



Lesson 3



18.12.2025

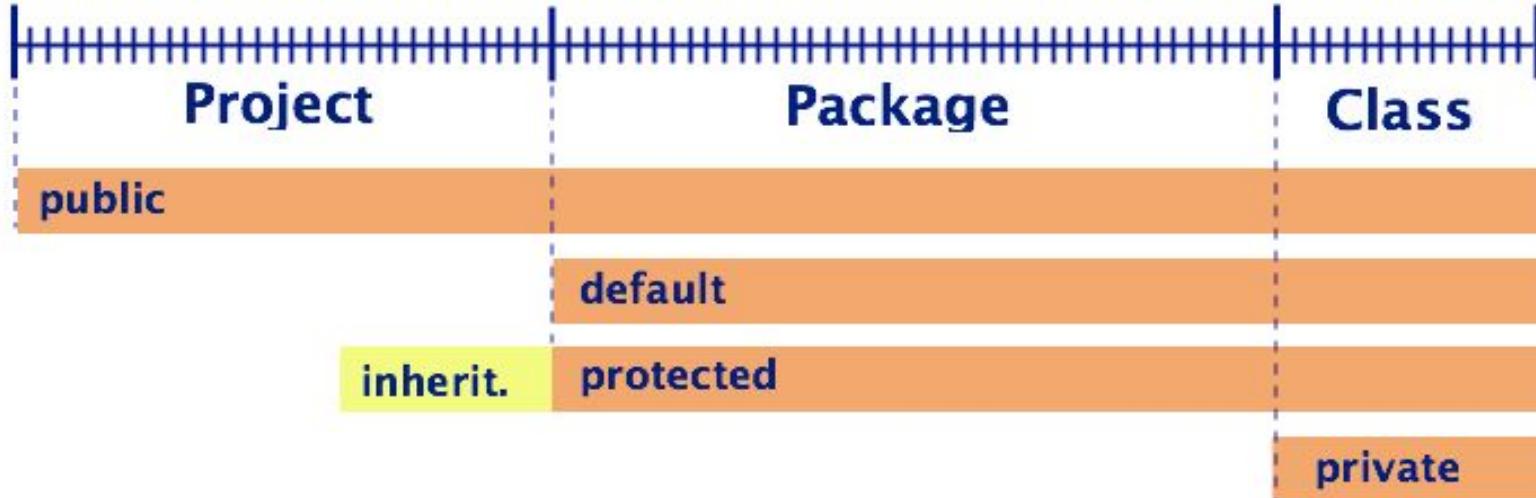
```
public class Task1 {  
    public static void main(String[] args) {  
        byte a = 100;  
        char b = 'A';  
        System.out.println(a + b);  
    }  
}
```

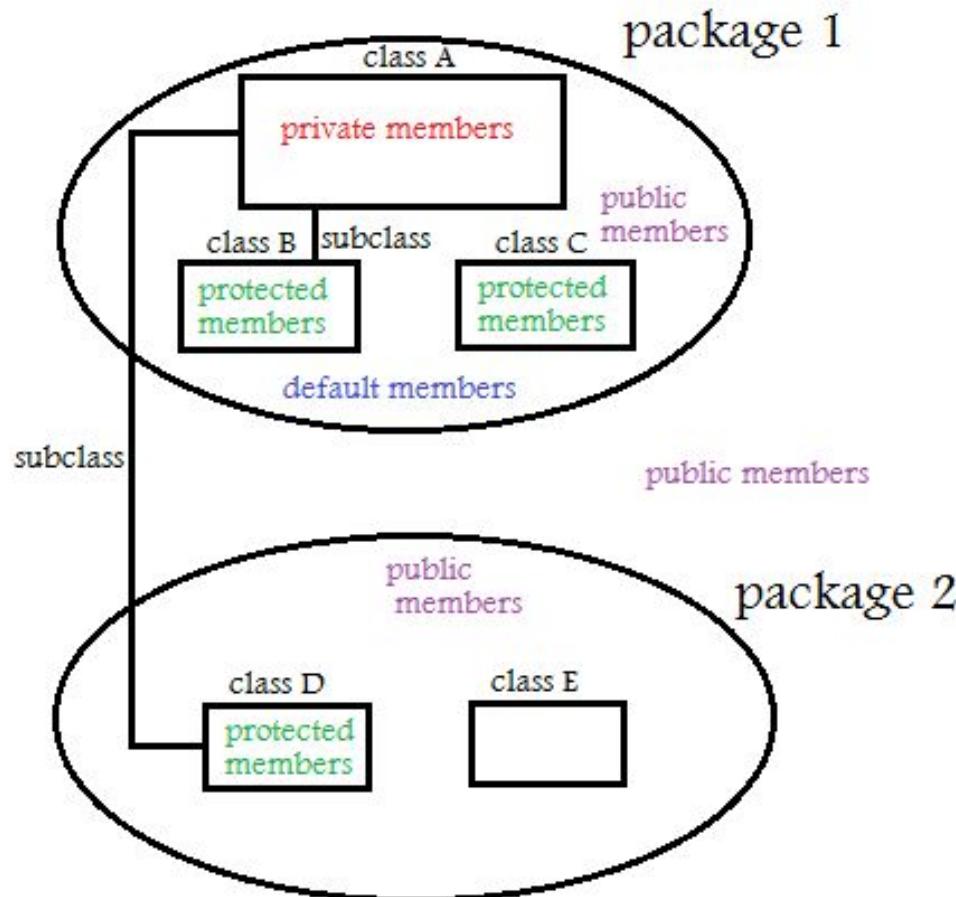
```
public class Task2 {  
    public static void main(String[] args) {  
        double a = 9.7;  
        int b = (int) a;  
        System.out.println(b);  
    }  
}
```

```
public class Task3 {  
    public static void main(String[] args) {  
        Integer a = 100;  
        int b = a;  
        System.out.println(b);  
    }  
}
```

```
public class Task4 {  
    public static void main(String[] args) {  
        Integer a = 127;  
        Integer b = 127;  
        System.out.println(a == b);  
  
        Integer c = 256;  
        Integer d = 256;  
        System.out.println(c == d);  
    }  
}
```

