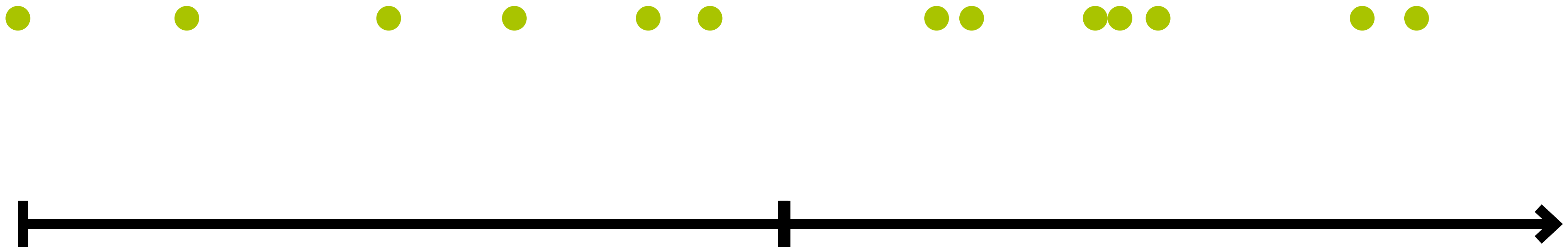
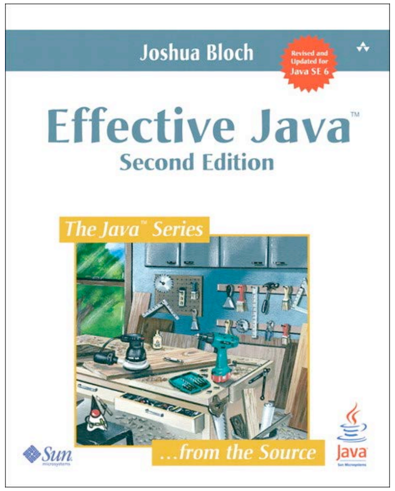
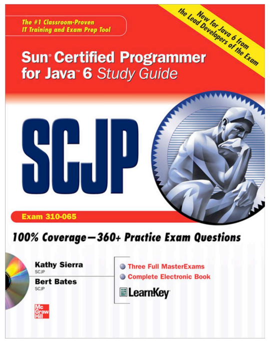
What’s next?

