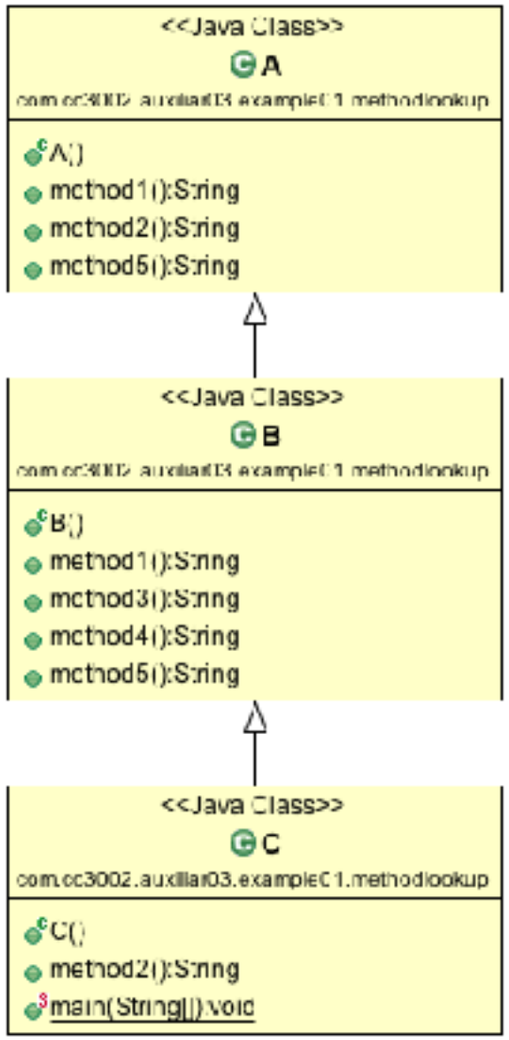
Metodologías de Diseño y Programación

Código: CC2003
Juraj Kubelka
Oficina: BP-P-2-14
Correo: juraj.kubelka@icloud.com

Method look-up, this, super

```java
public class A {
  public String method1() {
    return "A.method1()";
  }
  public String method2() {
    return "A.method2() > " + this.method1();
  }
  public String method5() {
    return "A.method5() > " + this.method2();
  }

public class B extends A {
  public String method1() {
    return "B.method1()";
  }
  public String method3() {
    return "B.method3() > " + super.method1();
  }
  public String method4() {
    return "B.method4() > " + super.method2();
  }
  public String method5() {
    return "B.method5() > " + super.method5();
  }
}

public class C extends B {
  public String method2() {
    return "C.method2() > " + this.method1();
  }
}
```




```java
  public static void main(String[] args) {
    System.out.println("1. " + new C().method1());
    System.out.println("2. " + new B().method1());
    System.out.println("3. " + new A().method1());
    System.out.println("4. " + new C().method2());
    System.out.println("5. " + new B().method2());
    System.out.println("6. " + new A().method2());
    System.out.println("7. " + new B().method3());
    System.out.println("8. " + new C().method4());
    System.out.println("9. " + new C().method5());
  }
```

Visibility

A, C, E

```java
public class A {
  private String method1() {
    return "A.method1()";
  }
  public String method2() {
    return "A.method2() > " + this.method1();
  }
}
```

