Reflection

- Provides access to type information at run-time
  - Interface discovery
  - Object inspection
  - Plug-ins

- if (x instanceof Foo) ... basic type info. access

- class Class

  How do you get instances of type Class?

  x.getClass() ⇒ Class c

  Class.forName("java.awt.Color");

  Color.class

  A Class instance is immutable, but has lots of accessors...
java.lang.reflect

Class has

Field[] getFields() — all public fields (variables) of the class, including inherited
Field[] getDeclaredFields() — all fields declared in this class

Class getSuperclass()

Method[] getMethods()
Method[] getDeclaredMethods()

Constructor[] getConstructors()
Primitives:

**Integer**

**Integer.TYPE** represents the primitive type `int`

See code examples for applications of reflection:

- `DataViewer`
- `YOPS`