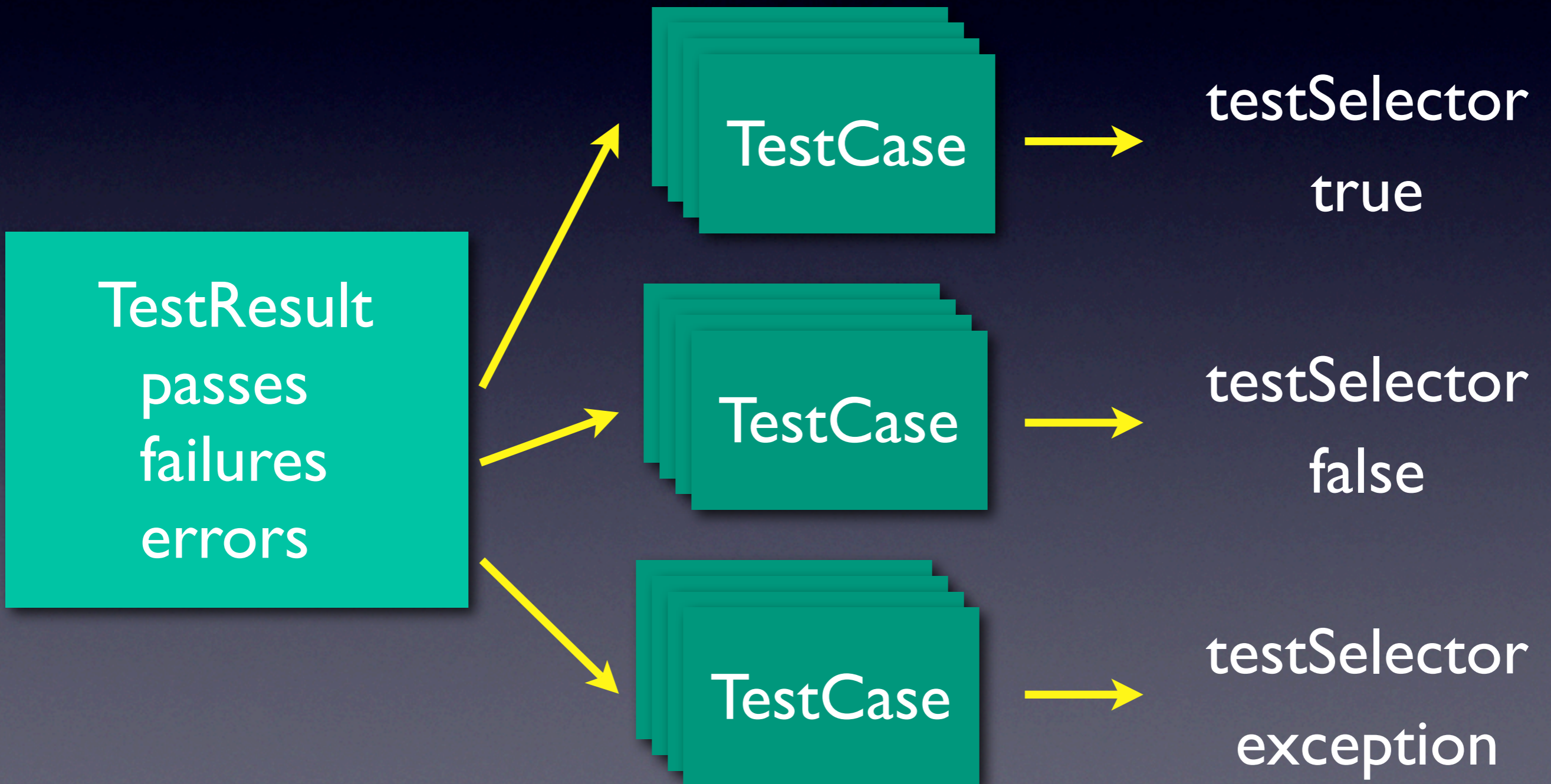


Advanced Techniques for Building Testing Tools

Andrés Valloud

SUnit recap



Hash Analysis Tool

HashResult
dataset
hashFunction
statistics

Dataset

Hash
function

Statistics

Distribution

Time

Verification



Calculation objects

Distribution

Time

Verification

$$\frac{|D|}{|f(D)|}$$

$$\frac{1}{|D|} \sum_{v \in f(D)} (|\varphi(v)| - 1)^2$$

$$\frac{1}{|D|} \sum_{v \in f_p(D)} (|\varphi_p(v)| - 1)^2$$

- Minimum
- Arithmetic mean
- Harmonic mean
- Maximum

Given $x = y$,
then
 $f(x) = f(y)$

Distribution calculations and hash buckets

Distribution

$$\frac{|D|}{|f(D)|}$$

$$\frac{1}{|D|} \sum_{v \in f(D)} (|\varphi(v)| - 1)^2$$

$$\frac{1}{|D|} \sum_{v \in f_p(D)} (|\varphi_p(v)| - 1)^2$$

Detailed information

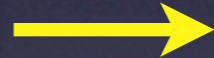
Faster execution

Hash value	Objects
1	d1, d2
4	d40
9	d3, d8, d70
...	...

Hash value	Count
1	2
4	1
9	3
...	...

Evaluation contexts

Distribution



Evaluation context



Hash value	Objects
1	d1, d2
4	d40
9	d3, d8, d70
...	...

Hash value	Count
1	2
4	1
9	3
...	...

Evaluation contexts

Distribution

← resultsInTheContextOf:

←

2 ↓

numberOfObjects

numberOfHashValues

hashValuesAndBucketSizesDo:

numberOfHashValuesMod:

hashValuesAndBucketSizesMod:do:

→ 3

Evaluation context

4 ↓

Hash buckets

|

Hierarchy issues

AbstractHashAnalysisDataSet

AbstractNumberDataSet

AbstractGbxLargeIntegerOopDataSet

...

FractionByProductFloatingPointDataSet

...

SmallFractionDataSet

MediumFractionDataSet

LargeFractionDataSet

...

Hash Analysis Tool

Data set hierarchy

- AbstractHashAnalysisDataSet
 - AbstractNumberDataSet
 - AbstractGbxLargeIntegerOopDataSet