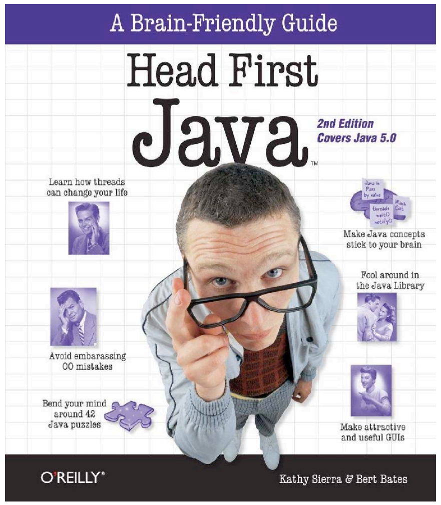
java@peterbloem.nl



```
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```




Sun Certified Programmer for Java 6 Study Guide
SCJP
Exam 310-065
100% Coverage—360+ Practice Exam Questions
Kathy Sierra
Bert Bates



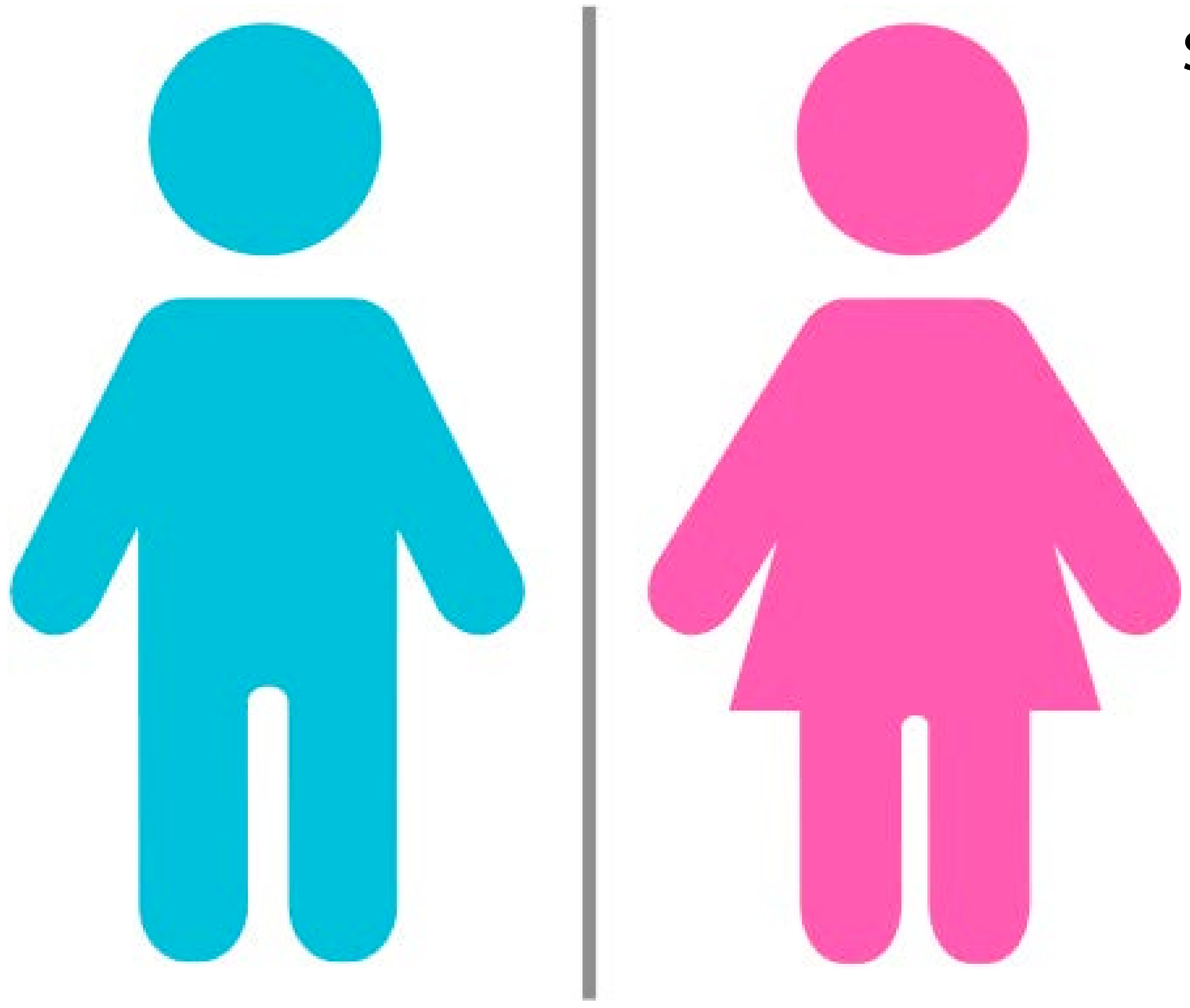
I you don’t feel stupid yet

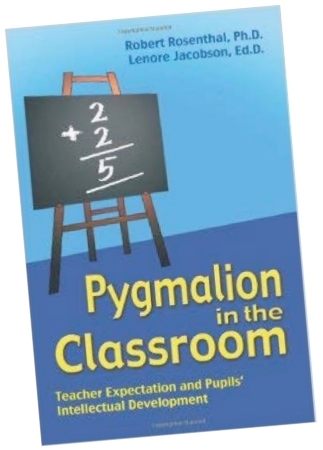
1. Discrete wiskunde (volgend blok)
2. Lineaire Algebra (tweede semester)
3. Formele Talen (tweede semester)
4. Numerical Recipes (tweede jaar)
5. Statistiek (tweede jaar)
6. Master
7. PhD

Dweck, Blackwell 2007:

http://www.npr.org/templates/story/story.php?storyId=7406521 pygmalion/golem effect

stereotype threat





Robert Rosenthal, Ph.D.
Lenore Jacobson, Ed.D.

Pygmalion in the Classroom

Teacher Expectation and Pupils' Intellectual Development