Модифікатори доступу





Перетворення типів

Implicit Type
Conversion

Explicit Type
Conversion

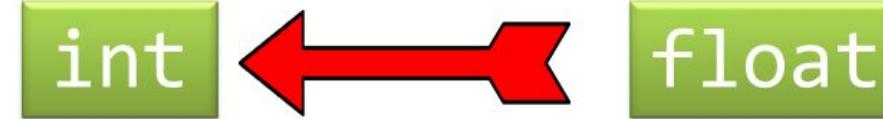
Numeric Type Conversions
Type Casting

Automatic Type Conversion
or
Auto Casting
or
Widening

byte -> short
short -> int
Etc.

Explicit Type Casting
or
Narrowing
or
Casting Incompatible Types

double -> float
float -> long
long -> int
int -> short
Etc.



```
int number;  
float fval= 32.33f;  
number= (int)fval;
```

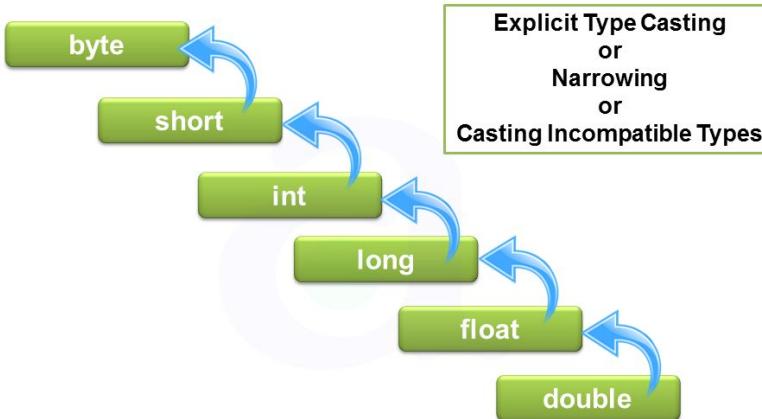
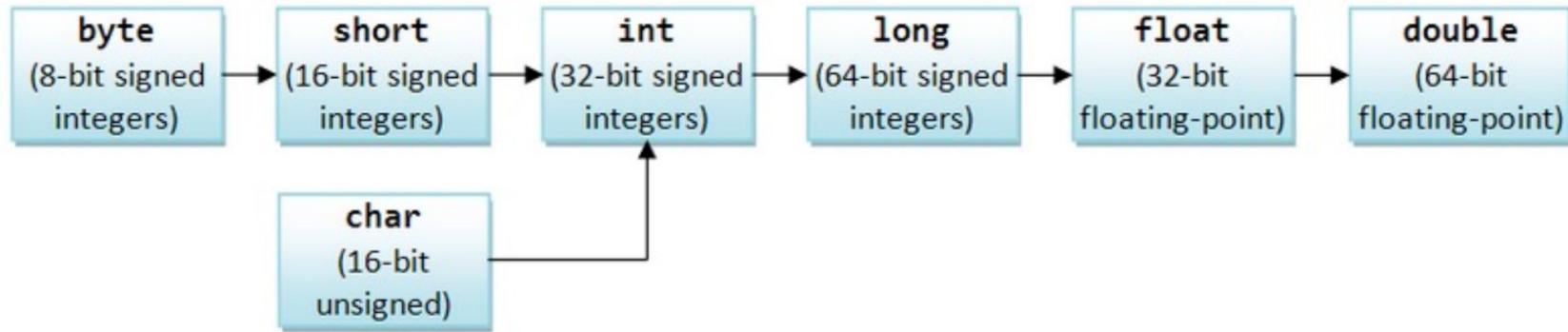
int