 - BlockedGbxLargeIntegerOopDataSet
 - DenseGbxLargeIntegerOopDataSet
 - MixedGbxLargeIntegerOopDataSet
 - SparseGbxLargeIntegerOopDataSet
 - FractionByProductFloatingPointDataSet
 - SmallDoubleDataSet
 - MediumDoubleDataSet
 - LargeDoubleDataSet
 - SmallFloatDataSet
 - MediumFloatDataSet
 - LargeFloatDataSet
 - SmallScaledDoubleDataSet
 - MediumScaledDoubleDataSet
 - LargeScaledDoubleDataSet
 - SmallScaledFloatDataSet
 - MediumScaledFloatDataSet
 - LargeScaledFloatDataSet
 - SmallFractionDataSet
 - MediumFractionDataSet
 - LargeFractionDataSet

Run details

Data set:

Hash function:

Ran:

Hash effectiveness

Number of objects:

Number of hash values:

Collision rate:

Hash quality %:

Normalized hash bucket χ^2 :

Avg normalized hash bucket $\chi^2 \bmod P$:

Normalized hash bucket $\chi^2 \bmod P$ values

Value of P	Load factor	Normalized hash bucket $\chi^2 \bmod P$

Hash Analysis

Analyze hash

Distribution analysis

Time analysis

Compare hashes...

Hash Debugging

Verify hash

Profile hash

Inspect buckets

Inspect large buckets

Profile HAT

Use thorough buckets

Hash efficiency

Minimum hash time (mus):

Harmonic hash time mean (mus):

Arithmetic hash time mean (mus):

Approximate maximum hash time (mus):

HashAnalysisToolUI >> newDataSetList

Browser Edit Find View Package Class Protocol Method Tools Help

Package Hierarchy PartialHierarchy Instance Class Shared Variable Instance Variable

BitStreams * +
ComplexConditions *
Hash Analysis Tool *
Hash Analysis Tool
Hash Analysis Tool
Hash Analysis Tool
Hash Analysis Tool
Hash Analysis Tool
Hash Analysis Tool
Hash Analysis Tool
Hash Analysis Tool
ImageResizing * +