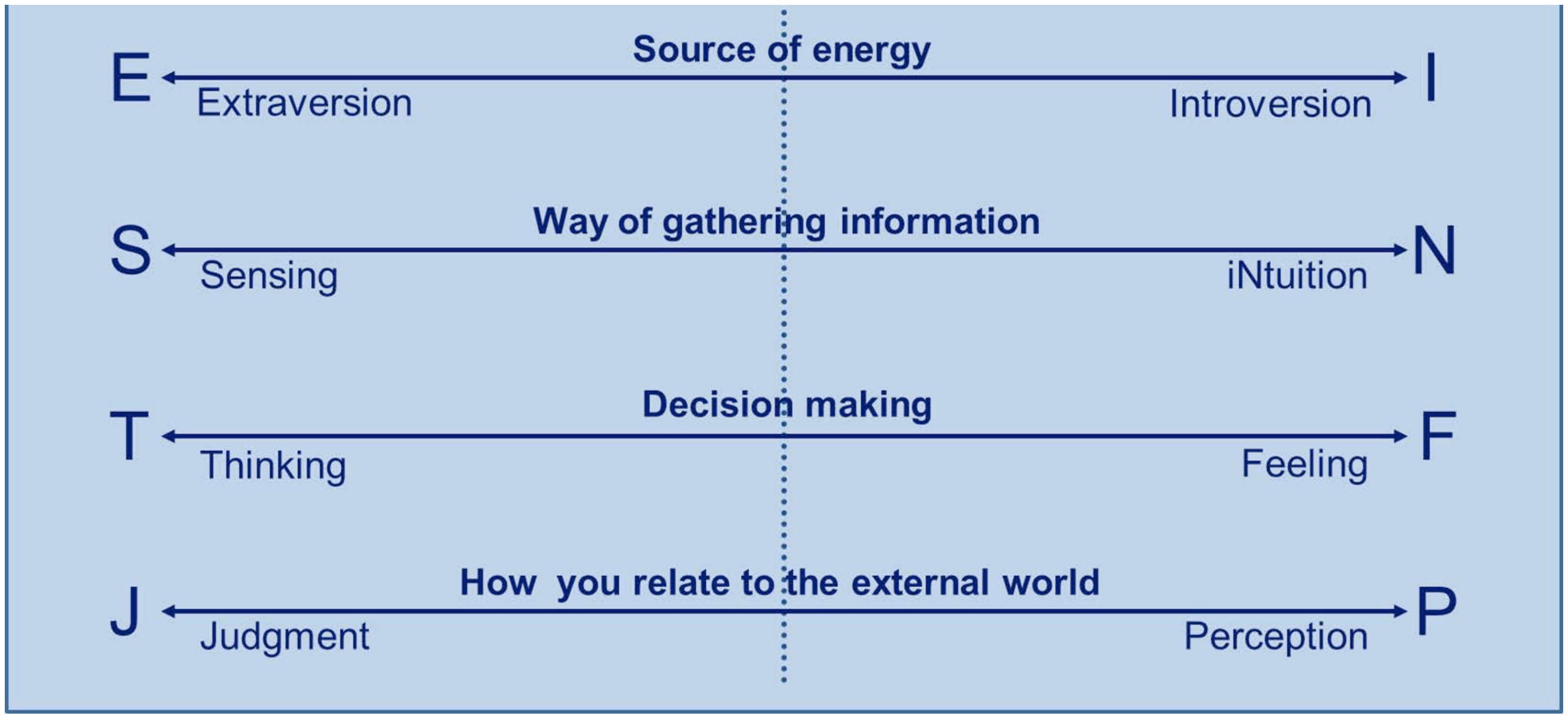
CORDELIA FINE

The Real Science Behind Sex Differences

DELUSIONS OF GENDER

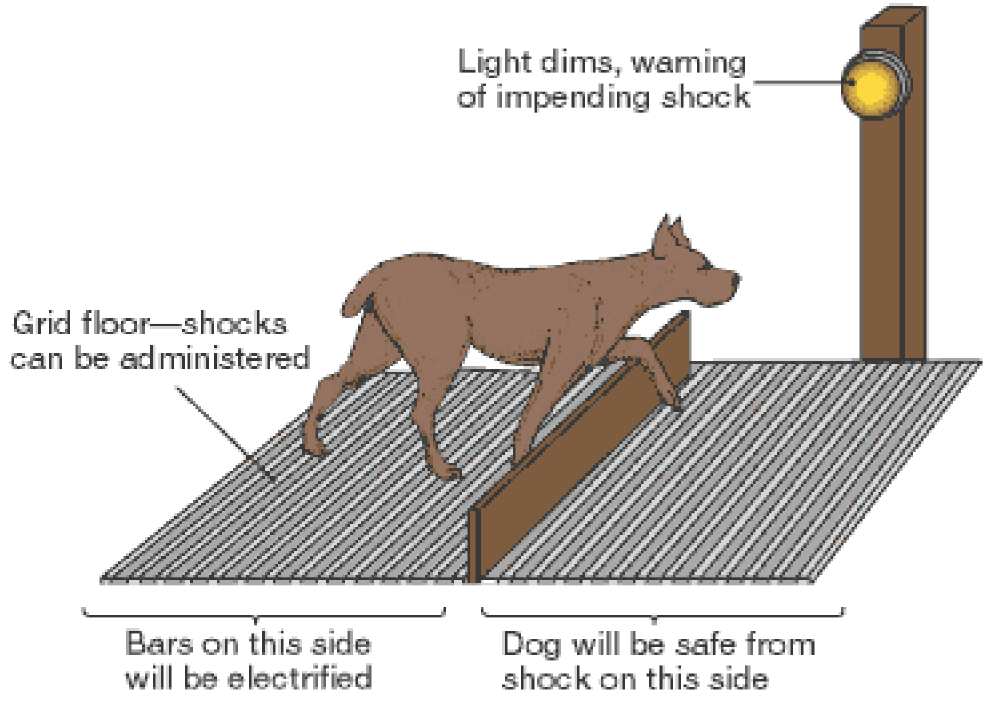
Shortlisted for the John Llewellyn Rhys Prize

Myers-Briggs



Evans, Simkin, 1989

Learned Helplessness


Light dims, warning of impending shock
Grid floor—shocks can be administered
Bars on this side will be electrified
Dog will be safe from shock on this side

Seligman, Maier, 1967

goed nieuws:

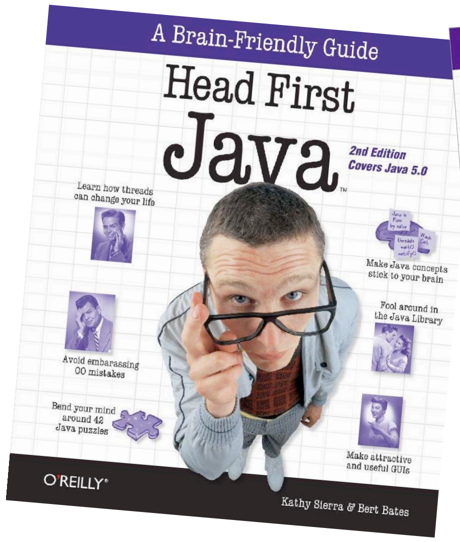
Je kunt je achterstand inhalen

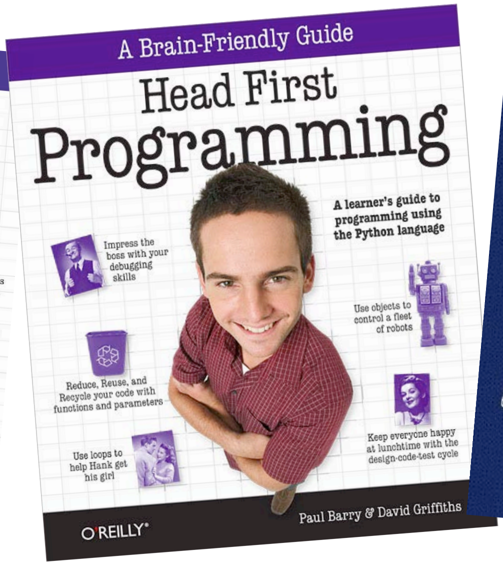
slecht nieuws:

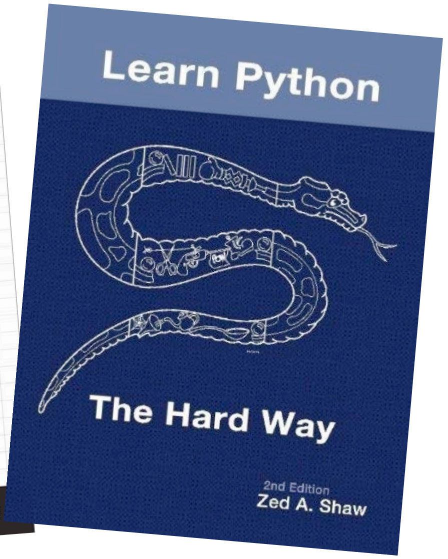
Je moet je achterstand inhalen

Bepaal je eigen ritme

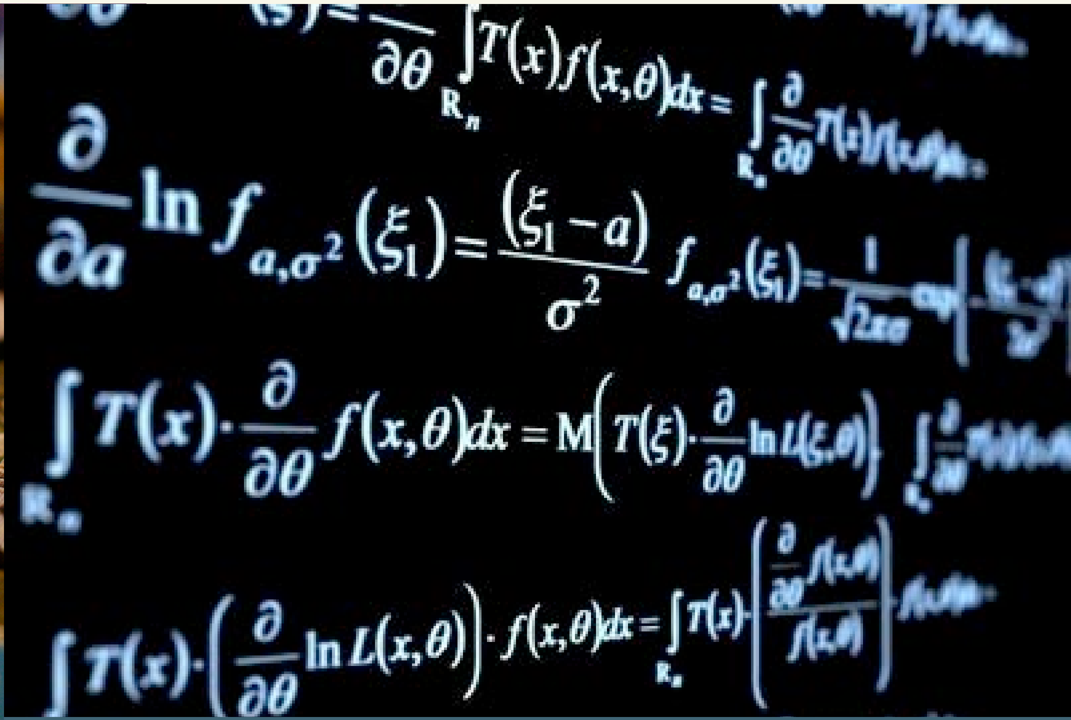
```
https://go-left.com/blog/programming/
                                 100-little-programming-exercises/

