valtech_

Récapitulatif UML / Java

Diagrammes de classe 1/3

Class

```
A
```

```
B {abstract}
```

```
class A { }
```

```
abstract class B { }
```

Attributes

```
+ : public
- : private
# : protected
$ : static
{const} : final
```

```
A
+ att0:int
+ att1:int = 1
- att2:int = 1
# att3:int = 1
+ CONSTANT:int = 12 {const}
- array:int[20];
-$ st:int = 1
```

```
class A {
  public int att0;
  public int att1 = 1;
  private int att2 = 1;
  protected int att3 = 1;
  public static final int CONSTANT = 12;
  private int array[] = new int[20];
  private static int st = 1;   ...}
```

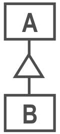
Method

```
{abstract} : abstract
```

```
A
+ m1(p:int):int
- m2(p:int):int
# m3(p:int):int
+ m4(p:int):int throws E;
+ m5(p:int):int {abstract}
+$ m6(p:int):int
```

```
class A {
  public int m1(int p) {...}
  private int m2(int p) {...}
  protected int m3(int p) {...}
  public int m4(int p) throws E {...}
  public abstract int m5(int p);
  public static int  m6(int p) {...}   ... }
```

Classification
Inheritance

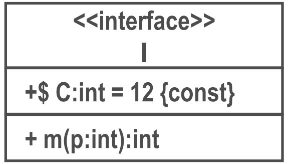


B is a kind of A

```
class B extends A {
}
```

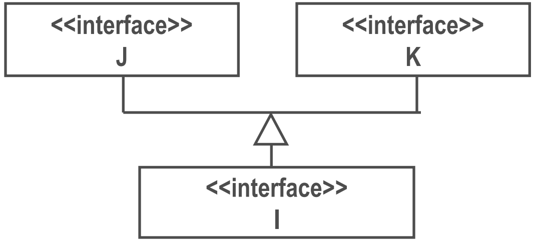
Diagrammes de classe 2/3

Interface



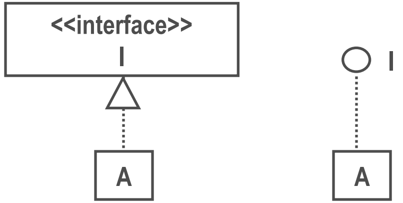
```
interface I {
  static final int C = 12;
  public int m(int p) {
   ...
  }
}
```

Interface inheritance



```
interface I extends J, K {
}
```

Interface implementation

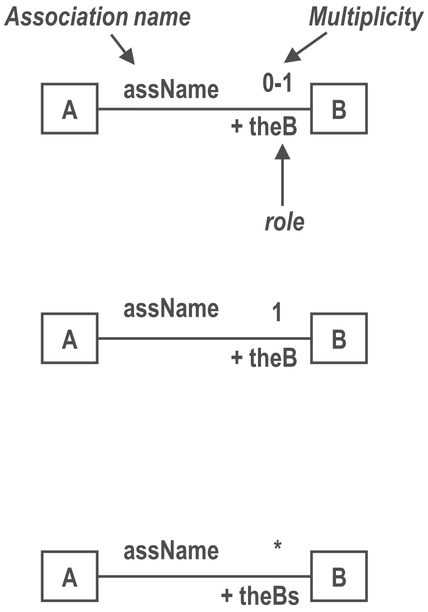


```
class A implements I {
}
```

valtech_

Diagramme de classe 3/3

Association



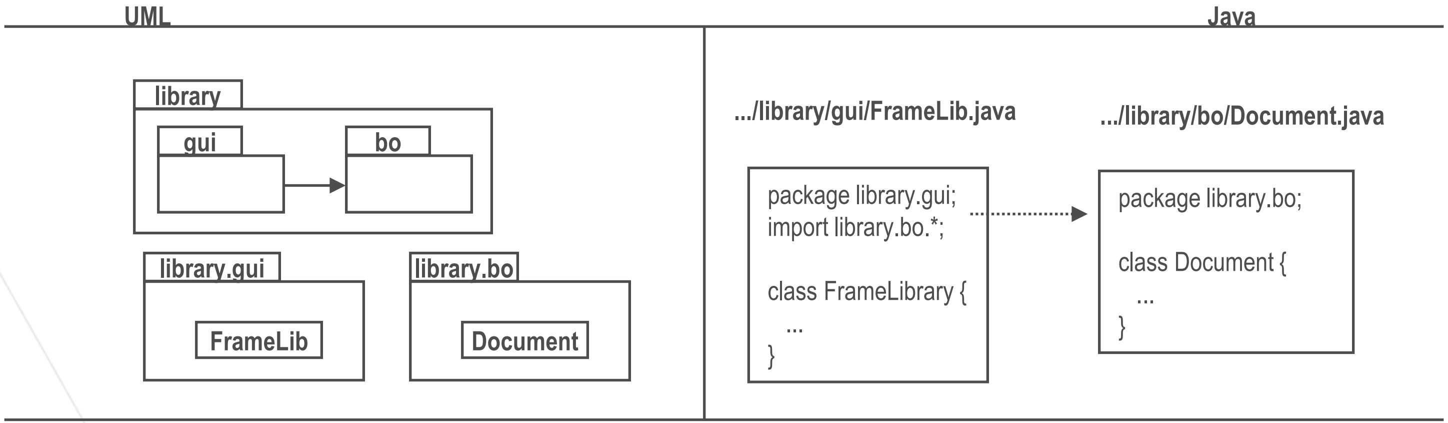
Multiplicity
1
0..1
*0..**
*

```
class A {
  private B _theB;
  public A() {...}
  public B getTheB() {...}
  public void setTheB(B theB) {...}
}
```

```
class A {
  private B _theB;
  public A(B theB) { _theB = theB; }
  public B getTheB() {...}
  public void setTheB(B theB) {...}
}
```

```
class A {
  private List _theBs = new ArrayList();
  public void addB(B b) { _theBs.add(b); }
  public void remB(B b) { _theBs.remove(b); }
  public Iterator theBs() { return _theBs.iterator(); }
}
```