PluggableAllInstancesDataS
PluggableHashedCollection
AbstractHashAnalysisResul
HashAnalysisDistributionR
HashAnalysisTimeResults
HashAnalysisComparisonUI
HashAnalysisDistributionRe
HashAnalysisRunUI
HashAnalysisToolButtonsUI
HashAnalysisToolUI
HashFunctionAnalysisWrap

accessing
accessing - selections
actions
actions - buttons
aspects
change messages
constants
data sets
hash functions
interface closing
interface opening

basicSortedSubclassesOf:
basicVisibleSortedSubclass
newDataSetList
sortedSubclassesOf:

Source Rewrite Code Critic Bytecode

newDataSetList

```
^TreeModel  
on: self dataSetRoot  
displayRoot: true  
childrenWith: [:aClass | self sortedSubclassesOf: aClass]  
testHasChildrenWith: [:aClass | aClass subclasses notEmpty]
```

Method: #newDataSetList (data sets) Parcel: none Package: Hash Analysis Tool UI

Straightforward

Trivial

Sorting datasets by type

“root abstract class of some sort”

“ByteArray super class”

ByteArrayDataSet

RepetitiveByteArrayDataSet

“ByteString super class”

AbstractWordListDataSet

DutchWordListDataSet

EnglishWordListDataSet

EnglishWordListMinus64DataSet

GermanWordListDataSet

...

Sorting datasets by type

- Abstract root class does not exist.
- “type” super classes do not exist.
- Hierarchy shape changes needed, but cannot do that on the real classes...

Ad-hoc datasets

- Zero code for new data sets.
- Ad-hoc data set class a subclass of the abstract data set class, but must not be visible...
- Hierarchy shape changes needed, but cannot do that on the real classes...

Adopting hashed collections

- What is the abstract root class?
- How are hashed collections represented as data sets, which are classes?
- Hierarchy shape... ???...

Solution

- Object
- AbstractDataSetMetaclass**
- AbstractDataSetRootMetaclass
 - DataSetTypeRootMetaclass
 - PluggableDataSetRootMetaclass
- AbstractDataSetSubMetaclass
 - PluggableDataSetSubMetaclass
 - PluggableAllGeneralInstancesDataSetMetaclass
 - PluggableAllInstancesDataSetMetaclass
 - PluggableHashedCollectionDataSetMetaclass
 - PluggableDataSetTypeSubMetaclass
- LeafDataSetMetaclass

- Hash Analysis Tool UI +**

- accessing
- accessing - data sets
- accessing - metaclass
- private - accessing

- allDataSets
- allVisibleDataSets
- expensiveObjectToHash
- hashAnalysisTool
- hashAnalysisTool:
- isCompatibleWith:
- matchesDataSet:
- name
- new**
- objectKinds
- objectsToAnalyze
- printOn:

```
new

^self
```


<ul style="list-style-type: none">ObjectAbstractDataSetMetaclassAbstractDataSetRootMetaclassDataSetTypeRootMetaclassPluggableDataSetRootMetaclassAbstractDataSetSubMetaclassPluggableDataSetSubMetaclassPluggableAllGeneralInstancesDataSetMetaclassPluggableAllInstancesDataSetMetaclassPluggableHashedCollectionDataSetMetaclassPluggableDataSetTypeSubMetaclassLeafDataSetMetaclass	Hash Analysis Tool UI +	<ul style="list-style-type: none">accessingaccessing - data setsaccessing - metaclassprivate - accessing	<ul style="list-style-type: none">allDataSetsallVisibleDataSetsexpensiveObjectToHashhashAnalysisToolhashAnalysisTool:isCompatibleWith:matchesDataSet:namenewobjectKindsobjectsToAnalyzeprintOn:
--	--------------------------------	---	---

```
new

^self
```

DataSetTypeRootMetaclass >> subclasses

Browser Edit Find View Package Class Protocol Method Tools Help

Package Hierarchy PartialHierarchy Instance Class Shared Variable Instance Variable

