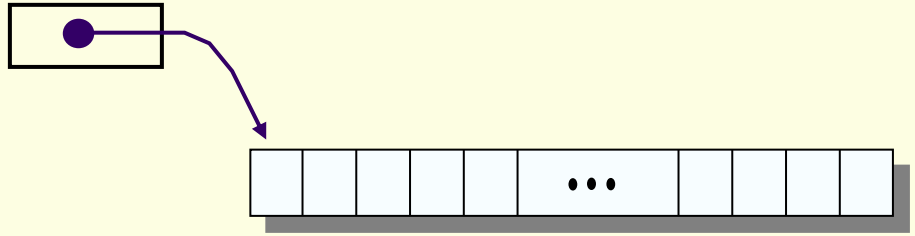
```
public int searchMinimum(float[] number))
{
    …
}
```

Санах ойн төлөв

searchMinimum ажилласаны дараа цэг дээр

D

arrayOne



D. Параметр устсан байна. Гэхдээ аргумент нь ижил объектыг заасаар байна.

Хоёр хэмжээст массив

- tabular мэдээллийг дүрслэхэд ихэвчлэн хэрэглэнэ.

Distance Table (in miles)

	Los Angeles	San Francisco	San Jose	San Diego	Monterey
Los Angeles	—	600	500	150	450
San Francisco	600	—	100	750	150
San Jose	500	100	—	650	50
San Diego	150	750	650	—	600
Monterey	450	150	50	600	—

Multiplication Table

	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
4	4	8	12	16	20	24	28	32	36
5	5	10	15	20	25	30	35	40	45
6	6	12	18	24	30	36	42	48	54
7	7	14	21	28	35	42	49	56	63
8	8	16	24	32	40	48	56	64	72
9	9	18	27	36	45	54	63	72	81

Tuition Table

	Day Students	Boarding Students
Grades 1 – 6	$ 6,000.00	$ 18,000.00
Grades 7 – 8	$ 9,000.00	$ 21,000.00
Grades 9 – 12	$ 12,500.00	$ 24,500.00

2 хэмжээст массив зарлах, үүсгэх

Зарлалт

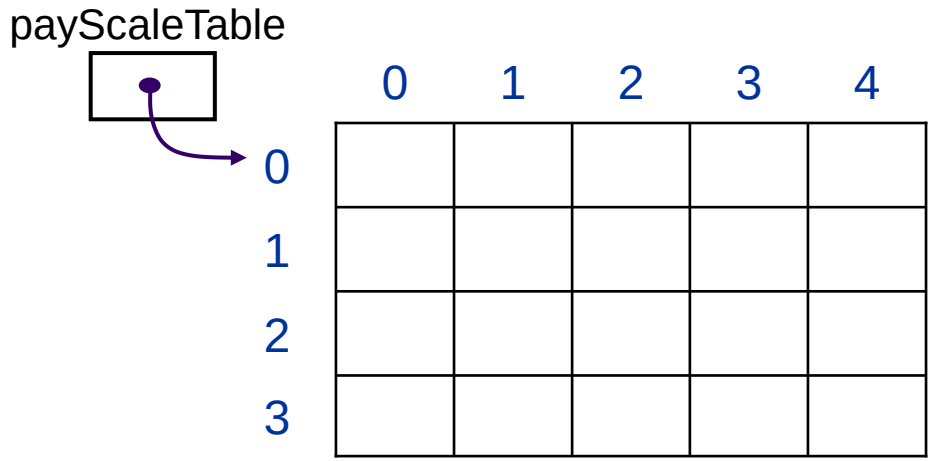
```
<өгөгдлийн төрөл> [][] <хувьсагч>      //хувилбар 1
<өгөгдлийн төрөл> <хувьсагч>[][]       //хувилбар 2
```