https://projecteuler.net/

http://learnpythonthehardway.org/

http://codegolf.stackexchange.com/
```

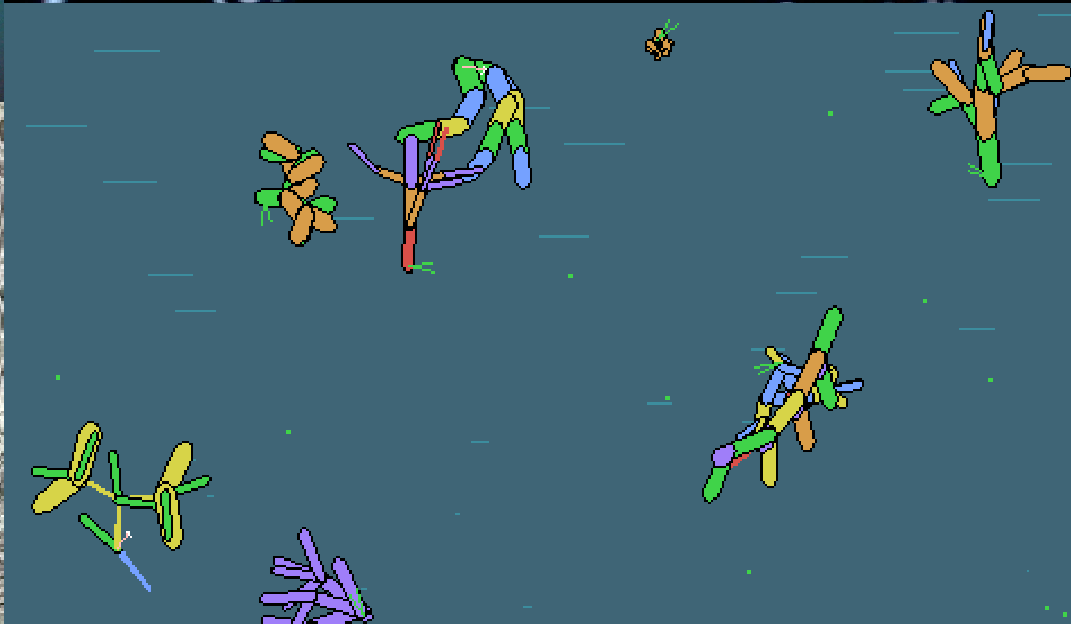



A Brain-Friendly Guide
Head First Programming
A learner's guide to programming using the Python language
Paul Barry & David Griffiths
O'REILLY



Wat is je motivatie?





Programmeren

probleem

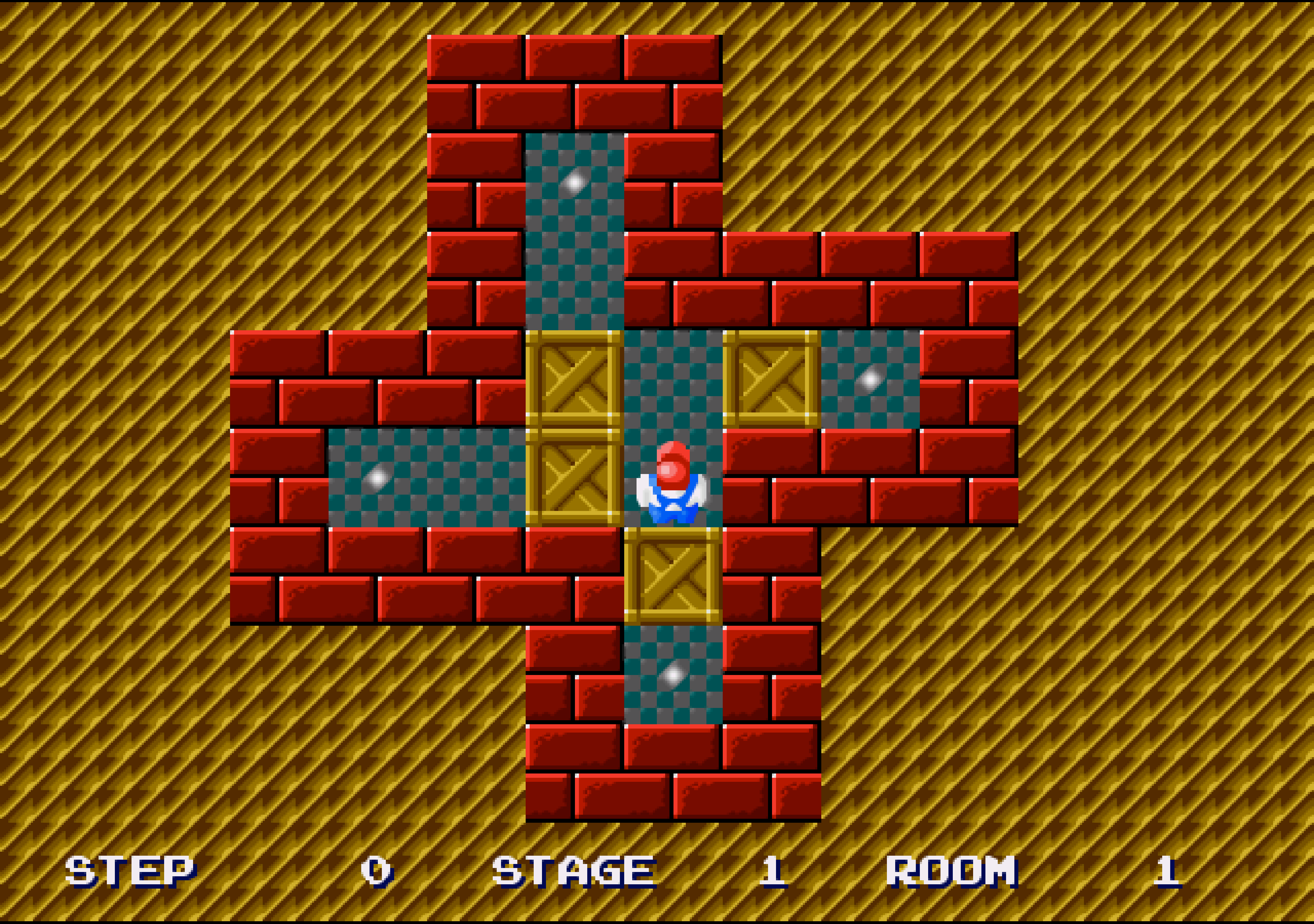
String omdraaien

pseudocode

```
functie(str)
    nw = nieuwe string

    for(i = 0 to l(str))
        i' = l(str) - i
        nw[i'] = str[i]
```

Java

```java
String rev(String s)
{
  int n = s.length();
  StringBuilder sb =
    new StringBuilder();

  for(
    int i = n-1;
    i >= 0;
    i--)
        sb.append(
            s.charAt());
}
```


STEP 0 STAGE 1 ROOM 1

Divide and Conquer

```
/**
 * Difference in days between two days.
 */
public int difference(Date d1, Date d2)
{
    // ??
}
```

Divide and Conquer

```
/**
 * Difference in days between two days.
 */
public int difference(Date date1, Date date2)
{
    int d1 = date1.daysSince1Jan1970();
    int d2 = date2.daysSince1Jan1970();