float

```
int number;  
float fval= 32.33f;  
number= (int)fval;
```

Type in which you
want to convert

Variable name
Which you want to convert

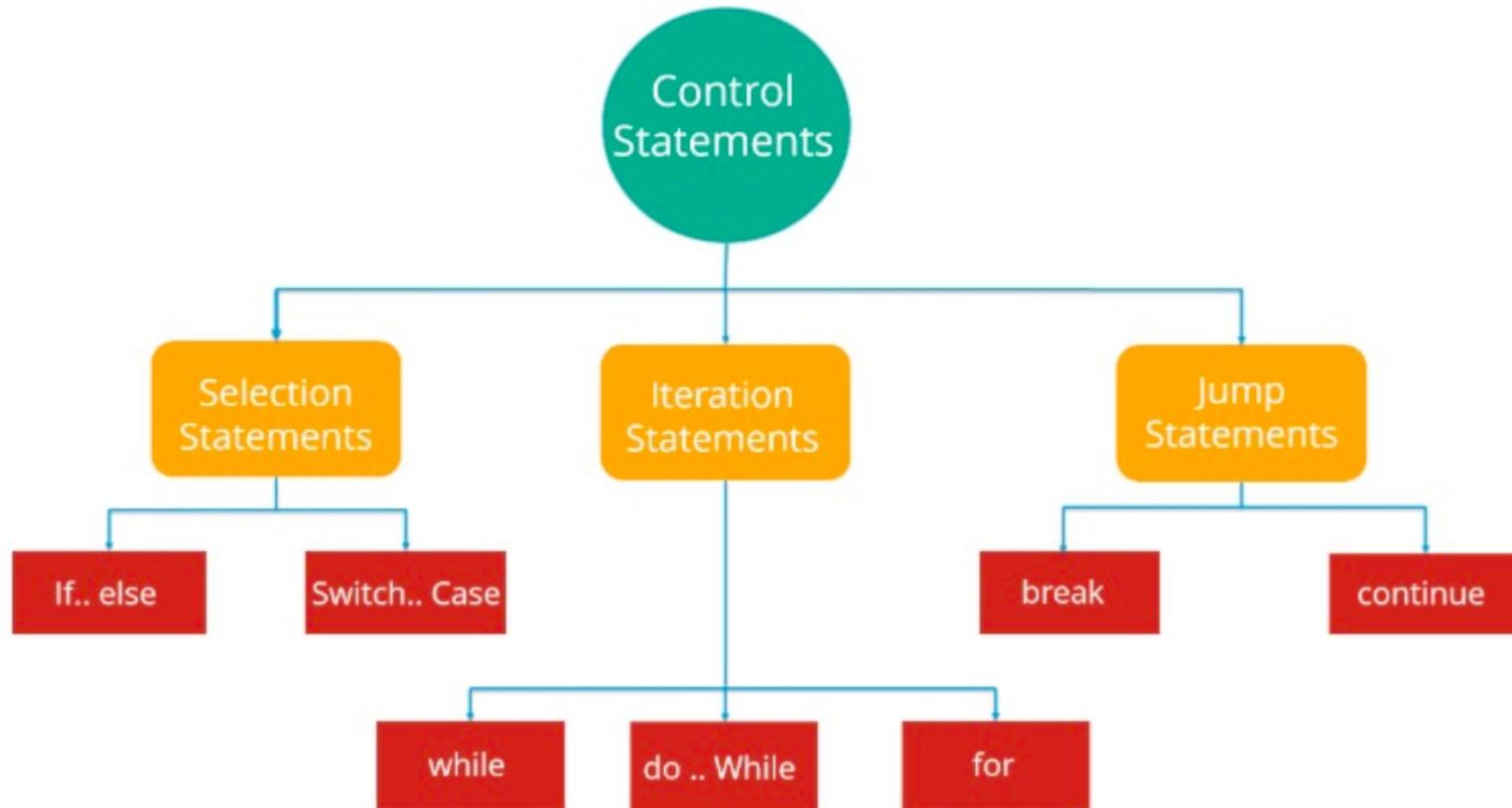


Неявне перетворення типів виконується у разі, якщо виконуються умови:

- Обидва типи сумісні
- Довжина цільового типу більша або дорівнює довжині вихідного типу

