Tag	XMLBefore	XMLAfter

Abstraction/Class generation	<o base="fibonacci$small"
	cut="11"
	line="9"
	name="small"
	ref="9">
	<o as="n" base="n" level="1" ref="3"/>
	<o as="@" base="@" level="1" ref="4"/>
	<o as="rec" base="rec" level="1" ref="15"/>
	</o>
		public final class EOfibonacci$EOsmall extends PhDefault {
		public EOfibonacci$EOsmall() {
		this(new PhEta());
		}
		public EOfibonacci$EOsmall(final Phi parent) {
		super(parent);
		this.add("n", new AtFree(/* default */));
		this.add("φ", new AtBound(new AtOnce(new AtLambda(this, self -> {
		Phi ret_base_base = new PhMethod(self, "n");
		Phi ret_base = new PhMethod(ret_base_base, "eq");
		ret_base = new PhCopy(ret_base);
		Phi ret_base_1 = new org.eolang.EOint(self);
		ret_base_1 = new PhCopy(ret_base_1);
		ret_base_1 = new PhWith(ret_base_1, "data", new Data.Value<Long>(2L));
		ret_base = new PhWith(ret_base, 0, ret_base_1);
		Phi ret = new PhMethod(ret_base, "if");
		ret = new PhCopy(ret);
		Phi ret_1 = new org.eolang.EOint(self);
		ret_1 = new PhCopy(ret_1);
		ret_1 = new PhWith(ret_1, "data", new Data.Value<Long>(1L));
		Phi ret_2 = new PhMethod(self, "n");
		ret = new PhWith(ret, 0, ret_1);
		ret = new PhWith(ret, 1, ret_2);
		return ret;
		})));
		}
		}

Application	<o ancestors="1" line="9" name="fibonacci$small" original-name="small" parent="fibonacci"> <o line="9" name="n"/> <o base=".if" line="10" name="@"> <o base=".eq" line="11" method=""> <o base="n" line="11" ref="9"/> <o base="org.eolang.int" data="int" line="11">2</o> </o> <o base="org.eolang.int" data="int" line="12">1</o> <o base="n" line="13" ref="9"/> </o> <o level="1" line="9.15" name="rec"/> </o>	</java> </class> <class ancestors="1" java-name="EOfibonacci$EOsmall" line="9" name="fibonacci$small" original-name="small" parent="fibonacci"> <java>