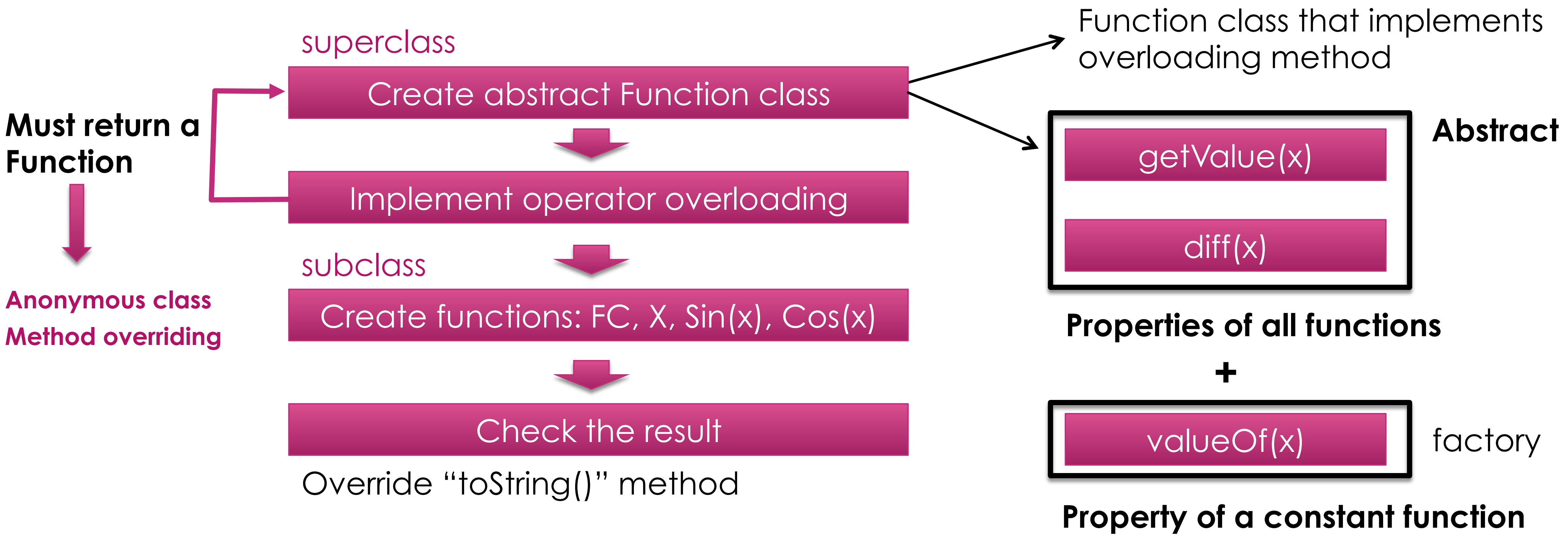
Symbolic Functions

Symbolic Functions using OpOv

✓ **Symbolic math** (univariate, **x**) using operator overloading


superclass
Create abstract Function class
Implement operator overloading
subclass
Create functions: FC, X, Sin(x), Cos(x)
Check the result
Override "toString()" method
Must return a Function
Anonymous class
Method overriding
Function class that implements overloading method
getValue(x)
diff(x)
Abstract
Properties of all functions
+
valueOf(x)
factory
Property of a constant function

Tip: Use "Anonymous class" and "method overriding".

Symbolic Functions using OpOv

✓ Useful math relations for operator overloading:

$$(f \pm g)' = f' \pm g'$$

$$\left(\frac{f}{g}\right)' = \frac{f'g - fg'}{g^2}$$

$$(fg)' = f'g + fg'$$

Chain rule: composition of function

$$(f(g(x))' = g'f'(g(x))$$

✓ Elementary functions were defined with **no-arg constructor: new Sin() →** sin(x) what if sin(2*x)

✓ How to implement **composition** of functions? $h(x) = f(g(x))$

- What is the best way?
    - Using a constructor or utility class?
        - new Sin(Function f) → Sin(f(x))
    - Let's try using constructor of subclasses since the behavior of composition is function specific