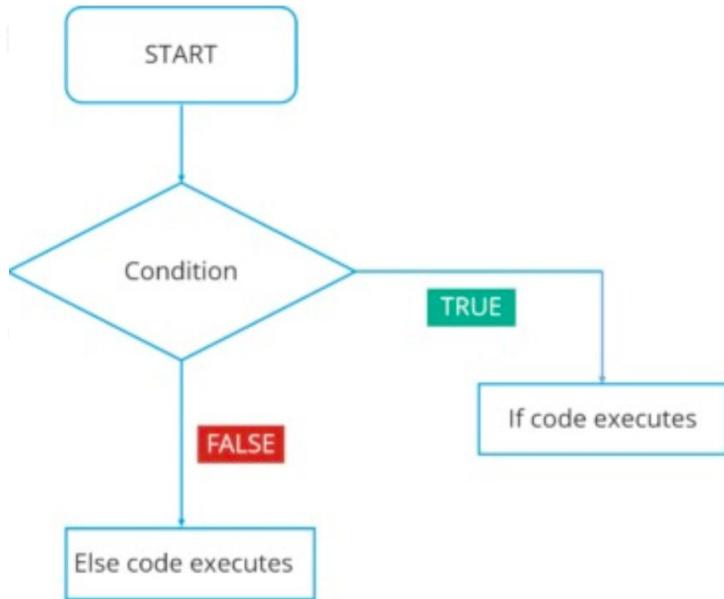
В інших випадках має використовуватися **явне** перетворення типів

Java control statements



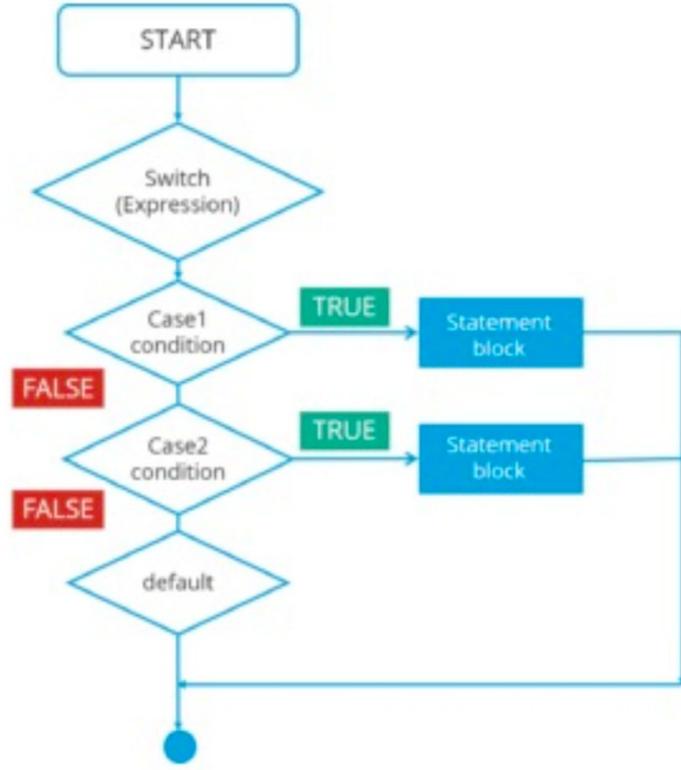


if-else statements



```
1 public class Compare {  
2     int a=10,  
3         b=5;  
4  
5     if(a>b)  
6         { // if condition  
7             System.out.println(" A is greater than B");  
8         }  
9     else  
10        { // else condition  
11            System.out.println(" B is greater");  
12        }  
13 }
```

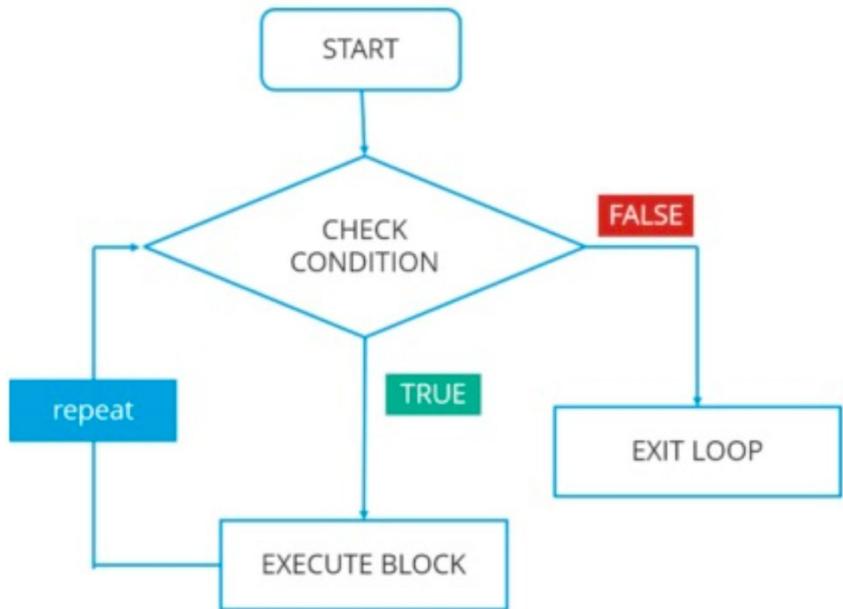
Switch case



```
1 public class SwitchExample {  
2     int week=7;  
3     String weeknumber;  
4  
5     switch(week){ // switch case  
6         case 1:  
7             weeknumber="Monday";  
8             break;  
9  
10        case2:  
11            weeknumber="tuesday";  
12            break;  
13  
14        case3:  
15            weeknumber="wednesday";  
16            break;  
17  
18        default: // default case  
19            weeknumber="invalid week";  
20            break;  
21     }  
22     System.out.println(weeknumber);  
23 }  
24 }
```