B, D, F

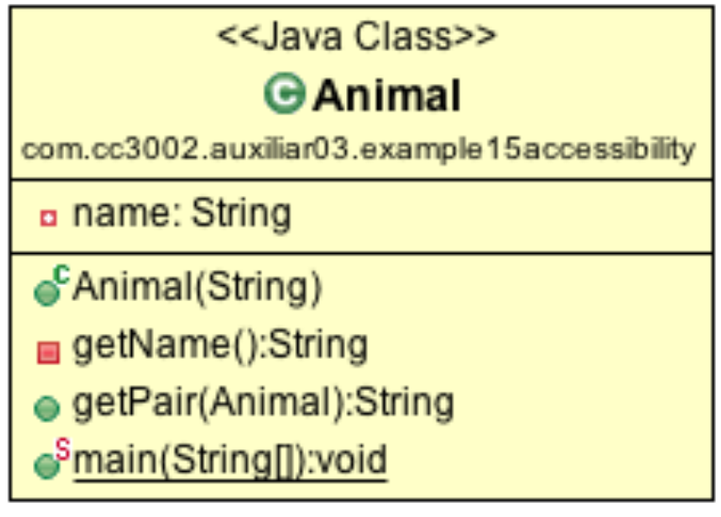
A, C, E

```java
public class B extends A {
  public String method1() {
    return "B.method1()";
  }
}
```

```java
public static void main(String[] args) {
  System.out.println("1. " + new A().method2());
  System.out.println("2. " + new B().method2());
  System.out.println("3. " + new C().method2());
  System.out.println("4. " + new D().method2());
  System.out.println("5. " + new E().method2());
  System.out.println("6. " + new F().method2());
}
```

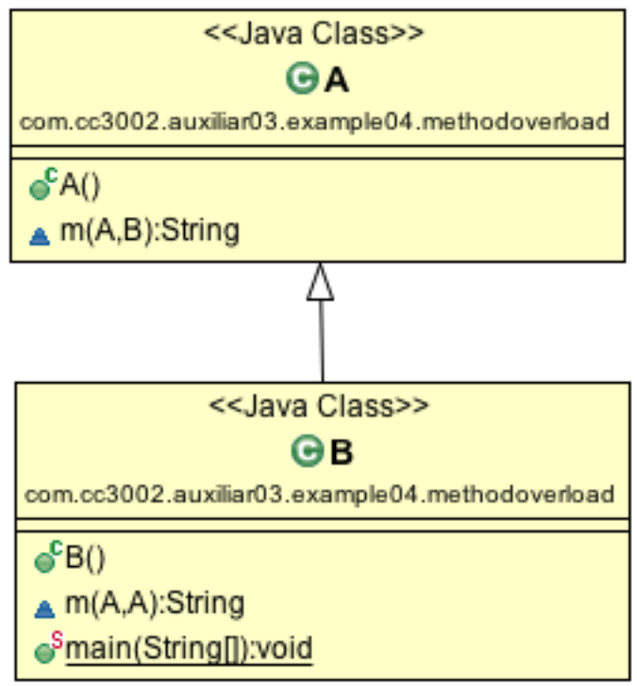
Accessibility

```java
public class Animal {
  private String name;
  public Animal(String name) {
    this.name = name;
  }
  private String getName() {
    return name;
  }
  public String getPair(final Animal paired) {
    return this.getName() + " with " + paired.getName();
  }



  public static void main(String[] args) {
    System.out.println("1. " + new Animal("Jirafa").getPair(new Animal("Antilope")));
    System.out.println("2. " + new Animal("Tigre").getName());
  }
}
```