Diagrammes de package



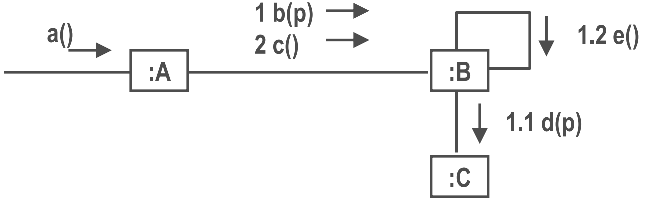
.../library/gui/FrameLib.java

```java
package library.gui;
import library.bo.*;

class FrameLibrary {
   ...
}
```

.../library/bo/Document.java

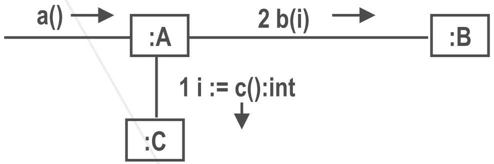
```java
package library.bo;

class Document {
   ...
}
```

Diagramme de communication 1/2



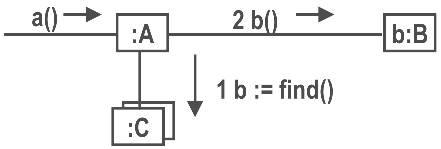
```
class A {
  public void a() {
       B  aB = .....;    // A has a reference on B
       aB.b();
       aB.c();  }
}
```

```
class B {
  public void b(int p) {
       C  aC = .....;    // B has a reference on C
       aC.d(p);
       e();   }
  public void e() {...}
  public void c() {...} }
```

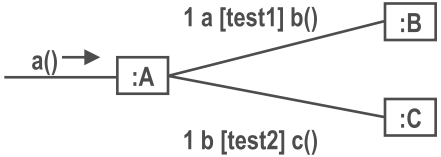


```
class A {
  public void a() {
    C aC = ...;
    B aB = ...;
    int i = aC.c();
    aB.b(i);   }
}
```

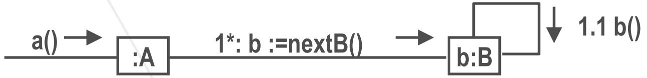
Diagramme de communication 2/2



```
class A {
  private List listOfC = new ArrayList();
  public void a() {
      Iterator cs = listOfC.iterator();
      B foundB = null;
      while(cs.hasNext()) {
          B aB = (B)cs.next();   // Select one object from the container }
      foundB.b();   } }
```



```
class A {
  public void a() {
       B  aB = .....;    // A has a reference on B
       C aC = ...; // A has a reference on C
       if( test1)
         aB.b();
       else if( test2 )
          aC.c();  } }
```



```
class A {
  private List listOfC = new ArrayList();
  public void a() {
      Iterator cs = listOfC.iterator();
      while(cs.hasNext()) {
          B aB = (B)cs.next();
          aB.b(); }  }  }
```

Récapitulatif : syntaxe Java

Class

```
public final class A extends B implements I, J {
  public static void main(String argv[]) {
    ...  } }
```

Types

Name	size unsigned	signed	Wrapper
byte	1	u	Byte
char	2	u	Character
short	2	s	Short
int	4	s	Integer
long	8	s	Long
float	4	s	Float
double	8	s	Double

Condition

```
if( i < 10 ) {
   ... }
```

```
int j = (i < 10) ? 12: 13;
```

```
switch(i) {
case 1:
  ...
  break;
case 2: case 3:
  ...
  break;
default:
  ...
  break
}
```

Operators

<table>
<tr><th>Arithmetic</th><th>Logic</th><th>Binary</th></tr>
<tr><td>+</td><td>!=</td><td>~</td></tr>
<tr><td>-</td><td>==</td><td>!</td></tr>
<tr><td>/</td><td><</td><td>&</td></tr>
<tr><td>*</td><td>></td><td>^</td></tr>
<tr><td>%</td><td><=</td><td>>></td></tr>
<tr><td>++</td><td>>=</td><td><<</td></tr>
<tr><td>--</td><td>||</td><td><<<</td></tr>
<tr><td></td><td>&&</td><td></td></tr>
<tr><td></td><td>instanceof</td><td></td></tr>
</table>

Interface

```
public final interface I extends J, K {
}
```

Loops

```
while(i < 0) {
   ...
}
```

```
do {
...
} while(i < 0);
```

```
for(int i = 0; i < 10; i++) {
}
```

Exception

```
public void m() throws E {
  if(...)
     throw new E();
}
```

```
public void l1() throws E {
   m();
}
```

```
public void l2() {
   try {
    m();
   } catch( E e1) {
     ...
     throw e1;
   } catch(Exception e2) {
   } finally { ...} }
```