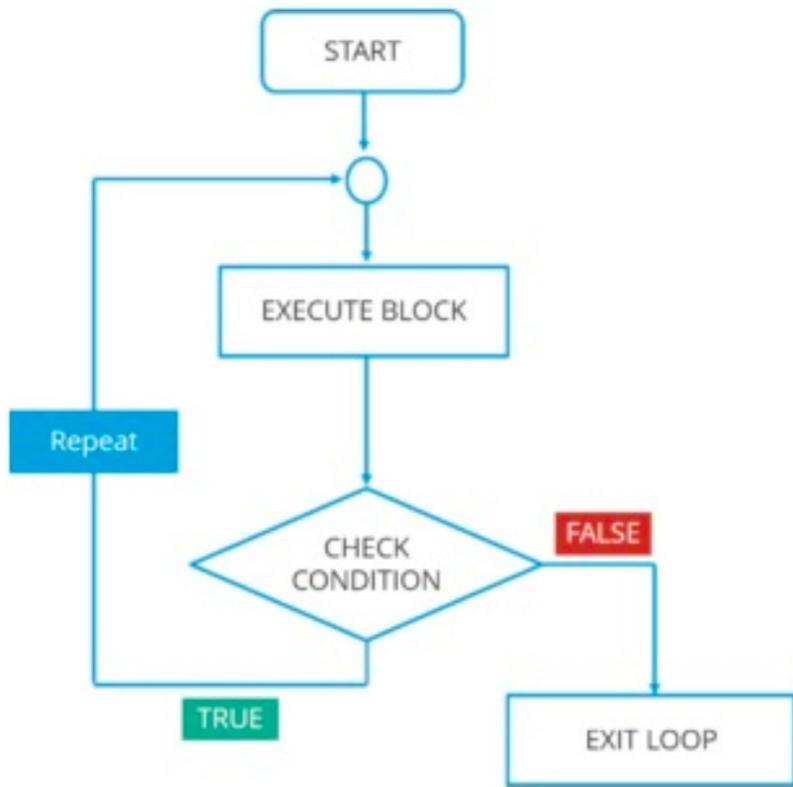


While statement



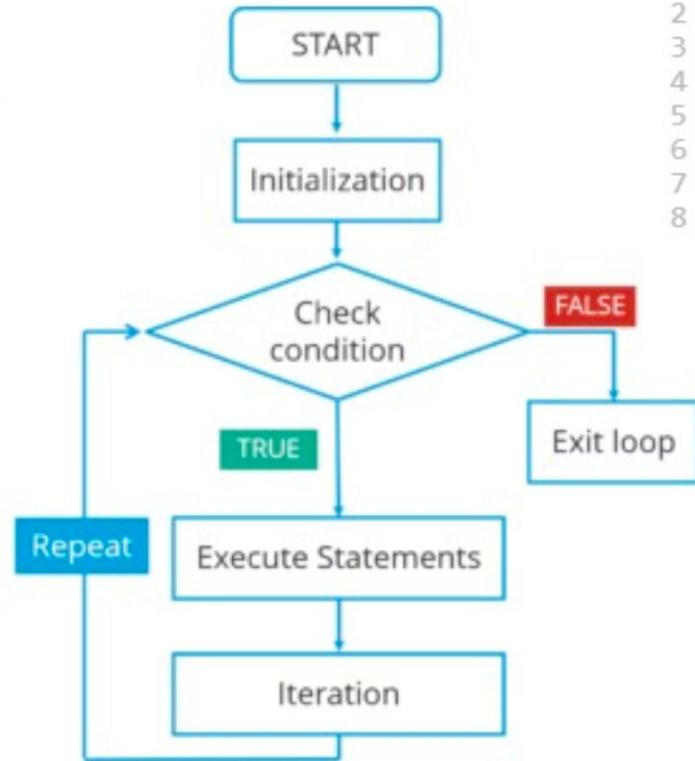
```
1 public class WhileExample {  
2     public static void main(String args[]) {  
3         int a=5;  
4         while(a<10) //while condition  
5         {  
6             System.out.println("value of a" +a);  
7             a++;  
8             System.out.println(""  
9         );  
10        }  
11    }  
12 }
```

Do-while statement:



```
1 public class DoWhileExample {  
2     public static void main(string args[]){  
3         int count=1;  
4     do {  
5         System.out.println("count is:"+count);  
6         count++;  
7     }  
8     while (count<10)  
9     }  
10 }
```