Method over-load

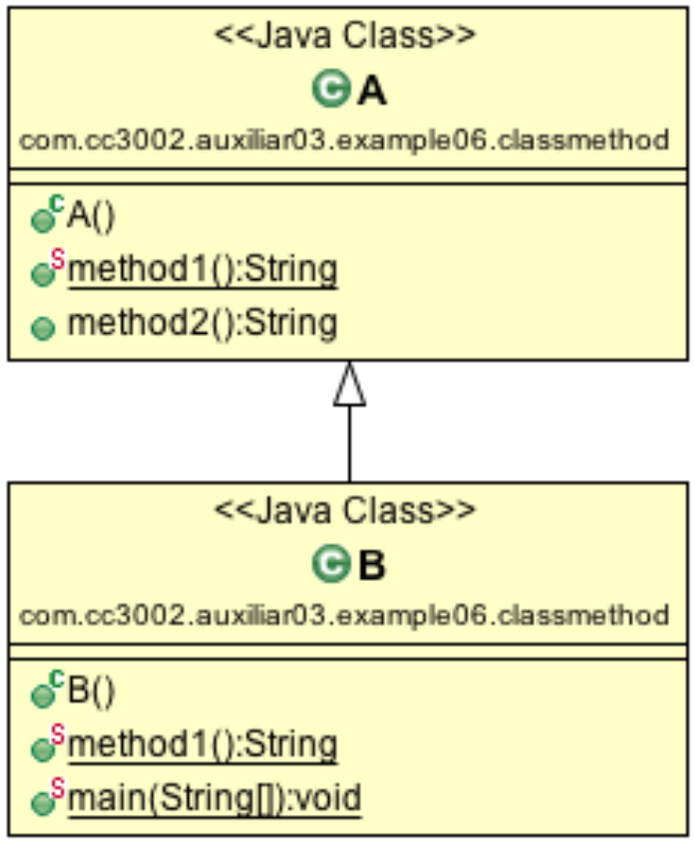
```java
class A {
    String m(A o1, B o2) {
        return "A.m(A,B)";
    }
}

public class B extends A {
    String m(A o1, A o2) {
        return "B.m(A,A)";
    }
}
```




```java
public static void main(String[] argv) {
  System.out.println("1. " + new B().m(new A(), new A()));
  System.out.println("2. " + new B().m(new A(), new B()));

  A object1 = new B();
  A object2 = new B();

  System.out.println("3. " + new B().m(object1, object2));
  System.out.println("4. " + new B().m((B) object1, object2));
  System.out.println("5. " + new B().m(object1, (B) object2));
