



unsigned short	16-bit	short	
long	32-bit	int	32-bit
unsigned long	32-bit	int	32-bit
long long	64-bit	long	64-bit
unsigned long long	64-bit	long	64-bit
float	single-precision floating number	float	
double	double-precision floating number	double	

### IDL Module

(module) IDL . IDL package

```

module com{
    module mcp{
        module library{
        };
    };
};

```

```
package com.mcp.library;
```

### IDL Interface

가 public 가 . 가  
 . IDL  
 Helper Holder 가

```

module com{
    module mcp{
        module library{
            interface catalog{
            };
        };
    };
};

```

```

package com.mcp.library;
public interface catalog extends org.omg.CORBA.Object {
}

```

▶ □

protected, public 가 가 , IDL private,  
 public .

- Attribute : . IDL

- : readonly IDL 가 accessor/get

```

module com{
    module mcp{
        module library{
            interface catalog{

                attribute string name;
                attribute string address;
                readonly attribute short age;

            };
        };
    };
};

```

```

package com.mcp.library;
public interface catalog extends org.omg.CORBA.Object {

    public void name(java.lang.String name);
    public java.lang.String name();
    public void address(java.lang.String address);
    public java.lang.String address();
    public short age();
}

```

**struct**

```

IDL
helper 가 final holder,
public final
.
.
Sample
가 Point final Sample1Packeage
. Sample.Sample 1Package Sample.Sample 1Package
가 가 IDL Holder Helper
, string

```

```

//IDL
module Sample
{
    interface Sample 1
    {
        struct Point
        {
            double x;
            double y;
        };
    };
};

```

```

//Point.java
package Sample.Sample 1Package;

```

```

final public class Point {
    public double x;
    public double y;
    public Point() {
    }
    public Point(
        double x,
        double y
    ) {
        this.x = x;
        this.y = y;
    }
    public java.lang.String toString() {
        org.omg.CORBA.Any any = org.omg.CORBA.ORB.init().create_any();
        Sample.Sample2Package.PointHelper.insert(any, this);
        return any.toString();
    }
}

```

## Enumeration

```

final          enum          . enum
가          ,          ( _ )          int
,
.

```

```

//IDL
enum Grade {A, B, C, D};

//Grade.java
package ComplexType;

public class Grade {
    final public static int _A = 0;
    final public static int _B = 1;
    final public static int _C = 2;
    final public static int _D = 3;
    final public static ComplexType.Grade A = new ComplexType.Grade(_A);
    final public static ComplexType.Grade B = new ComplexType.Grade(_B);
    final public static ComplexType.Grade C = new ComplexType.Grade(_C);
    final public static ComplexType.Grade D = new ComplexType.Grade(_D);
    private int __value;
    protected Grade(int value) {
        this.__value = value;
    }
    public int value() {
        return __value;
    }
    public static ComplexType.Grade from_int(int $value) {
        switch($value) {
            case _A:
                return A;
            case _B:
                return B;
            case _C:
                return C;
            case _D:
                return D;
            default:

```

```

        throw new org.omg.CORBA.BAD_PARAM("Enum out of range: [0.." + (4 - 1) + "]: " + $value);
    }
}
public java.lang.String toString() {
    org.omg.CORBA.Any any = org.omg.CORBA.ORB.init().create_any();
    ComplexType.GradeHelper.insert(any, this);
    return any.toString();
}
}

```

## Union

### Template

```

IDL      string      sequence      가
    * string
    string      IDL      가      complex      .. string
    chars
    . IDL      bound      unbound string      ,      string
    unbound
module com{
    module mcp{
        module library{
            interface catalog{
                typedef string<20> aName;
                struct name{
                    string first;
                    string middle;
                    string last;
                };
            };
        };
    };
};

```

```

package com.mcp.library.catalogPackage;
final public class name {
    public java.lang.String first;
    public java.lang.String middle;
    public java.lang.String last;

    public name() {
    }
    public name (
        public java.lang.String first,
        public java.lang.String middle,
        public java.lang.String last
    ) {
        this.first = first;
        this.middle = middle;
        this.last = last;
    }
}
가 typedef

```

가 . bound unbound  
string 가 가

◆ sequence  
sequence . string , sequence bound  
unbound 가 . bound sequence 가  
, unbound

```

module com{
    module mcp{
        module library{
            interface catalog{

                typedef string<20> aName;

                struct name{
                    string first;
                    string middle;
                    string last;
                };

                struct register {
                    sequence<name, 10> somemembers;
                    sequence<name> allmembers;
                };
            };
        };
    };
};

```

package com.mcp.library.catalogPackage;

```

final public class register {
    public com.mcp.library.catalogPackage.name[] somemembers;
    public com.mcp.library.catalogPackage.name[] allmembers;
    public register () {
    }
    public register(
        com.mcp.library.catalogPackage.name[] somemembers,
        com.mcp.library.catalogPackage.name[] allmembers,
    ) {
        this.somemembers = somemembers;
        this.allmembers = allmembers;
    }
}

