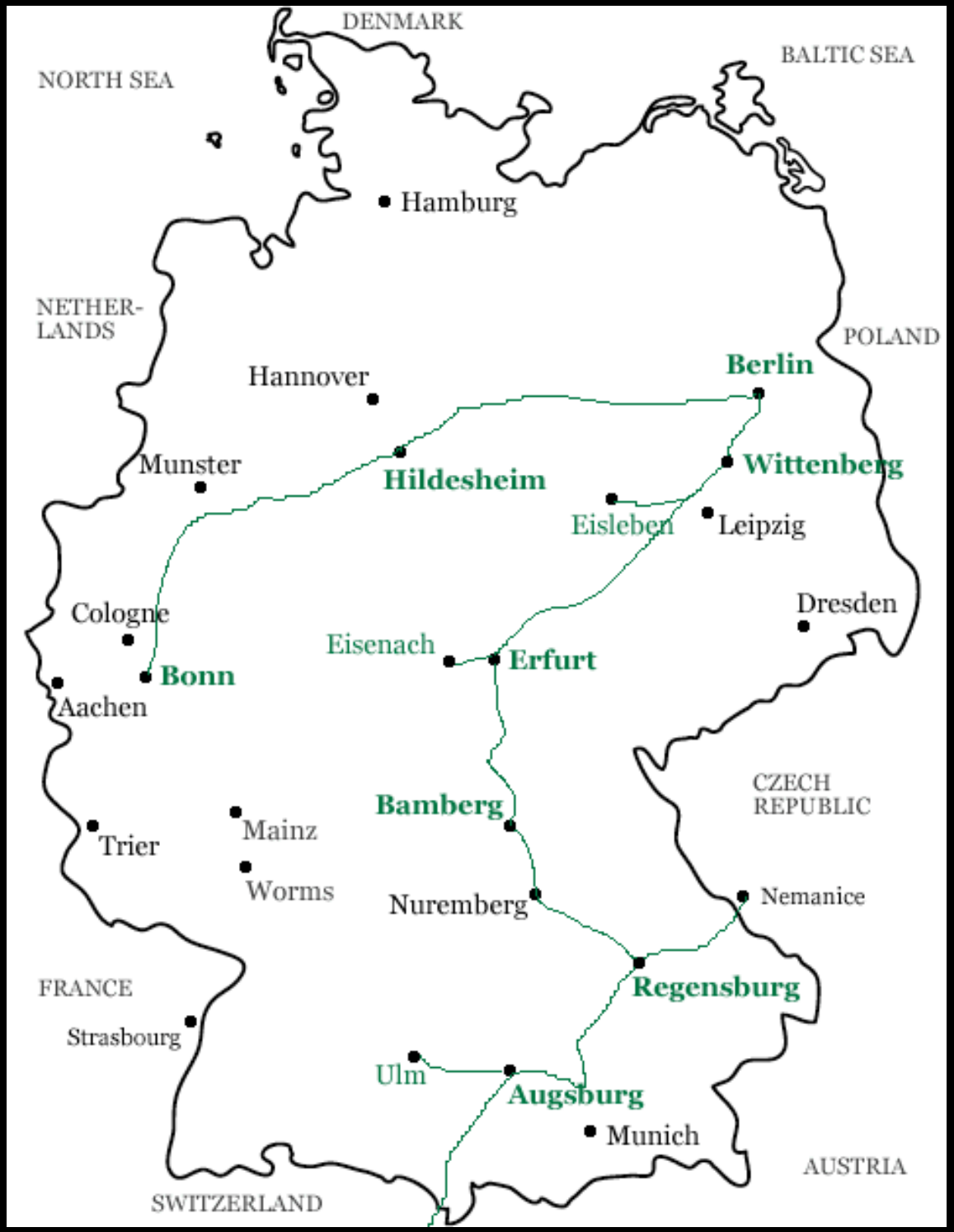
Be Careful,
Your Java is Showing

Joe O'Brien, artisan
EdgeCase, LLC


DENMARK
BALTIC SEA
NORTH SEA
Hamburg
NETHER-
LANDS
POLAND
Berlin
Hannover
Munster
Hildesheim
Wittenberg
Eisleben
Leipzig
Dresden
Cologne
Eisenach
Erfurt
Bonn
Aachen
CZECH
REPUBLIC
Bamberg
Mainz
Trier
Worms
Nuremberg
Nemanice
Regensburg
FRANCE
Strasbourg
Ulm
Augsburg
Munich
AUSTRIA
SWITZERLAND

“A language that doesn't affect the way you think about programming, is not worth knowing. “

Alan J. Perlis, Yale University


A

.NET / Java Background

EIA / SOA

ThoughtWorks

Columbus Ruby Brigade

EdgeCase

conventions

```
ClassNames

method_names

@instance_variables
```

$global_variables

SOME_CONSTANTS

OtherConstants

```
if_asking_a_question?
```

String

all?
any?
between?
empty?
include?
...

```
if_doing_something_dangerous!
```

String

capitalize
capitalize!

chomp
chomp!