}
```

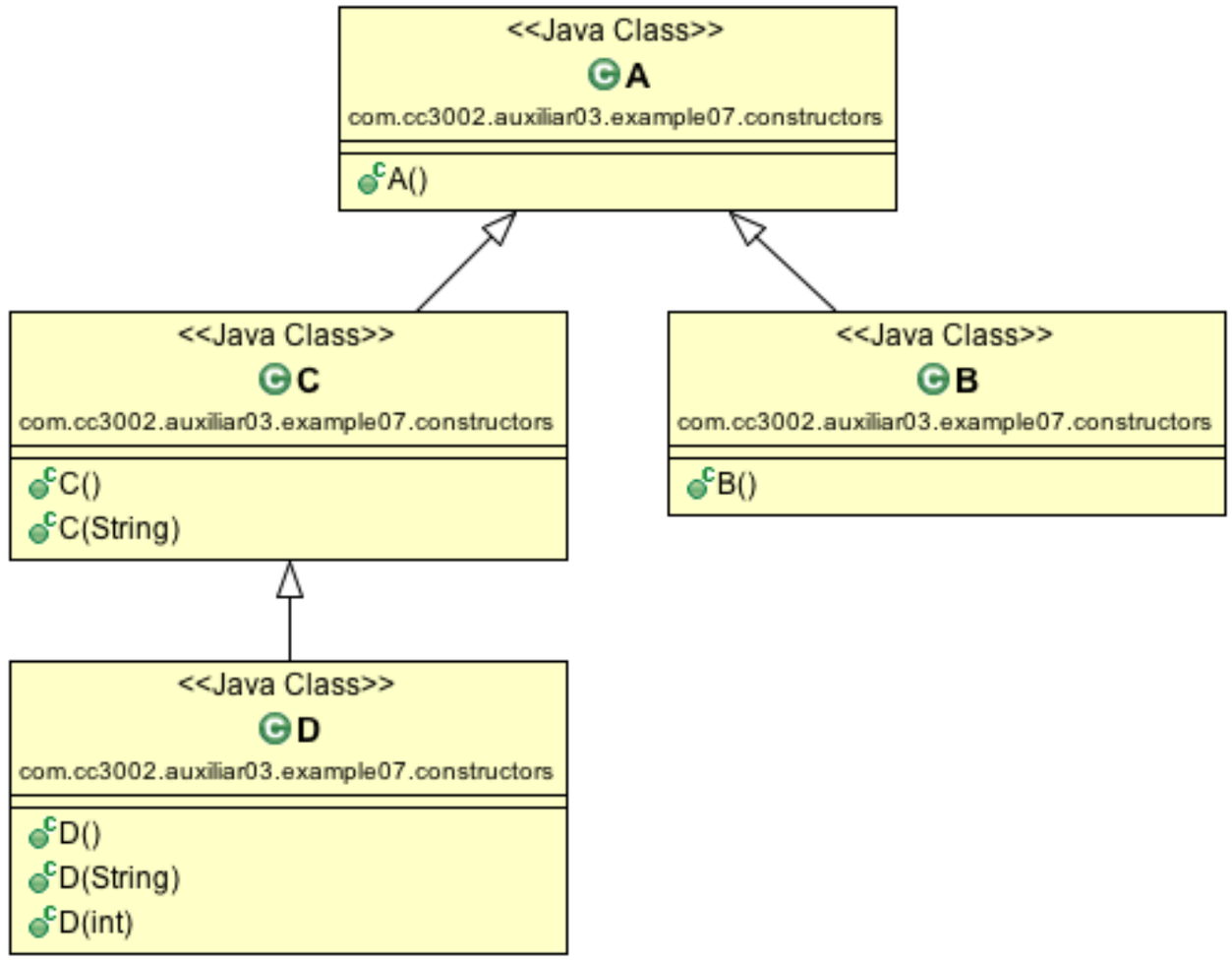
Static Methods

```java
public class A {
  public static String method1() {
    return "A.method1()";
  }
  public String method2() {
    return "A.method2() > " + method1();
  }
}

public class B extends A {
  public static String method1() {
    return "B.method1()";
  }


  public static void main(String[] args) {
    System.out.println("1. " + new A().method2());
    System.out.println("2. " + new B().method2());
    System.out.println("3. " + A.method1());
    System.out.println("4. " + B.method1());
    System.out.println("5. " + method1());
  }
}
```