Үүсгэх

```
<хувьсагч> = new <өгөгдлийн төрөл> [ <хэмжээ1> ][ <хэмжээ2> ]
```

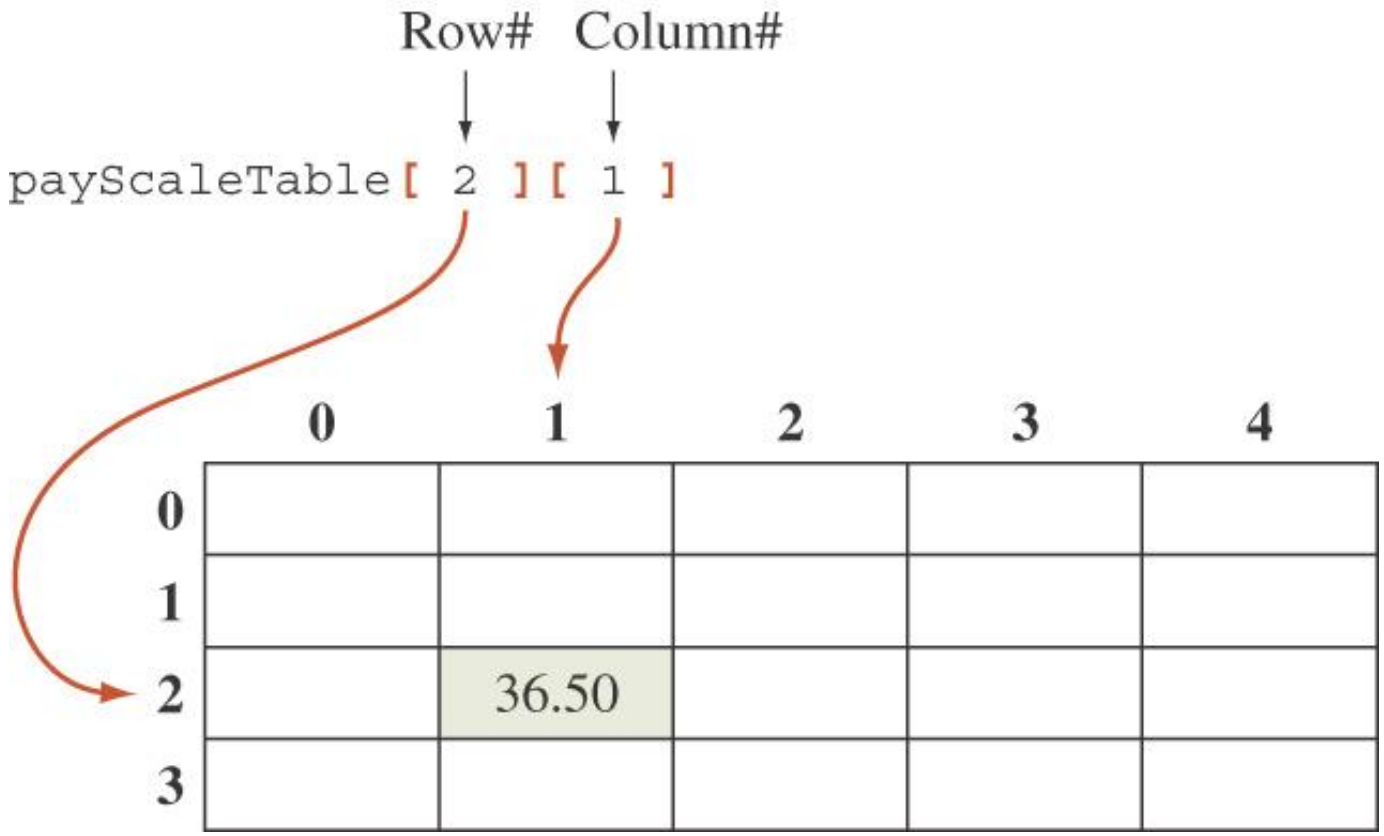
Жишээ

```
double[][] payScaleTable;
payScaleTable = new double[4][5];
```



Элементэд хандах

- Хоёр хэмжээст массивын элементэд мөр, баганы дугаараар хандана.



2 хэмжээст массивтай ажиллах

- Мөрүүдийн дунджийг бодох.

```
double[ ] average = { 0.0, 0.0, 0.0, 0.0 };
for (int i = 0; i < payScaleTable.length; i++) {
   for (int j = 0; j < payScaleTable[i].length; j++) {
      average[i] += payScaleTable[i][j];
   }
   average[i] = average[i] / payScaleTable[i].length;
}
```

2 хэмжээст массивыг Жава хэлэнд хэрэглэх

- массив үүсгэх

```
payScaleTable = new double[4][5];
```

Дэлгэрэнгүйгээр:

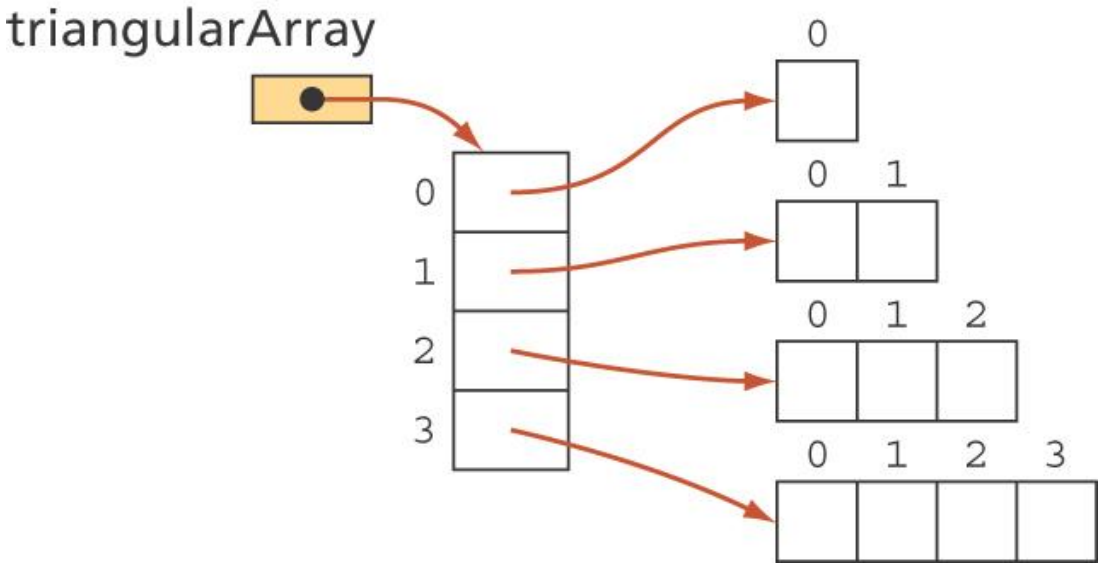
```
payScaleTable = new double [4][ ];

payScaleTable[0] = new double [5];
payScaleTable[1] = new double [5];
payScaleTable[2] = new double [5];
payScaleTable[3] = new double [5];
```

Хоёр хэмжээст массив

- Дэд массивүүдийн хэмжээ өөр өөр байж болно.
- Ажиллуулах

```
triangularArray = new double[4][ ];
for (int i = 0; i < 4; i++)
  triangularArray[i] = new double [i + 1];
```

дээрхийн үр дүн дараах байдлаар харагдана:



Collection классууд: Жагсаалт List, Буулгалт Map ба Олонлог Set

- **java.util** стандарт пакет нь объектуудын олонлогтой ажилладаг төрөл бүрийн классуудтай.
- Эдгээр классуудыг нийтээр нь *Java Collection Framework (JCF)* гэнэ.

Жагсаалт List

- Шугаман жагсаалт хэлбэрээр объектуудыг хадгалахад зориулагдсан.

$$L = (l_0, l_1, l_2, \ldots, l_N)$$

- Өгөгдсөн жагсаалтад объект нэмэх, устгах, хандах боломжтой.
- Хэдэн ч объект нэмэх боломжтой.

Жагсаалт List

- Нийт 25 метод байдгаас 5-г жишээгээр үзүүлэв:

E нь классыг илэрхийлнэ.
E-н оронд өөрийн классыг бичнэ.

Метод	Тайлбар
boolean add(E o)	Жагсаалтад объект нэмэх
void clear ( )	Жагсаалтыг цэвэрлэх буюу хоослох
E get (int idx)	Idx байрлал дахь элементийг буцаах
boolean remove(int idx)	Idx байрлал дахь элементийг устгах
int size( )	Жагсаалтын хэмжээ буюу элементийн тоог буцаах

Жагсаалт ашиглах

- Програмдаа жагсаалт ашиглахын тулд List интерфейсийг хэрэгжүүлсэн классын инстансыг үүсгэх хэрэгтэй.
- **List** интерфейсийг хэрэгжүүлсэн дараах 2 класс байдаг:
  - ArrayList
  - LinkedList

- **ArrayList** класс нь өгөгдлийг удирдахад массив ашигладаг.
- **LinkedList** класс нь *холбоост-зангилаагаар илэрхийлэх (linked-node representation)* гэх техникийг ашигладаг.