    return d1 - d2;
}

class Date
{
    public daysSince1Jan1970()
    {
        ...
    }
}
```

Doodling

Andere talen

- Python
- Ruby
- Javascript

Bomen/bos

variabelen
control flow
(ifs/loops)
methoden
logische expressies
wiskunde
Collections
Strings

Objecten
Generics
Exceptions
Encapsulation
Object copying
Information hiding
Interfaces

Eigen generics
Reguliere Exp.
Files en IO
Autoboxing
Floating point
arithmetic

Onthoud het idee, niet de regel

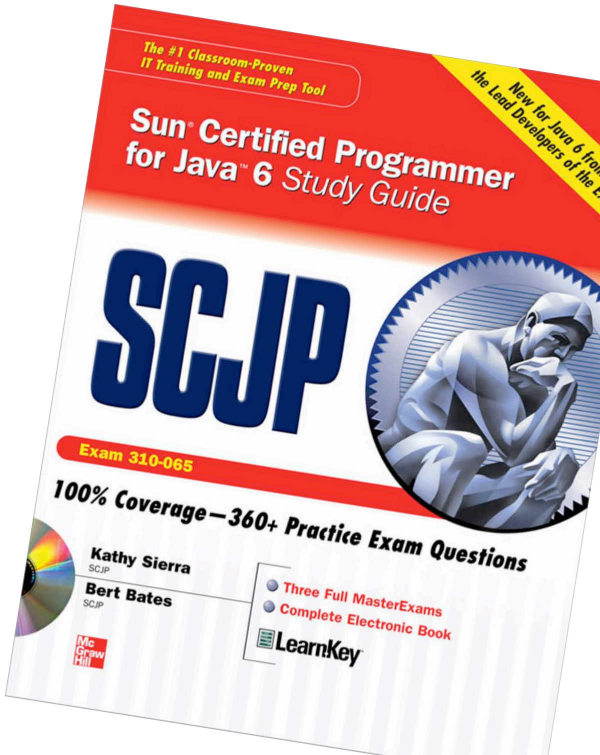
Encapsulation

Onthoud het idee, niet de regel

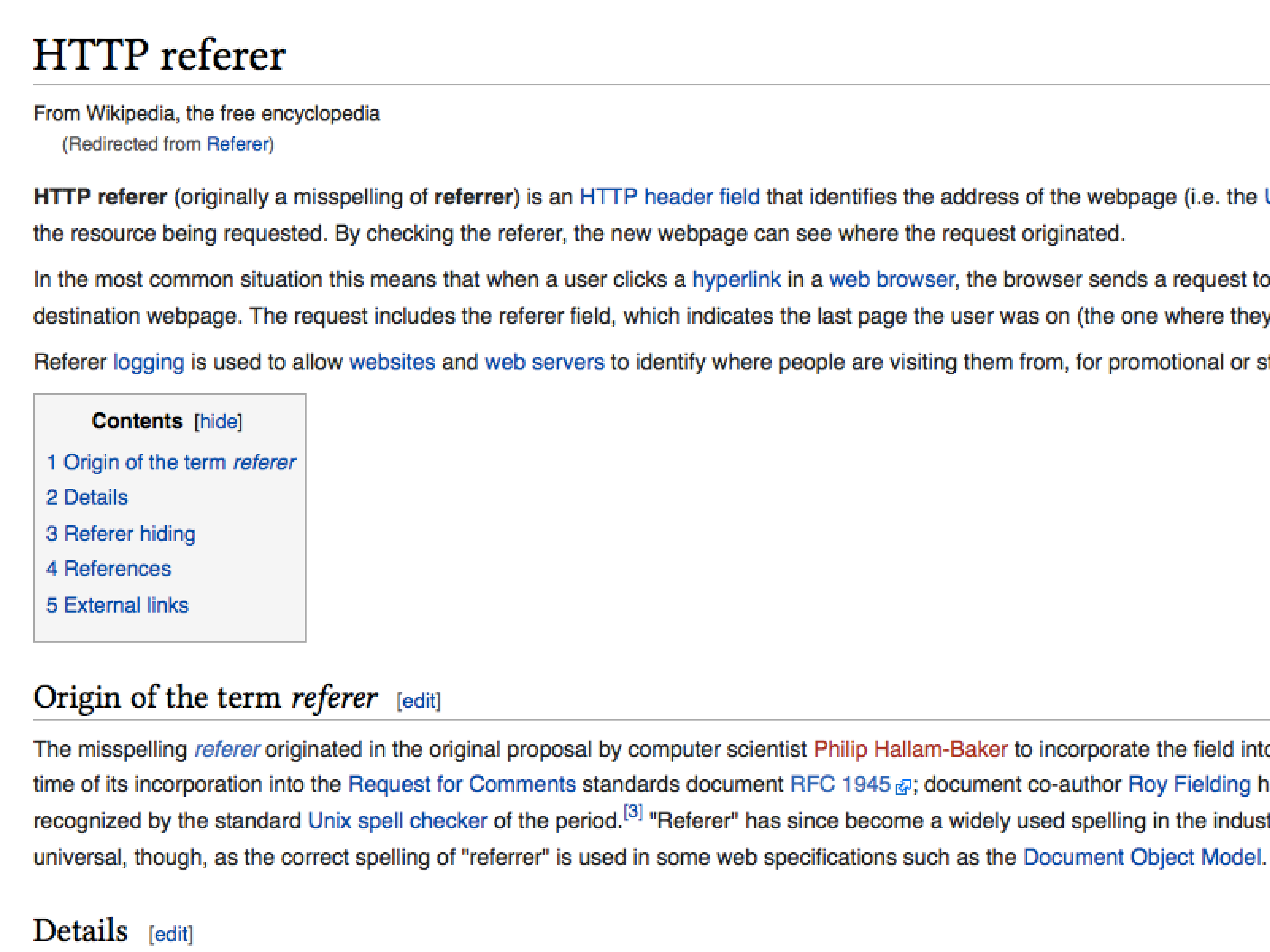
**Gebruiker,
Interface,
Implementatie**

Voor het tentamen

- OOP (static vs instance)
- Typing: objectreferenties
- Inheritance
- Interfaces en abstract classes
- Exceptions
- casting, instanceof
- primitives en Objecten



`SCJP Sun Certified Programmer for Java 6 Study Guide, Kathy Sierra & Bert Bates`



HTTP referer

From Wikipedia, the free encyclopedia

(Redirected from Referer)

HTTP referer (originally a misspelling of **referrer**) is an HTTP header field that identifies the address of the webpage (i.e. the U
the resource being requested. By checking the referer, the new webpage can see where the request originated.

In the most common situation this means that when a user clicks a hyperlink in a web browser, the browser sends a request to
destination webpage. The request includes the referer field, which indicates the last page the user was on (the one where they

Referer logging is used to allow websites and web servers to identify where people are visiting them from, for promotional or st