Constructors

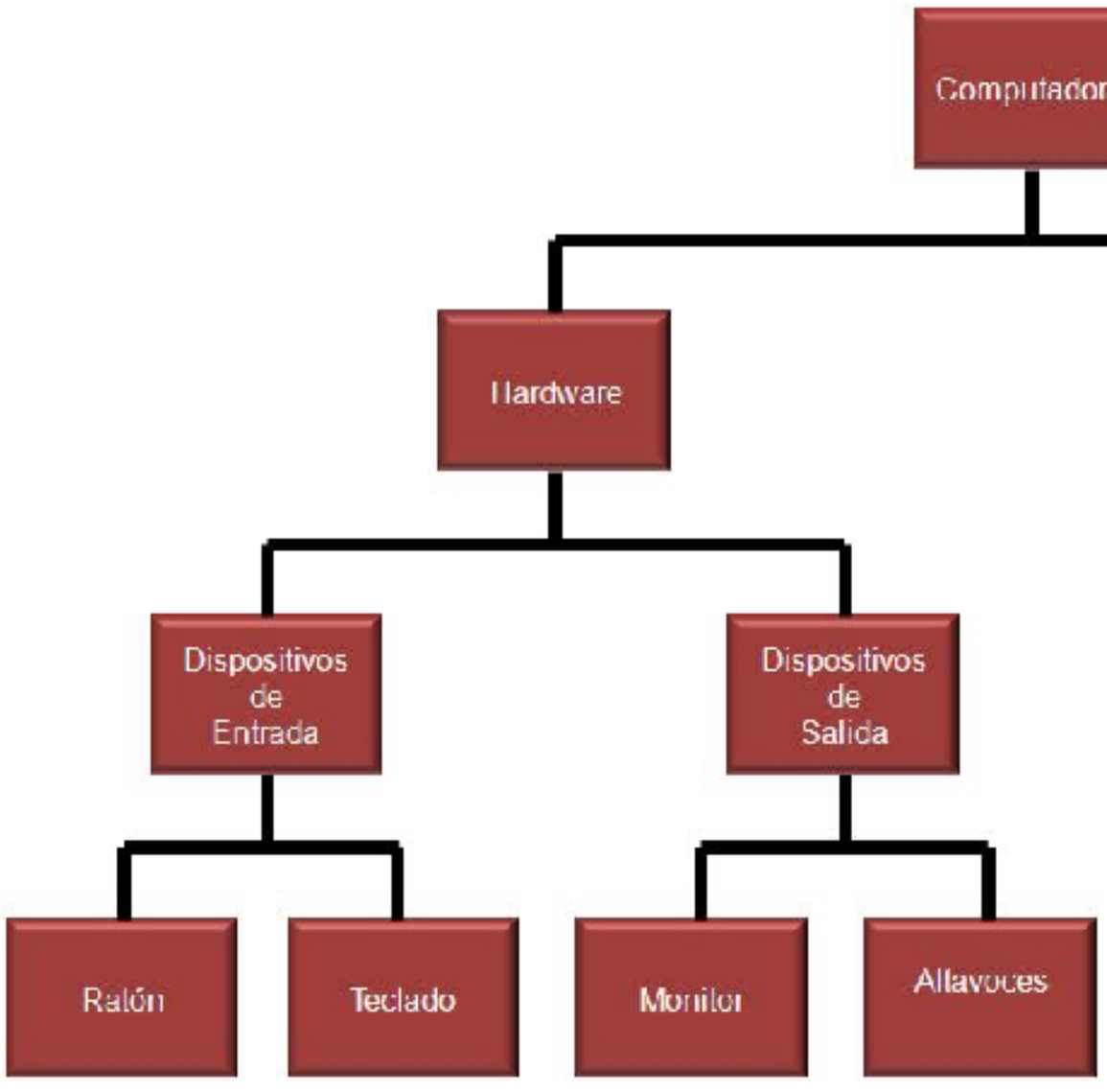
```java
public class A {
  public A() {
    System.out.print("new A() > ");
  }
}

public class B extends A {
  public B() {
    System.out.print("new B() > ");
  }
}

public class C extends A {
  public C () {
    System.out.print("new C() > ");
  }
  public C (String name) {
    System.out.print("new C(" + name + ") > ");
  }
}

public class D extends C {
  public D() {
    this("dog");
    System.out.print("new D() > ");
  }
  public D(String name) {
    super(name);
    System.out.print("new D(" + name + ") > ");
  }
  public D(int number) {
    super();
    System.out.print("new D(" + number + ") > ");
  }
}
```