For statement



```
1 public class ForExample {  
2     public static void main(String args[]) {  
3         for(int i=0; i<=10; i++) // for condition  
4         {  
5             System.out.println(i);  
6         }  
7     }  
8 }
```

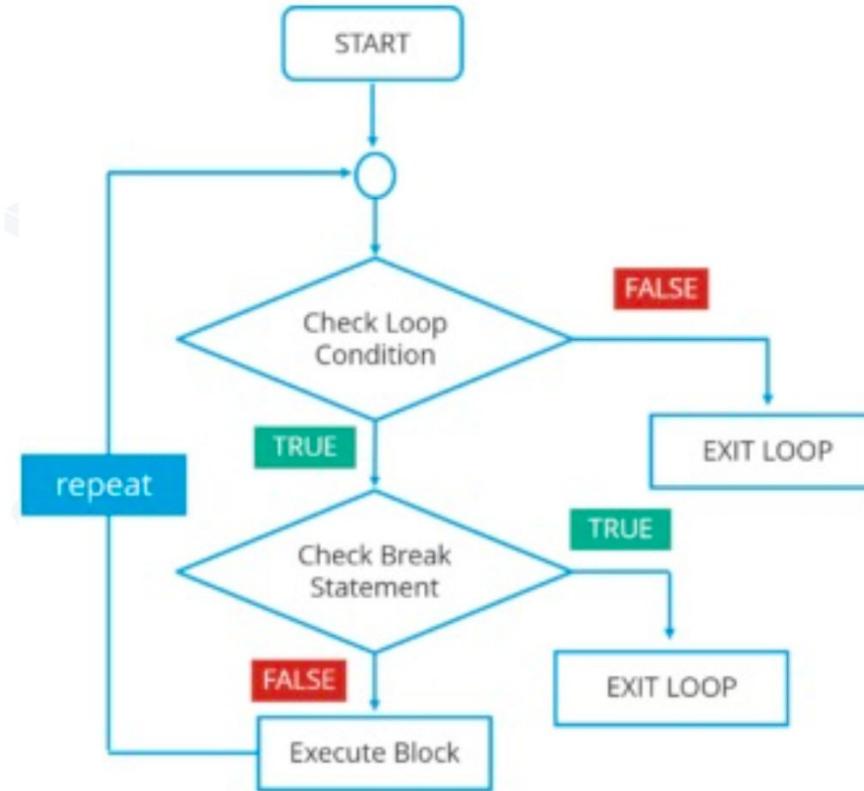
Foreach

- это разновидность цикла for
- используется для перебора элементов массива или коллекции

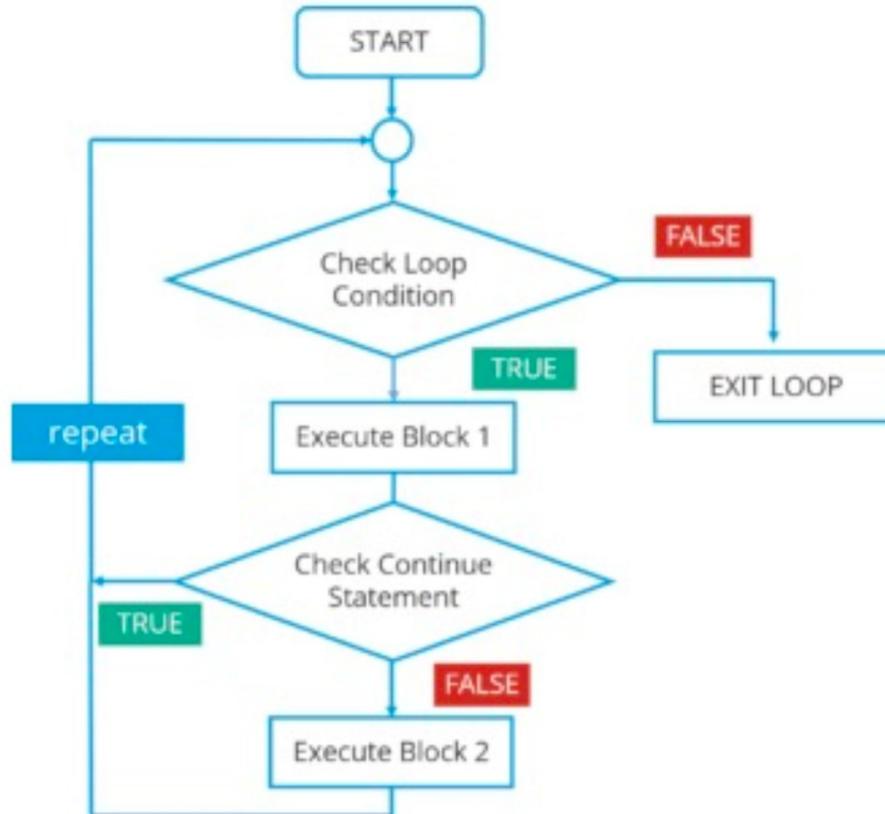
```
1 class Test {  
2  
3     public static void main(String[] args) {  
4         int[] array = {51,136, 387};  
5  
6         for (int i = 0; i < array.length; i++)  
7             System.out.println(array[i]);  
8     }  
9 }  
10 }
```

```
1 class Test {  
2  
3     public static void main(String[] args) {  
4         int[] array = {51,136,387};  
5  
6         for (int i:array) {  
7             System.out.println(i);  
8         }  
9     }  
10 }
```