```

sequence IDL struct  
. sequence IDL ,  
interface, struct, basic

◆ Array  
sequenc , array

IOP (collection)

IDL sequence  
module com{



```

        };

        typedef name theNames[10];

        struct theWhiteBook{
            theNames allTheName;
            string year;
        };

        register getRegister();
        theWhiteBook getWhiteBook();
};
};
};
};

```

*\_example\_catalog.java*

```

package com.mcp.library;
import com.mcp.library.catalogPackage.*;

public class example_catalog extends com.mcp.library._catalogImplBase {
    public example_catalog(String serverName) {
        super(serverName);
    }
    public register getRegister() {
    }
    public theWhiteBook getWhiteBook() {
    }
}

```

```

getRegister()      register      . register
                  .              array
                  somemembers  10      sequence
                  .              sequence
                  allmembers
theWhiteBook      name      array      . theWhiteBook
                  register      . bound sequence      array
                  . bound sequence      array
                  ,              , out      inout
getRegister()
                  .
                  sequenceType.somemembers  5
                  가              , 5              sequence 가
                  . sequenceType.allmembers  11
bounded sequence      10              5
                  가              . array      10
sequence 가              가              array
                  ,              가              가              , array
                  가

```

<< source .....>>

◆ sequencArray

.....  
<< Source .....>>

- ◆ [sequenceArray](#)  
.....

**Typedef**      **Constant**  
 typedef      가      typedef  
 IDL      typedef

<< source 154 >>

typedef IDL long int const final static  
 int

**IDL**

-      가      void  
 -      가      ,

Void operation\_name (in type name, out type name, inout type naem);  
 Atype operations\_name(in type name, out type name, inout type name);

-      in  
 -      out  
 -      inout

- IDL in      pass-by-value      가  
 , in  
 - IDL out inout      IDL out      inout  
 IDL-Java

Holder      가 IDL pass-by-reference  
 IDL      Holder  
 가      Holder

**Holder**

IDL      in, out, inout

SequenceArray

```
Module com{
  Module mcp{
    Module library{
      Interface params{
        Struct parameter {
          String name;
          Long version;
        };

        long primitiveParams(in long anumber,
          out string astring,
          inout string anotherstring
        );
        parameter userParams(in parameter inparam,
          out parameter outparam,
          inout parameter inoutparam
        );
      };
    };
  };
};
```

package com.mcp.library;

```

public interface params extends org.omg.CORBA.Object {
    public int primitiveParams(
        int anumber,
        org.omg.CORBA.StringHolder astring,
        org.omg.CORBA.StringHolder anotherstring
    );
    public com.mcp.library.paramsPackage.parameter
    userParams(
        com.mcp.library.paramsPackage.parameter inparam,
        com.mcp.library.paramsPackage.parameterHolder outparam,
        com.mcp.library.paramsPackage.parameterHolder inputparam
    );
}

```

Native IDL 가 Holder 가 . OMG

```

Public int primitiveParams(
    Int anumber,
    Org.omg.CORBA.StringHolder astring,
    Org.omg.CORBA.StringHolder anotherstring
);
    anumber . astring StringHolder
    . Holdr . Holder 가
    anotherstring . Holder

```

**Constructed IDL** 가 ( )

- ▶ □ params ...
- ▶ □ params

**IDL Exception**

```

가 가 .
    가
exception 가
▶ □ CORBA System Exception
ORB 가 exception
exceptiond org.omg.CORBA.SystemException 가 final
    java.lang.RuntimeException
exception
▶ □ Exception
CORBA 가 exception
exception 가
exception 가
exception final
org.omg.CORBA.UserException , java.lang.Exception
exception IDL
exception , Java exception

```

IDL Contract final  
exception

```

module com{
  module mcp{
    module library{
      interface Contract{
        const short MaleLifeExpectancy = 76;
        exception violation {string condition; short code;};
        void checkInvariant (in short age) raises (violation);
      };
    };
  };
};

```

```

public void checkInvariant(short age)
throws com.mcp.library.ContractPackage.violation {
  if (age < MaleLifeExpectancy) {
    //Implement the use case...
  }
  else
  throw violation("Expectancy exceed", 1001);
}

```

```

try {
  contractRef.checkInvariant(77)
}
catch(ContractPackage.violation v) {
  System.out.println(e.toString());
}
catch(CORBA.SystemException e) {
  e.printStackTrace();
}

```

IDL-Java          OMG          CORBA 가          가  
pass-by-reference

IDL 가  
IDL module          package          IDL interface          Java interface  
CORBA          IDL  
final

IDL 가  
pass-by-reference          가          Holder  
out, inout          IDL exception  
exceptin          Holder  
가