...

two spaces

```ruby
namespace "p4" do
  desc "Display a list of files that are not in p4."
  task :missing do
    puts P4.new.missing.collect { |fn| fn.sub(/^/, '? ') }
  end

  desc "Files to be ignored by p4"
  task :ignored do
    puts P4.new.ignore_files
  end

  desc "Add all missing files to p4."
  task :add_missing => :require_run_from_root do
    P4.new.missing.each do |fn|
      system "p4 add #{fn}"
    end
  end
```

convert

tabs

to

spaces

idioms

```
if ! something?
  do_more_stuff
  and_even_more
end
```

```
unless something?
  do_more_stuff
  and_even_more
end
```

```
if something?
  do_more_stuff
end

unless something?
  do_more_stuff
end
```

```
do_more_stuff if something?

do_more_stuff unless something?
```

```
nil == false
```

```
if this_object.nil?
  do_more_stuff
  and_even_more
end
```

```
unless this_object
  do_more_stuff
  and_even_more
end
```

```
def get_age
  @age
end

def set_age(age_in)
  @age = age_in
end
```

```
attr_accessor :age
attr_reader   :age
attr_writer   :age
```

```
foo ||= Foo.new
```

```
e = “oh’

s = “Me ” + e + “ my”
```

```
e = “oh’

s = “Me #{e} my”
```

loops

```
for item in @items
  puts item.name
end
```

```
@items.each do |item|
  puts item
end
```

```
# watch the dots!

(1..100).each do |i|
  puts “#{i} time”
end
```

```
99.times do |i|
  puts “#{i} time”
end
```

code == data

```
@items.each do |item|
  puts item
end
```

```
 try {
     Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
     String url = "jdbc:odbc:SSPer";
     Connection con =
          DriverManager.
          getConnection(url,"username","password");
     Statement stmt = con.createStatement();
     ResultSet rs = stmt.executeQuery
       ("SELECT name, age FROM user");

     while (rs.next()) {
         // HERE IS WHAT I NEED
     }
     stmt.close();
 } catch (java.lang.Exception ex) {
     // Error handling
 } finally {
     con.close();
}
```

```
File.open("myfile") do |file|
  file.each_byte { ...}
end
```

interfaces

```
module MyInterface
  def method_one
    raise “not implemented yet”
  end
end

class MyJavaClass
  include MyInterface
end
```

config files

```
RAILS_GEM_VERSION = '2.0.2' unless defined? RAILS_GEM_VERSION

require File.join(File.dirname(__FILE__), 'boot')

Rails::Initializer.run do |config|
  # config.frameworks -= [ :active_record, :active_resource, ...

  # config.plugins = [ :exception_notification, :ssl_requirement, ...

  # config.load_paths += %W( #{RAILS_ROOT}/extras )
  # config.log_level = :debug

  config.action_controller.session = {
    :session_key => '_black_book_session',
    :secret      => 'my_secret'
  }
  # config.action_controller.session_store = :active_record_store
  # config.active_record.schema_format = :sql

  # config.active_record.observers = :cacher, :garbage_collector

  # config.active_record.default_timezone = :utc
end
```

class methods

```
class Contact
  def Contact.count
    # ...
  end
  def Contact.find_some
    # ...
  end
end
```

```
class Contact
  def self.count
    # ...
  end
  def self.find_some
    # ...
  end
end
```