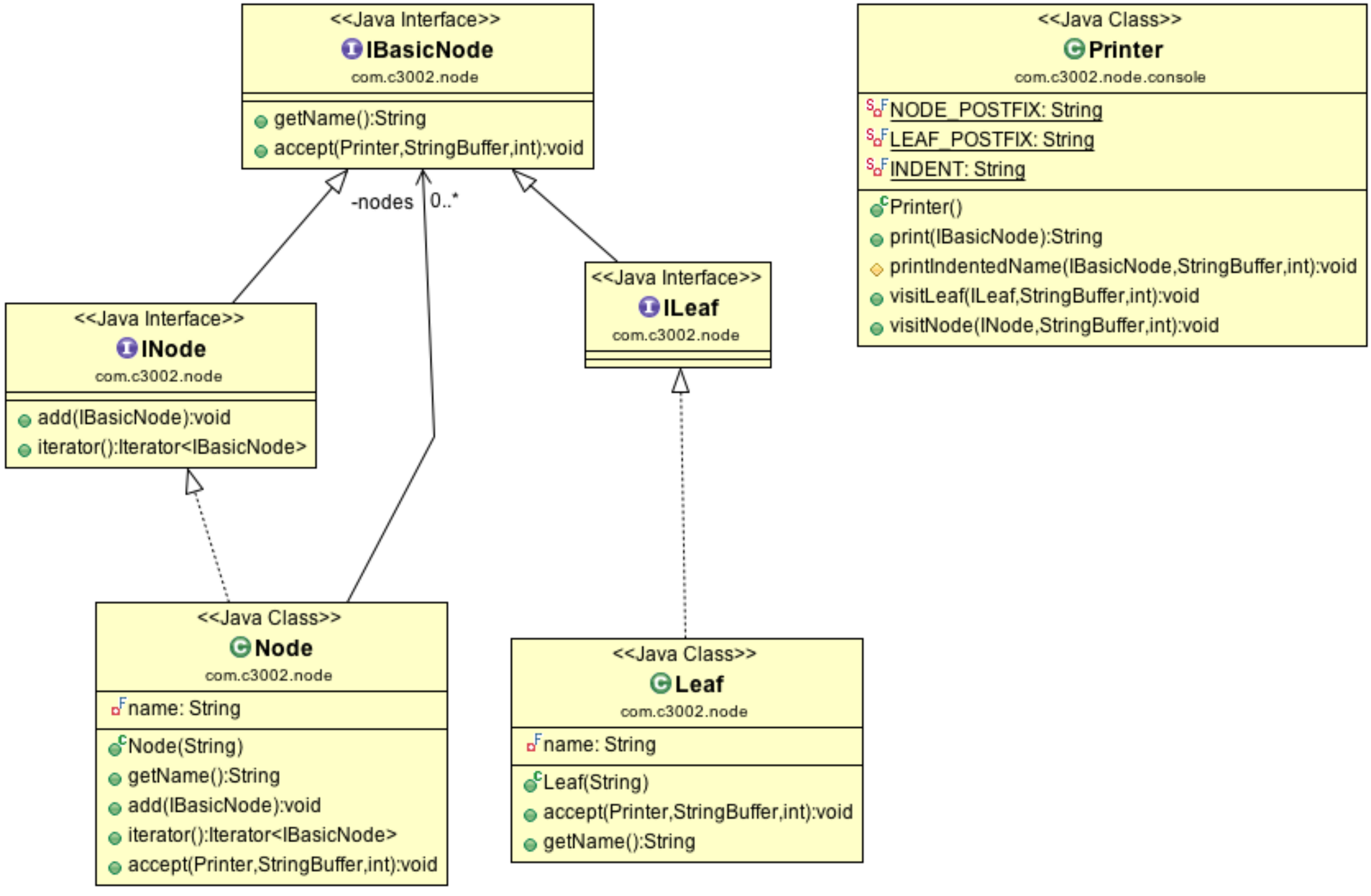

```java
public static void main(String[] args) {
  System.out.print("1. "); new A();
  System.out.print("2. "); new B();
  System.out.print("3. "); new C();
  System.out.print("4. "); new C("animal");
  System.out.print("5. "); new D();
  System.out.print("6. "); new D("jirafa");
  System.out.print("7. "); new D(3);
}
```