Contents [hide]

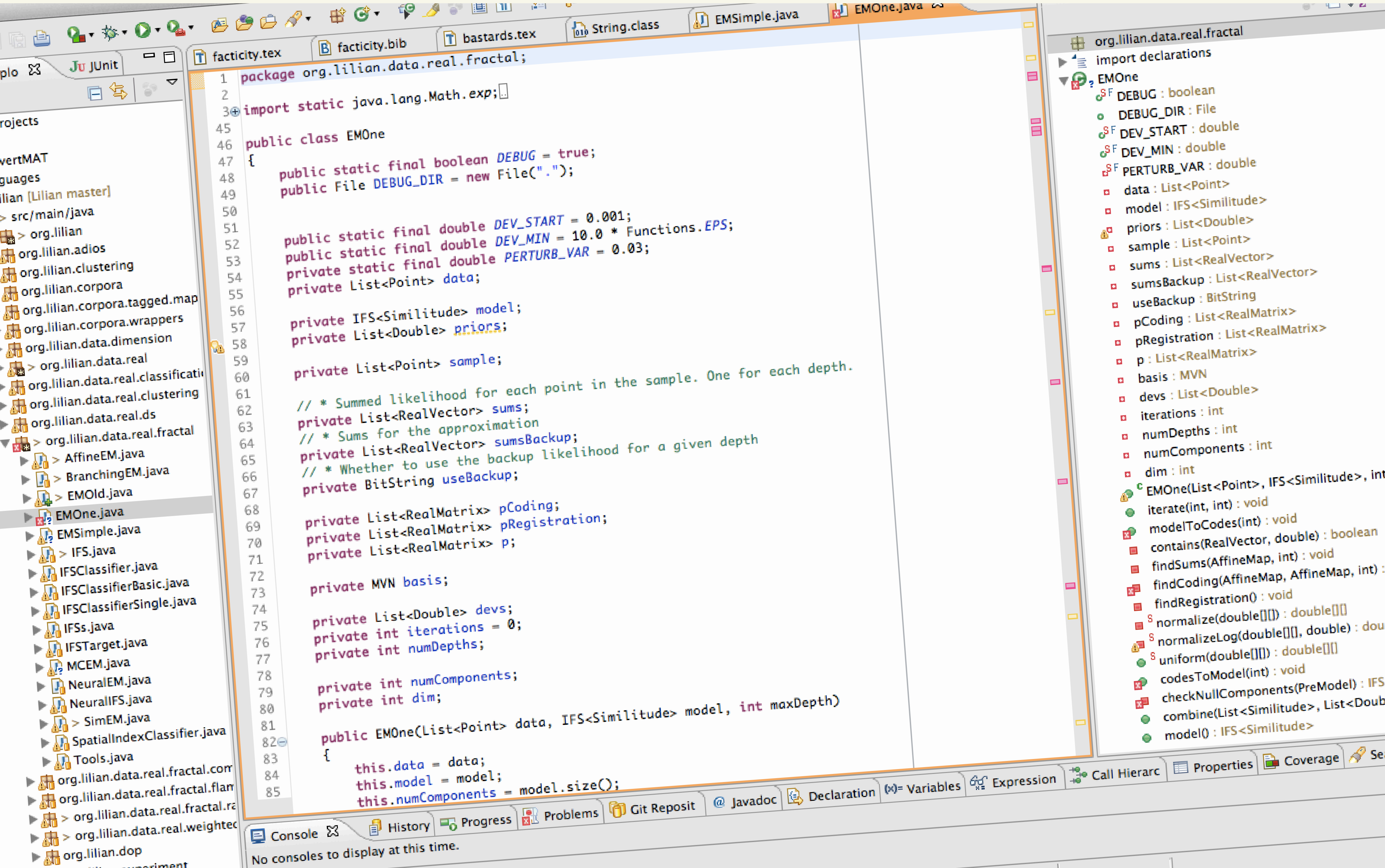
Origin of the term *referer* [edit]

The misspelling *referer* originated in the original proposal by computer scientist Philip Hallam-Baker to incorporate the field into
time of its incorporation into the Request for Comments standards document RFC 1945; document co-author Roy Fielding h
recognized by the standard Unix spell checker of the period.[3] "Referer" has since become a widely used spelling in the indust
universal, though, as the correct spelling of "referrer" is used in some web specifications such as the Document Object Model.

Details [edit]

vragen?

IDE



plo ✕ Ju JUnit

rojects

vertMAT
guages
ilian [Lilian master]
> src/main/java
> org.lilian
org.lilian.adios
org.lilian.clustering
org.lilian.corpora
org.lilian.corpora.tagged.map
org.lilian.corpora.wrappers
org.lilian.data.dimension
> org.lilian.data.real
org.lilian.data.real.classificati
org.lilian.data.real.clustering
org.lilian.data.real.ds
> org.lilian.data.real.fractal
> AffineEM.java
> BranchingEM.java
> EMOld.java
EMOne.java
EMSimple.java
> IFS.java
IFSClassifier.java
IFSClassifierBasic.java
IFSClassifierSingle.java
IFSs.java
IFSTarget.java
MCEM.java
NeuralEM.java
NeuralIFS.java
> SimEM.java
SpatialIndexClassifier.java
Tools.java
org.lilian.data.real.fractal.com
org.lilian.data.real.fractal.flam
> org.lilian.data.real.fractal.ra
> org.lilian.data.real.weighte
org.lilian.dop

facticity.tex | facticity.bib | bastards.tex | String.class | EMSimple.java | EMOne.java ✕

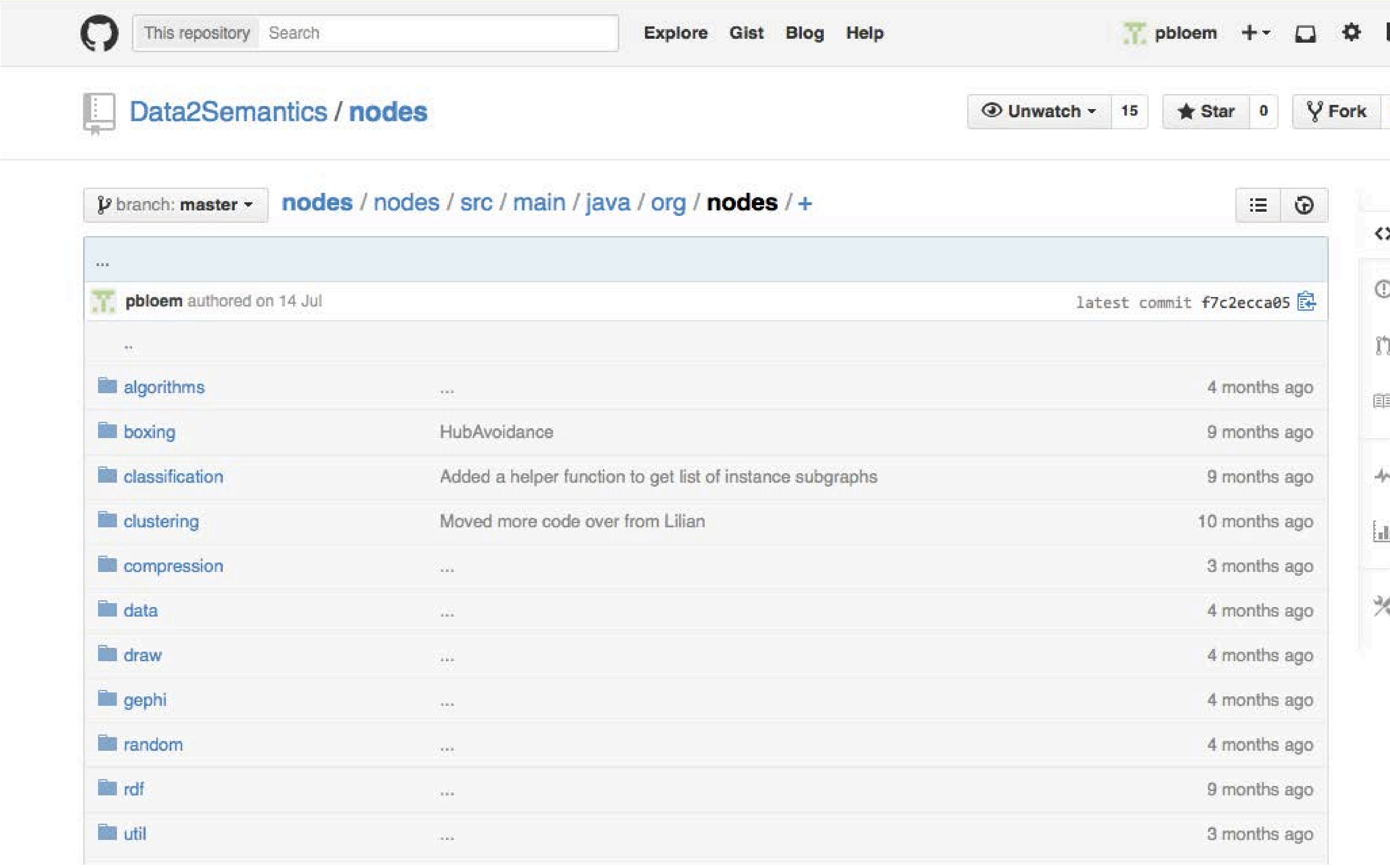
```
package org.lilian.data.real.fractal;