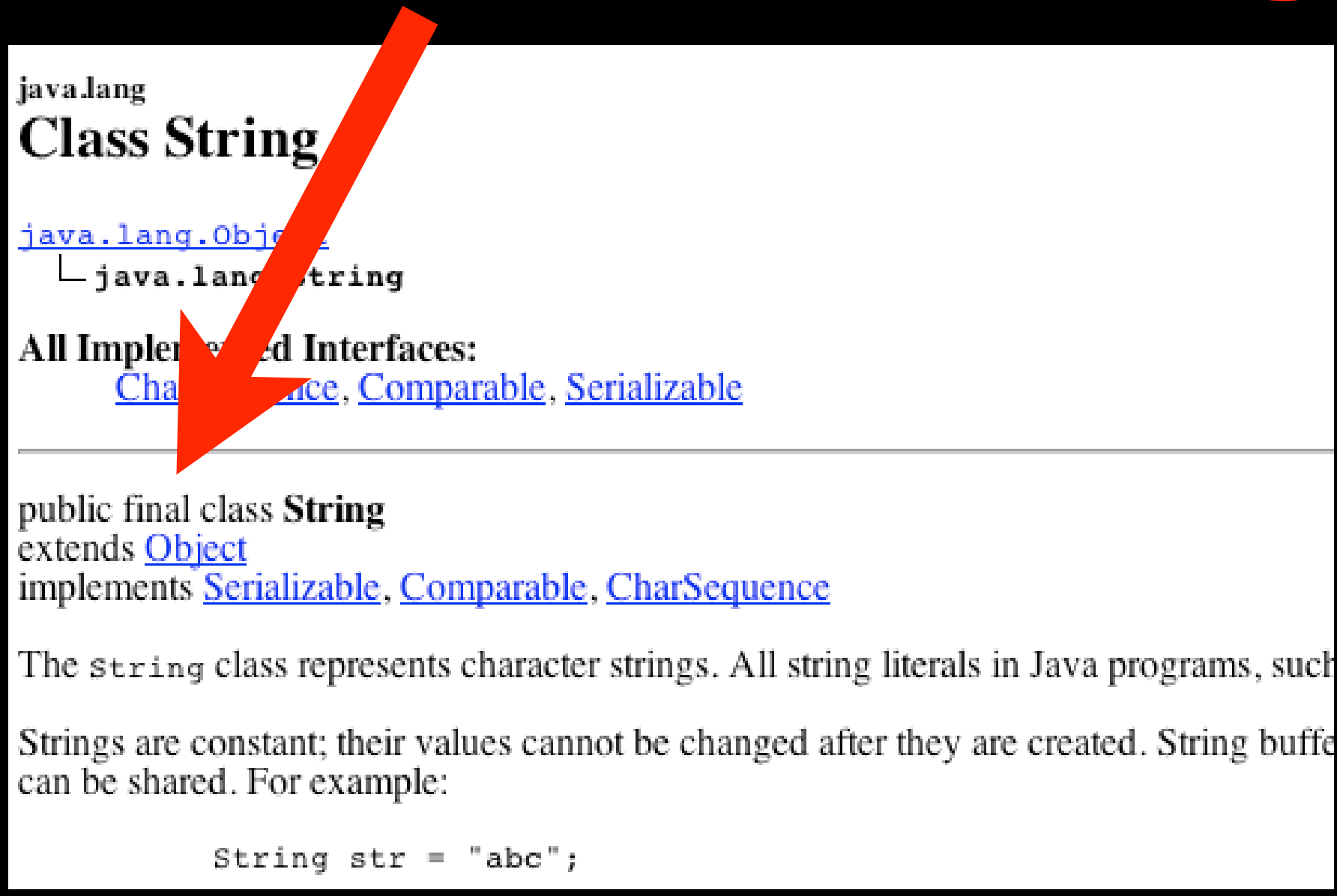
```ruby
class Contact
  class << self
    def count
      # ...
    end
    def find_some
      # ...
    end
  end
end
```

```
class Contact
  class << self
    def count
      # ...
    end
    def find_some
      # ...
    end
  end
end
```

behavior

Utility Classes

are you kidding?



java.lang

Class String

java.lang.Object
  └ java.lang.String

All Implemented Interfaces:
CharSequence, Comparable, Serializable

---

public final class **String**
extends Object
implements Serializable, Comparable, CharSequence

The `String` class represents character strings. All string literals in Java programs, such