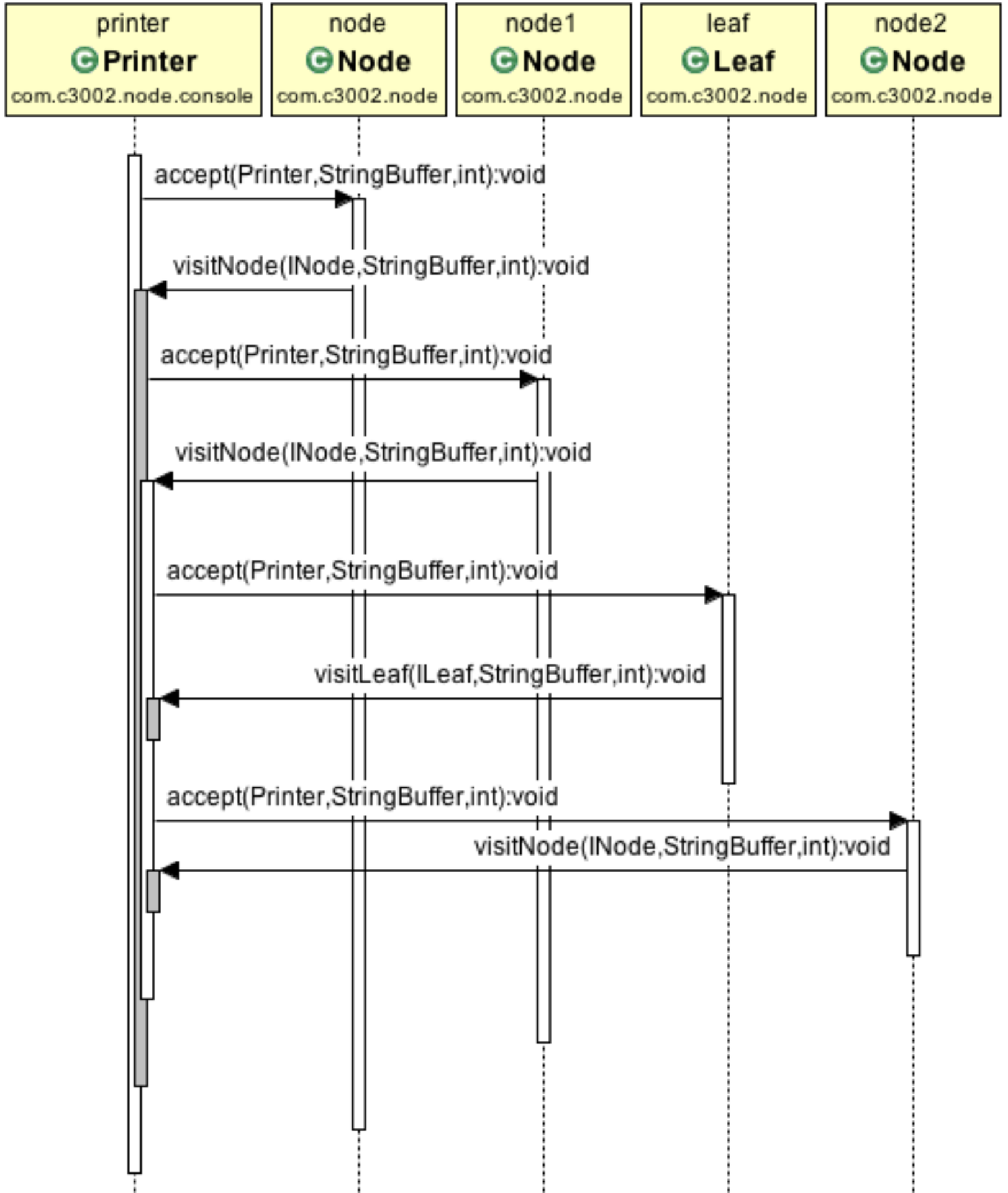
Double Dispatch

Ejercicio: Imprime un Árbol




```
Computadora
  Hardware
    Dispositivos de Entrada
      Ratón
      Teclado
    Dispositivos de Salida
      Monitor
      Altavoces
  Software
    …
```


<<Java Interface>>
IBasicNode
com.c3002.node
getName():String
accept(Printer,StringBuffer,int):void
-nodes
0..*
<<Java Interface>>
INode
com.c3002.node
add(IBasicNode):void
iterator():Iterator<IBasicNode>
<<Java Interface>>
ILeaf
com.c3002.node
<<Java Class>>
Printer
com.c3002.node.console
NODE_POSTFIX: String
LEAF_POSTFIX: String
INDENT: String
Printer()
print(IBasicNode):String
printIndentedName(IBasicNode,StringBuffer,int):void
visitLeaf(ILeaf,StringBuffer,int):void
visitNode(INode,StringBuffer,int):void
<<Java Class>>
Node
com.c3002.node
name: String
Node(String)
getName():String
add(IBasicNode):void
iterator():Iterator<IBasicNode>
accept(Printer,StringBuffer,int):void
<<Java Class>>
Leaf
com.c3002.node
name: String
Leaf(String)
accept(Printer,StringBuffer,int):void
getName():String


printer
Printer
com.c3002.node.console
node
Node
com.c3002.node
node1
Node
com.c3002.node
leaf
Leaf
com.c3002.node
node2
Node
com.c3002.node
accept(Printer,StringBuffer,int):void
visitNode(INode,StringBuffer,int):void
accept(Printer,StringBuffer,int):void
visitNode(INode,StringBuffer,int):void
accept(Printer,StringBuffer,int):void
visitLeaf(ILeaf,StringBuffer,int):void
accept(Printer,StringBuffer,int):void
visitNode(INode,StringBuffer,int):void