Object
AbstractDataSetMetaclass
AbstractDataSetRootMetaclass
DataSetTypeRootMetaclass
PluggableDataSetRootMetaclass
AbstractDataSetSubMetaclass
PluggableDataSetSubMetaclass
PluggableAllGeneralInstancesDataSetMetaclass
PluggableAllInstancesDataSetMetaclass
PluggableHashedCollectionDataSetMetaclass
PluggableDataSetTypeSubMetaclass
LeafDataSetMetaclass

Hash Analysis Tool UI +

accessing - metaclass

allObjectKindClasses
subclasses

Source Rewrite Code Critic Bytecode

subclasses

```
^self allObjectKindClasses collect:  
  [:each |  
    PluggableDataSetTypeSubMetaclass  
    forHashAnalysisTool: self hashAnalysisTool  
    with: each  
  ]
```

Method: #subclasses (accessing - metaclass) Parcel: none Package: Hash Analysis Tool UI

Arbitrary class hierarchy manipulation

Object

- AbstractDataSetMetaclass
 - AbstractDataSetRootMetaclass
 - DataSetTypeRootMetaclass
 - PluggableDataSetRootMetaclass**
 - AbstractDataSetSubMetaclass
 - PluggableDataSetSubMetaclass
 - PluggableAllGeneralInstancesDataSetMetaclass
 - PluggableAllInstancesDataSetMetaclass
 - PluggableHashedCollectionDataSetMetaclass
 - PluggableDataSetTypeSubMetaclass
 - LeafDataSetMetaclass

Hash Analysis Tool UI +

accessing - metaclass

subclasses

```
subclasses

^Array
  with: (PluggableAllInstancesDataSetMetaclass forHashAnalysisTool: self hashAnalysisTool)
  with: (PluggableAllGeneralInstancesDataSetMetaclass forHashAnalysisTool: self hashAnalysisTool)
  with: (PluggableHashedCollectionDataSetMetaclass forHashAnalysisTool: self hashAnalysisTool)
```

- Object
- AbstractDataSetMetaClass**
- AbstractDataSetRootMetaClass
- DataSetTypeRootMetaClass
- PluggableDataSetRootMetaClass
- AbstractDataSetSubMetaClass
- PluggableDataSetSubMetaClass**
- PluggableAllGeneralInstancesDataSetMetaClass
- PluggableAllInstancesDataSetMetaClass
- PluggableHashedCollectionDataSetMetaClass
- PluggableDataSetTypeSubMetaClass
- LeafDataSetMetaClass

Hash Analysis Tool UI +

accessing - metaclass

- △ browse
- ▼ **classToImpersonate**
- leafMetaClassWith:
- △ sortLabel
- △ subclasses

```

classToImpersonate

self subclassResponsibility

```

- Object
- AbstractDataSetMetaClass**
 - AbstractDataSetRootMetaClass
 - DataSetTypeRootMetaClass
 - PluggableDataSetRootMetaClass
- AbstractDataSetSubMetaClass
- PluggableDataSetSubMetaClass**
 - PluggableAllGeneralInstancesDataSetMetaClass
 - PluggableAllInstancesDataSetMetaClass
 - PluggableHashedCollectionDataSetMetaClass
 - PluggableDataSetTypeSubMetaClass
- LeafDataSetMetaClass

Hash Analysis Tool UI +

accessing - metaclass

- browse**
- classToImpersonate
- leafMetaClassWith:
- sortLabel
- subclasses

```

browse

^self classToImpersonate browse

```

Package Hierarchy PartialHierarchy Instance Class Shared Variable Instance Variable

- Object
 - AbstractDataSetMetaClass**
 - AbstractDataSetRootMetaClass
 - DataSetTypeRootMetaClass
 - PluggableDataSetRootMetaClass
 - AbstractDataSetSubMetaClass
 - PluggableDataSetSubMetaClass**
 - PluggableAllGeneralInstancesDataSetMetaClass
 - PluggableAllInstancesDataSetMetaClass
 - PluggableHashedCollectionDataSetMetaClass
 - PluggableDataSetTypeSubMetaClass
 - LeafDataSetMetaClass