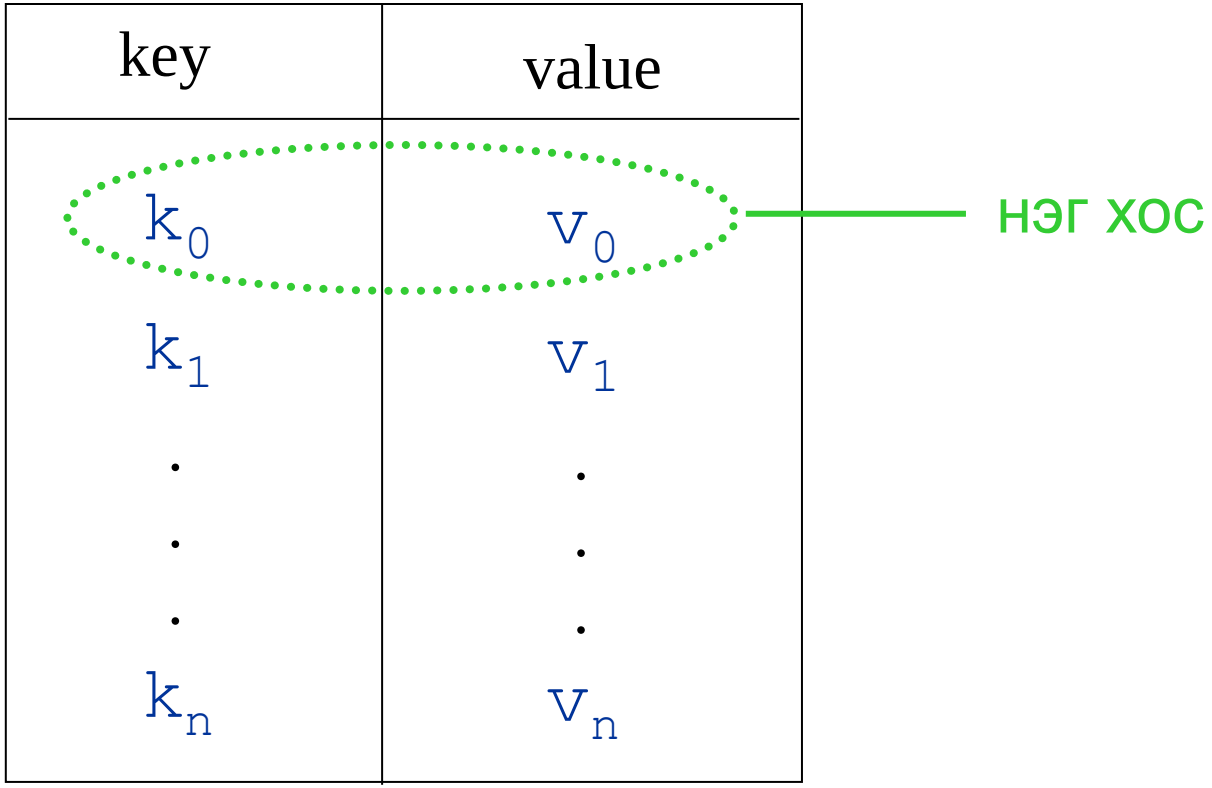
Жагсаалтын хэрэглээ

- Person объектын жагсаалттай ажиллах:

```
import java.util.*;

List<Person> friends;
Person person;

friends = new ArrayList<Person>( );

person = new Person("jane", 10, 'F');
friends.add( person );
person = new Person("jack", 6, 'M');
friends.add( person );

Person p = friends.get( 1 );
```

Буулгалт Map

- **Map** буюу буулгалт нь түлхүүр(key) ба утга(value) гэсэн хосын цуглуулга юм.



Буулгалт

- 14 метод байдгаас 5 нь:

Метод	Тайлбар
void clear ()	хоослох
boolean containsKey (Object key)	өгөгдсөн түлхүүр байгаа эсэхийг шалгах
V put (K key, V value)	түлхүүрт утга оноох, түлхүүр байхгүй бол хосыг нэмэх
V remove (Object key)	түлхүүр бүхий хосыг устгах
int size ( )	хэмжээ авах

Буулгалтын хэрэглээ

- Буулгалт ашиглахын тулд Map интерфейсийг хэрэгжүүлэгч классын тохиолдолыг үүсгэх хэрэгтэй.
- **Map** интерфейсийг хэрэгжүүлэгч классууд:
  - HashMap
  - TreeMap

Буулгалтын хэрэглээ - Жишээ

```
import java.util.*;

Map catalog;
catalog = new TreeMap<String, String>( );

catalog.put("CS101", "Intro Java Programming");
catalog.put("CS301", "Database Design");
catalog.put("CS413", "Software Design for Mobile Devices");

if (catalog.containsKey("CS101")) {
    System.out.println("We teach Java this semester");
} else {
    System.out.println("No Java courses this semester");
}
```

Олонлог Set

- Set – математик олонлогийг илэрхийлсэн загвар бөгөөд ижил элемент агуулахыг зөвшөөрдөггүй.

Метод	Тайлбар
add( )	олонлогт объект нэмэх
clear( )	олонлогоос объект устгах
contains( )	Өгөгдсөн объект олонлогийн элемент мөн бол үнэн утга буцаана.
isEmpty( )	олонлог хоосон үед үнэн утга буцаана.
iterator( )	Объектыг буцаахад ашиглагдах Iterator-г буцаана.
remove( )	Өгөгдсөн объектыг олонлогоос устгана.
size( )	олонлогийн элементийн тоог буцаана.

Дүгнэлт

- Массивт анхдагч өгөгдлийн төрөл бүхий өгөгдөл хадгалахаас гадна объект хадгалах боломжтой.
- Хүснэгтэн мэдээллийг дүрслэх бол 2 хэмжээст массивыг ашиглана.
- Хүснэгтийн хэмжээ тогтмол байдаг тул ихэсгэх боломжгүй. Харин жагсаалт нь хэдэн ч объект нэмэх боломжтойгоороо онцлог.
- Буулгалт нь түлхүүр, утга бүхий хослолыг агуулна.