import static java.lang.Math.exp;

public class EMOne
{
	public static final boolean DEBUG = true;
	public File DEBUG_DIR = new File(".");


	public static final double DEV_START = 0.001;
	public static final double DEV_MIN = 10.0 * Functions.EPS;
	private static final double PERTURB_VAR = 0.03;
	private List<Point> data;

	private IFS<Similitude> model;
	private List<Double> priors;

	private List<Point> sample;

	// * Summed likelihood for each point in the sample. One for each depth.
	private List<RealVector> sums;
	// * Sums for the approximation
	private List<RealVector> sumsBackup;
	// * Whether to use the backup likelihood for a given depth
	private BitString useBackup;

	private List<RealMatrix> pCoding;
	private List<RealMatrix> pRegistration;
	private List<RealMatrix> p;

	private MVN basis;

	private List<Double> devs;
	private int iterations = 0;
	private int numDepths;

	private int numComponents;
	private int dim;

	public EMOne(List<Point> data, IFS<Similitude> model, int maxDepth)
	{
		this.data = data;
		this.model = model;
		this.numComponents = model.size();
```

org.lilian.data.real.fractal
import declarations
EMOne
DEBUG : boolean
DEBUG_DIR : File
DEV_START : double
DEV_MIN : double
PERTURB_VAR : double
data : List<Point>
model : IFS<Similitude>
priors : List<Double>
sample : List<Point>
sums : List<RealVector>
sumsBackup : List<RealVector>
useBackup : BitString
pCoding : List<RealMatrix>
pRegistration : List<RealMatrix>
p : List<RealMatrix>
basis : MVN
devs : List<Double>
iterations : int
numDepths : int
numComponents : int
dim : int
EMOne(List<Point>, IFS<Similitude>, int
iterate(int, int) : void
modelToCodes(int) : void
contains(RealVector, double) : boolean
findSums(AffineMap, int) : void
findCoding(AffineMap, AffineMap, int) :
findRegistration() : void
normalize(double[][]) : double[][]
normalizeLog(double[][], double) : dou
uniform(double[][]) : double[][]
codesToModel(int) : void
checkNullComponents(PreModel) : IFS
combine(List<Similitude>, List<Doub
model() : IFS<Similitude>

Javadoc | Declaration | (x)= Variables | Expression | Call Hierarc | Properties | Coverage | Se
Console ✕ | History | Progress | Problems | Git Reposit

No consoles to display at this time.

Versioning



This repository Search | Explore | Gist | Blog | Help | pbloem

Data2Semantics / nodes

Unwatch 15 | Star 0 | Fork

branch: master | nodes / nodes / src / main / java / org / nodes / +

...

pbloem authored on 14 Jul — latest commit f7c2ecca05

..

algorithms	...	4 months ago
boxing	HubAvoidance	9 months ago
classification	Added a helper function to get list of instance subgraphs	9 months ago
clustering	Moved more code over from Lilian	10 months ago
compression	...	3 months ago
data	...	4 months ago
draw	...	4 months ago
gephi	...	4 months ago
random	...	4 months ago
rdf	...	9 months ago
util	...	3 months ago

Package management & deployment

maven

gradle

Unit testing

```
@Test
public void testJBC()
{
    DGraph<String> graph = Graphs.jbcDirected();

    List<Integer> nodes = Arrays.asList(13, 15, 16);

    DGraph<String> subgraph = Subgraph.dSubgraphIndices(graph, nodes);
    System.out.println(subgraph);

    assertEquals(3, subgraph.size());
    assertEquals(2, subgraph.numLinks());
}
```

```java
public boolean containsNumber(List<Object> list)
{
    return (list instanceof List<Number>);
}
```

Type erasure

```
list instanceof List<?>
```

@SuppressWarnings

```
private MyNode first, second;

@SuppressWarnings("unchecked")
public Collection<? extends DTNode<L, T>> nodes()
{
    return Arrays.asList(first, second);
}
```

Generics: bounds

```
public static double mean(List<Number> list)
{
    double sum = 0.0;

    for (Number n : list)
        sum += n.doubleValue();

    return sum / list.size();
}

List<Integer> ints = ...;
double m = mean(ints) // ?
```

Slechte oplossing

```java
public static <N> double mean(List<N> list)
{
    double sum = 0.0;

    for (N n : list)
        if (n instanceof Number)
            sum += ((Number)n).doubleValue();
        else
            throw new IllegalArgumentException("...");

    return sum / list.size();
}

List<Integer> ints = ...;
double m = mean(ints) // ?
```

Generics: bounds

```java
public static <N extends Number> double mean(List<N> list)
{
    double sum = 0.0;

    for (Number n : list)
        sum += n.doubleValue();

    return sum / list.size();
}

List<Integer> ints = ...;
double m = mean(ints)
```

Generic bounds

```
public class myClass<N extends Foo>
{
    public N get(int i) {}

    public boolean isCorrect(N instance) {}
}

//----

public static <N extends Foo> double m(List<N> input)
{
    ...
    for (N n : input)
        ...
}
```

Bounds

```
<N extends Number>

<N super Integer>


<T extends Comparable<T>>
```

Wildcard: ?

```java
// Collections.java
public static <T> void sort(List<T> list, Comparator<? super T> c)
{
    ...
}
```

Contravariance, covariance, invariance

```
class Animal {}

class Cat extends Animal{}
class Dog extends Animal{}
```

arrays

covariant:	Cat[] is een Animal[]
contravariant:	Animal[] is een Cat[]
Invariant:	geen van beide

Return types en argumenten

```
class AnimalShelter {

    Animal getAnimalForAdoption() {}

    void putAnimal(Animal animal) {}
}

class CatShelter extends AnimalShelter{

    Cat getAnimalForAdoption() {}

    void putAnimal(Cat animal) {}
}
```

<-- covariant return type

<-- contravariant arguments

Met generics

PECS: Producer extends, consumer super

```java
/**
 * list is een consumer
 */
static public double mean(List<? extends Number> list)
{
    ...
}

/**
 * list is een producer
 */
static public void generateRandom(List<? super Number> container)
{
    ...
}
```

3 dingen die we nog niet gezien hebben

Inner classes

```java
public class A
{
    public class B {}
    public static class BStatic {}

    public static void main(String[] args) {
        A a = new A();

        A.B b = a.new B();

        A.BStatic bStat = new A.BStatic();