Strings are constant; their values cannot be changed after they are created. String buffe
can be shared. For example:

```
String str = "abc";
```

```
def pm(hour)
  # ...
end

pm(3)
```

```
class Numeric
  def am
    # ...
  end
  def pm
    # ...
  end
end
```

```
3.pm

4.am

30.days.from_now
```

```ruby
module Kernel
  private

  def noon
    12.pm
  end

  def midnight
    12.am
  end
end
```

changing functionality

```
class ExistingClass


  def out

    # ...
  end

end
```

```
class ExistingClass

  alias_method :old_out, :out

  def out
    old_out if conditions_not_right
    # do your stuff
  end

end
```

```
def method_missing(name, *args)
  super(*args) if conditions_not_right
  # do your stuff
end
```

meta programming

```
class Resident
  has_many :entitlements
end
```

```
def test_resident_should_have_many_entitlements
  association = Resident.
    reflect_on_association(:entitlements)

  assert_not_nil association,
    "Could not find an association for entitlements"

  assert_equal :has_many, association.macro

  assert association.klass.
    column_names.include?
    (association.primary_key_name),
    'Expected the association to have' +
    'a column referencing the class'
end
```

```
def assert_model_has_many(model, association_name)
  association = Resident.
    reflect_on_association(association_name.to_sym)

  assert_not_nil association,
    "Could not find an association for
    #{association_name}"

  assert_equal :has_many, association.macro

  assert association.klass.
    column_names.include?
    (association.primary_key_name)
end
```

```
class ResidentTest < ActiveSupport::TestCase

  def test_resident_should_have_many_entitlements

    assert_model_has_many Resident, :entitlements

  end

end
```

```ruby
class Resident < ActiveRecord::Base
  has_many :entitlements
end

class ResidentTest < ActiveSupport::TestCase
  def test_resident_should_have_many_entitlements
    assert_model_has_many Resident, :entitlements
  end
end
```

“If I can write it in one line, I want to test it in one line”

Stuart Halloway, Relevance LLC
thinkrelevance.com

```
class Resident < ActiveRecord::Base
  has_many :entitlements
end

class ResidentTest < ActiveSupport::TestCase
  test_should_have_many :entitlements
end
```

```
class Test::Unit::TestCase
   class << self
 # ...
    def test_has_many(association_name)
      evaluate create_association_test(association_name, :has_many)
    end
 # ....
    private
    def create_association_test(association_name, association_type)
      class_under_test = self.to_s.gsub('Test', '').constantize
      meth =  "def test_should_have_association_#{assoc....
      meth << "  model_name = #{class_under_test}\n"
      meth << "  association = model_name.reflect_on_association....
      meth << "  assert_not_nil association, \"Could not find an ...
      meth << "  assert_equal :#{association_type}, association.mac...
      meth <<    add_custom_checks(association_type, class_under_test)
      meth << "end"
      meth
    end
     def evaluate(string)
       puts "\n\n#{string}\n\n" if ENV['DEBUG']
       class_eval string, __FILE__, __LINE__
     end
  end
```

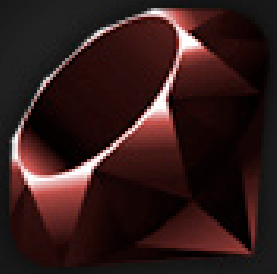

Default
class Resident < ActiveRecord::Base
has_many :entitlements
has_many :weigh_ins
belongs_to :resident_type
belongs_to :maritial_status
belongs_to :ethnicity
validates_presence_of :first_name
validates_presence_of :last_name
validates_presence_of :social_security_number
validates_presence_of :gender
validates_inclusion_of :gender, :in => %w{ Male Female Unk\
nown }
def us_citizen
(read_attribute(:us_citizen) ? "Yes" : "No")
end
-11-:%%-F1 resident.rb Top L1 (Ruby)----------
Loading /Users/objo/.elisp/packages/rubydb3x.el (source)...d\

Default

```
require File.dirname(__FILE__) + '/../test_helper'

class ResidentTest < ActiveSupport::TestCase
  test_has_many :entitlements
  test_has_many :weigh_ins
  test_belongs_to :resident_type
  test_belongs_to :maritial_status
  test_belongs_to :ethnicity

  test_should_validate_presence_of :first_name
  test_should_validate_presence_of :last_name
  test_should_validate_presence_of :social_security_number
  test_should_validate_presence_of :gender

  def test_resident_returns_citizen_as_yes_or_no
    assert_equal 'Yes', resident_with_citizenship(true).us_c\
itizen
    assert_equal 'No', resident_with_citizenship(false).us_c\
itizen
-11-:%%-F1  resident_test.rb    Top L1      (Ruby)-------------
```

method_missing

class_eval



erubycon

SUMMER 2008

"The Enterprise is not ready for Ruby, it is desperate for it."

– Glenn Vanderburg

What I hear, I forget.
What I see, I remember.
What I do, I understand.

- Kung Fu Tzu (Confucius)

Twitter: objo

objo.com

theedgecase.com