

Discussion on multimethods

More on multi methods

- enable multi-methods by default to avoid surprises
- very useful for equals() method...

```
class Cheese {
    @Property name

    boolean equals(Cheese that) {
        return this.name == that.name
    }

    // the equals(Object) version is auto-overloaded for you
}
```

- see example below

```
class Foo {
  def m(x){
    foo(x)

    // same as
    Foo f = this
    f.foo(x)
  }

  def foo(x){1}
}

class Foo2 extends Foo {
  def foo(String a){2}

  // this method is autogenerated under the covers
  // unless the user has defined a new foo(Object)
  def foo(x) {
    if (x instanceof String) {
      return foo((String) x)
    }
    else
      return super.foo(x)
  }
}

def f = new Foo2()
assert f.foo("a") == 2
assert f.m("a") == 2

Foo g = new Foo2()
assert g.foo("a") == 2
assert g.m("a") == 2

Foo2 h = new Foo2()
assert h.foo("a") == 2
assert h.m("a") == 2
```