        // zelfs in een methode
        class Local {}
        Local local = new Local();
    }
}
```

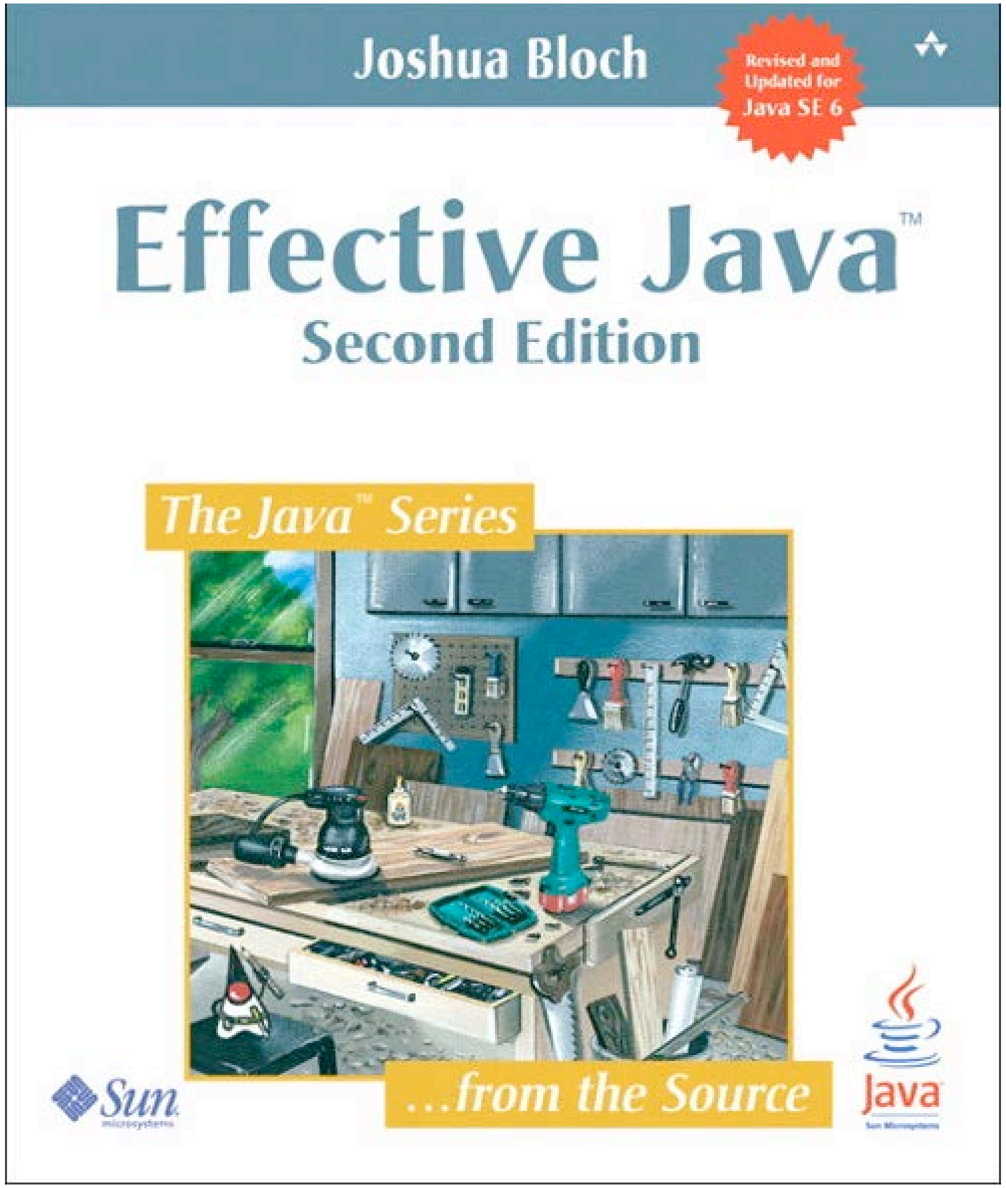
Anonymous classes

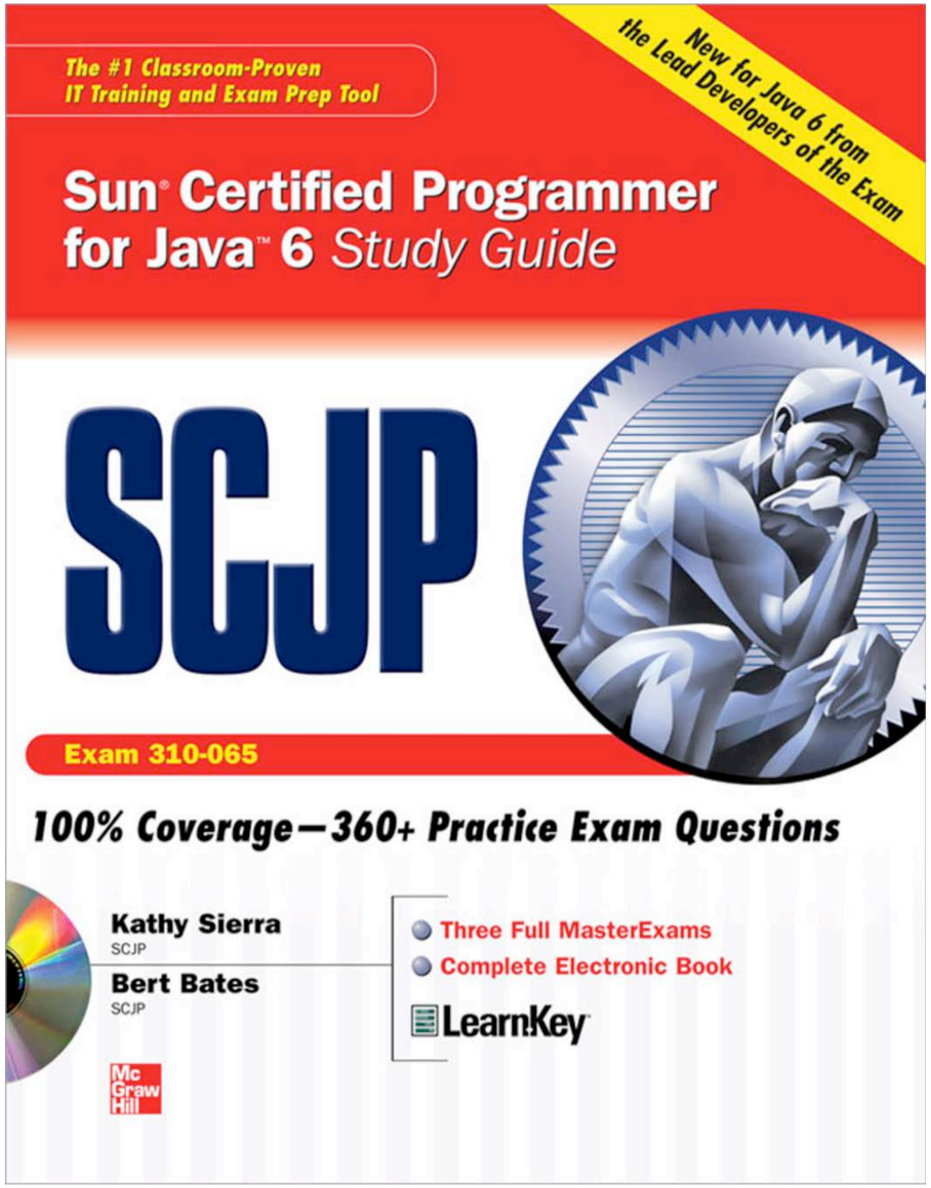
```
public void start(Stage primaryStage) {
       primaryStage.setTitle(“Hello World!”);
       Button btn = new Button();
       btn.setText(“Say ‘Hello World’”);
       btn.setOnAction(new EventHandler<ActionEvent>() {

           @Override
           public void handle(ActionEvent event) {
               System.out.println(“Hello World!”);
           }
       });

       StackPane root = new StackPane();
       root.getChildren().add(btn);
       primaryStage.setScene(new Scene(root, 300, 250));
       primaryStage.show();
   }
```

Closures & Lambda expressions (java 8)

```
class CalculationWindow extends JFrame {
private volatile int result;
  ...
  public void calculateInSeparateThread(final URI uri) {
    // the code () -> { /* code */ } is a closure
    new Thread(() -> {
        calculate(uri);
        result = result + 10;
    }).start();
  }
}
```

1) Boeken


Joshua Bloch
Revised and Updated for Java SE 6
Effective Java™
Second Edition
The Java™ Series
...from the Source
Sun microsystems
Java


The #1 Classroom-Proven IT Training and Exam Prep Tool
New for Java 6 from the Lead Developers of the Exam
Sun Certified Programmer for Java™ 6 Study Guide
SCJP
Exam 310-065
100% Coverage—360+ Practice Exam Questions
Kathy Sierra SCJP
Bert Bates SCJP
Three Full MasterExams
Complete Electronic Book
LearnKey
Mc Graw Hill

2) Sites

http://thedailywtf.com/

THE DAILY WTF
Curious Perversions in Information Technology

http://programmers.stackexchange.com/

codinghorror.com
programmingisterrible.com
joelonsoftware.com

3) Open source

github
SOCIAL CODING