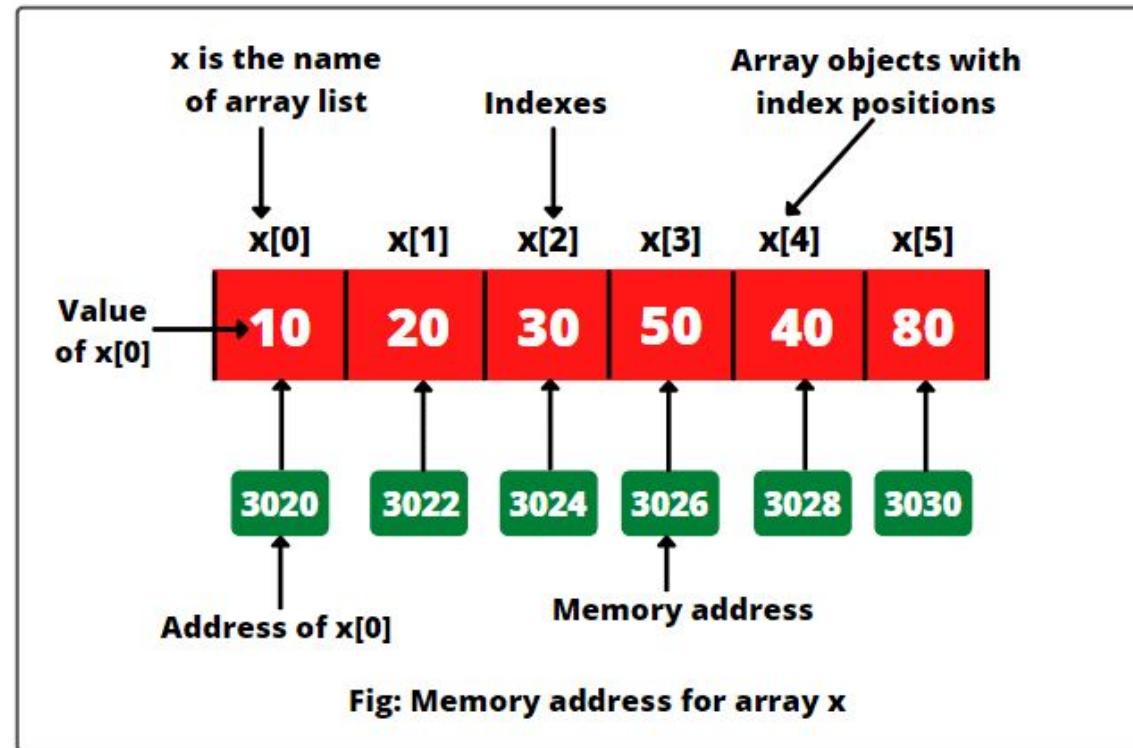
Break statement



Continue statement

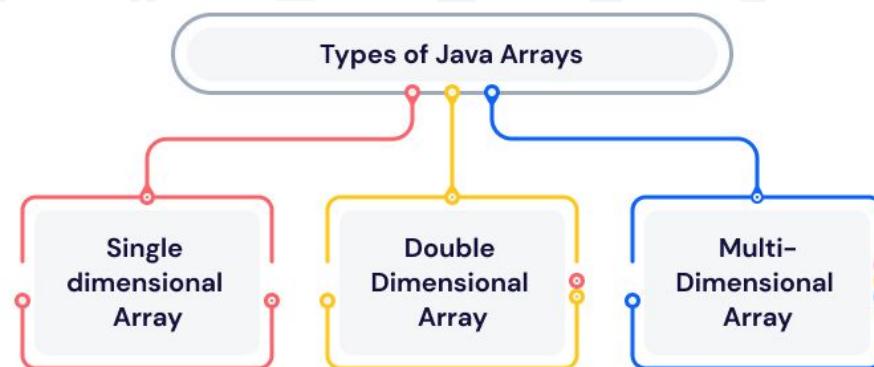


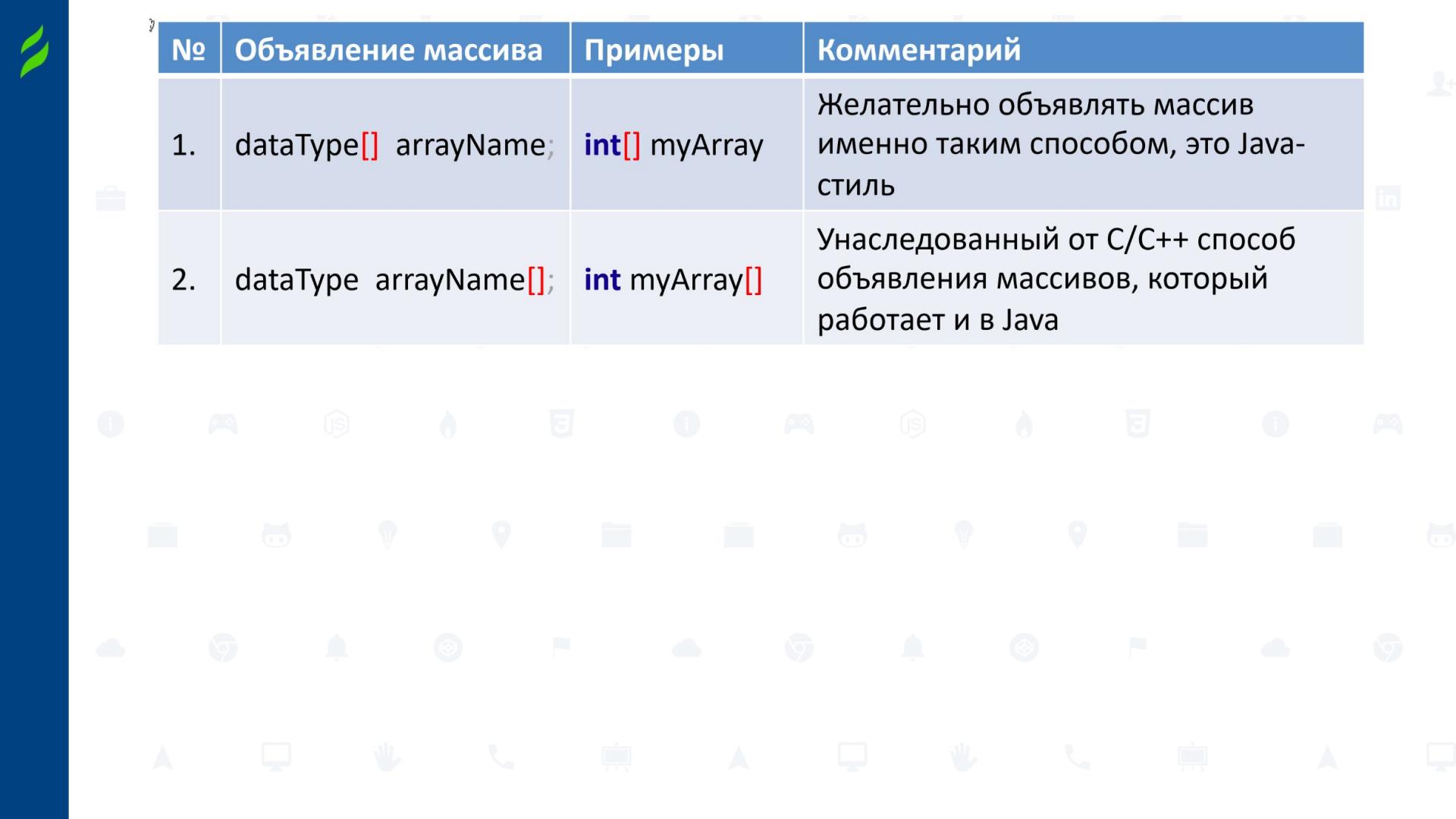
Масив – це структура даних, у якій зберігаються елементи одного типу. Його можна уявити, як набір пронумерованих осередків, у кожну з яких можна помістити якісь дані (один елемент даних в одну комірку). Доступ до конкретного осередку здійснюється через її номер. Номер елементу в масиві також називають індексом



Нижче наведено деякі важливі моменти щодо масивів Java.

- Масиви в Java є об'єктами.
- У Java всі масиви розподіляються динамічно.
- Масив Java також можна використовувати як статичне поле, локальну змінну або параметр методу.
- Розмір масиву має бути задано значенням int, а не long або short.
- Прямим супер класом типу масиву є Object.
- Масив може містити примітивні типи даних, а також об'єкти класу залежно від визначення масиву.
- У разі примітивних типів даних фактичні значення зберігаються в безперервних розташуваннях пам'яті.
- У випадку об'єктів класу фактичні об'єкти зберігаються в сегменті heap.





№	Объявление массива	Примеры	Комментарий
1.	dataType[] arrayName;	int[] myArray	Желательно объявлять массив именно таким способом, это Java-стиль
2.	dataType arrayName[];	int myArray[]	Унаследованный от C/C++ способ объявления массивов, который работает и в Java