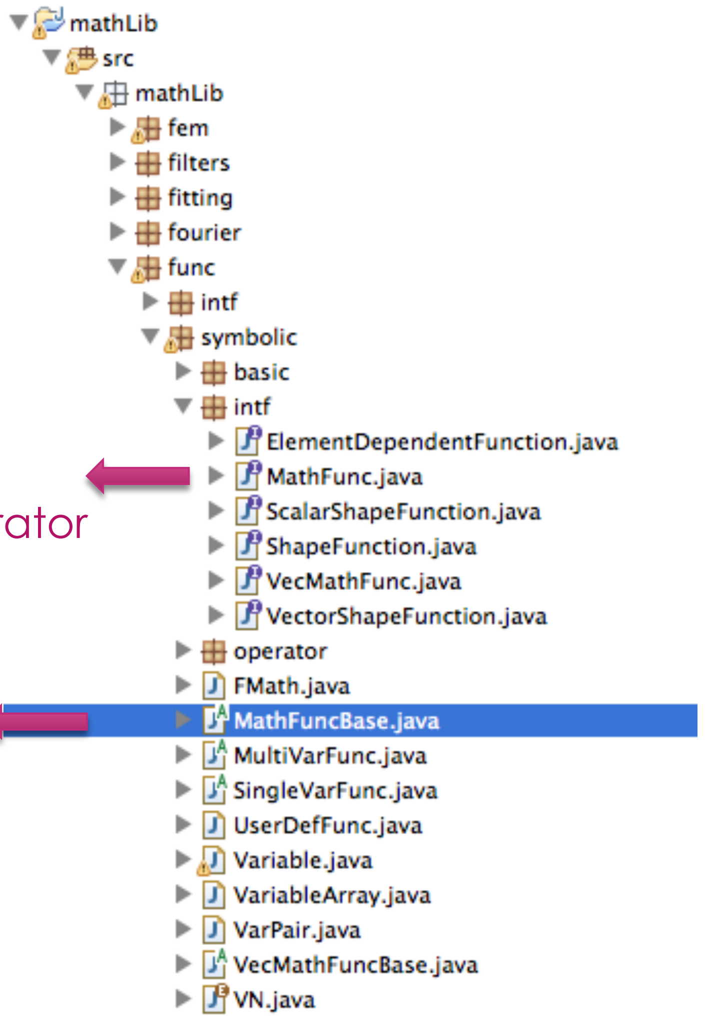
Operator Overloading

- ✓ More advanced version of symbolic math is in "mathLib" library
  - Package: **mathLib.func.symbolic**
- ✓ Adopted from this github repository:
  - **https://github.com/lymanzhang/Futureye_v2**
- ✓ Here's how to use it:

1. Import **static** from **FMath**
   - Contains all the basic functions: x, sin, …
   - Contains all the basic math operations
     - ✓ pow, log, …
2. Use MathFunc interface to define functions
3. All the operators are overloaded for MathFunc type

Function interface implementing operator overloading

Abstract function



```
▼ mathLib
  ▼ src
    ▼ mathLib
      ▶ fem
      ▶ filters
      ▶ fitting
      ▶ fourier
      ▼ func
        ▶ intf
        ▼ symbolic
          ▶ basic
          ▼ intf
            ▶ ElementDependentFunction.java
            ▶ MathFunc.java
            ▶ ScalarShapeFunction.java
            ▶ ShapeFunction.java
            ▶ VecMathFunc.java
            ▶ VectorShapeFunction.java
          ▶ operator
          ▶ FMath.java
          ▶ MathFuncBase.java
          ▶ MultiVarFunc.java
          ▶ SingleVarFunc.java
          ▶ UserDefFunc.java
          ▶ Variable.java
          ▶ VariableArray.java
          ▶ VarPair.java
          ▶ VecMathFuncBase.java
          ▶ VN.java
```