

Using just XML create an XML specification of the IDL representation of an interface for a banking system.

```
module ACME-Bank
{
    interface    Bank
    {
        void    open( in String name );
        Accounts    getAccounts( in integer branched );
        oneway    void    processNewAccounts();
    };

    interface    Account
    {
        void        closeAccount();
        void        debit( in double amount );
        void        credit( in double amount );
        double      queryBalance( out double odLimit );
    }
};
```

A module is way of grouping common interfaces. It helps reduce the risk of interface name clashes between different groups.

Notice each parameter is marked with mode {in, out, inout}

- **in**; the parameter must be passed in
- **out**; expect a parameter back
- **inout**; a parameter must be passed in and one shall be passed back

Each operation must specify a result/return type

- void; expect nothing back
- short; number $-2^{15} - 2^{15}$
- long; number $-2^{31} - 2^{31}$
- float; floating point number 32 bit IEEE single precision
- double; floating point number 64 bit IEEE single precision
- long double; floating point number 128 bit IEEE single precision
- boolean; a value of TRUE or FALSE
- char; signed byte
- string; signed sequence of bytes
- octet; un-interpreted byte used for network protocols

This list is not exhaustive, but give you an idea of the type of things you be thinking about.

A oneway operation is an asynchronous operation that cannot return a result.