Hash Analysis Tool UI +

accessing - metaclass

- browse
- classToImpersonate
- leafMetaClassWith:
- sortLabel
- subclasses**

```

subclasses

| adhocDataSets |
adhocDataSets := self hashAnalysisTool adhocDataSets
  select: [:any | any isKindOf: self classToImpersonate].
^adhocDataSets collect: [:each | self leafMetaClassWith: each]

```

Object

- AbstractDataSetMetaClass
 - AbstractDataSetRootMetaClass
 - DataSetTypeRootMetaClass
 - PluggableDataSetRootMetaClass
- AbstractDataSetSubMetaClass
 - PluggableDataSetSubMetaClass**
 - PluggableAllGeneralInstancesDataSetMetaClass
 - PluggableAllInstancesDataSetMetaClass
 - PluggableHashedCollectionDataSetMetaClass
 - PluggableDataSetTypeSubMetaClass
- LeafDataSetMetaClass

Hash Analysis Tool UI +

accessing - metaclass

- browse
- classToImpersonate
- leafMetaClassWith:**
- sortLabel
- subclasses

```
leafMetaClassWith: aDataSet

^LeafDataSetMetaClass new
  dataSet: aDataSet;
  yourself
```

- Object
- AbstractDataSetMetaclass**
- AbstractDataSetRootMetaclass
- DataSetTypeRootMetaclass
- PluggableDataSetRootMetaclass
- AbstractDataSetSubMetaclass
- PluggableDataSetSubMetaclass
- PluggableAllGeneralInstancesDataSetMetaclass
- PluggableAllInstancesDataSetMetaclass
- PluggableHashedCollectionDataSetMetaclass
- PluggableDataSetTypeSubMetaclass
- LeafDataSetMetaclass**

- Hash Analysis Tool UI +**

- accessing
- accessing - data sets
- accessing - metaclass
- private - accessing

- △ browse
- dataSet
- dataSet:
- △ isCompatibleWith:
- △ matchesDataSet:
- △ name
- △ new**
- △ sortLabel
- △ subclasses

```
new

^self dataSet
```


Results

Hash Analysis Tool

Data set type hierarchy

- DataSetTypeRootMetaclass
 - AnnotatedMethod
 - KindOfAnnotatedMethodDataSet
 - ByteArray
 - ByteArrayDataSet
 - RepetitiveByteArraysDataSet
 - ByteString
 - AbstractWordListDataSet
 - DutchWordListDataSet
 - EnglishWordListDataSet
 - EnglishWordListMinus64DataSet
 - GermanWordListDataSet
 - ItalianWordListDataSet
 - NorwegianWordListDataSet
 - SwedishWordListDataSet
 - ByteStringDataSet
 - FourCCDataSet
 - ByteSymbol
 - ByteSymbolDataSet
 - ColorValue
 - VisualWorksColorValueDataSet
 - Date
 - DateDataSet

Run details

Data set:

Hash function:

Ran:

Hash effectiveness

Number of objects:

Number of hash values:

Collision rate:

Hash quality %:

Normalized hash bucket χ^2 :

Avg normalized hash bucket $\chi^2 \bmod P$:

Normalized hash bucket $\chi^2 \bmod P$ values

Value of P	Load factor	Normalized hash bucket $\chi^2 \bmod P$

Hash Analysis

Analyze hash

Distribution analysis

Time analysis

Compare hashes...

Hash Debugging

Verify hash

Profile hash

Inspect buckets

Inspect large buckets

Profile HAT

Use thorough buckets

Hash efficiency

Minimum hash time (mus):

Harmonic hash time mean (mus):

Arithmetic hash time mean (mus):

Approximate maximum hash time (mus):

Hash Analysis Tool

Ad-hoc data sets

- PluggableDataSetRootMetaclass
 - PluggableAllGeneralInstancesDataSetMetaclass
 - PluggableAllInstancesDataSetMetaclass
 - PluggableHashedCollectionDataSetMetaclass

Run details

Data set:

Hash function:

Ran:

Hash effectiveness

Number of objects:

Number of hash values:

Collision rate:

Hash quality %:

Normalized hash bucket χ^2 :

Avg normalized hash bucket $\chi^2 \bmod P$:

Normalized hash bucket $\chi^2 \bmod P$ values

Value of P	Load factor	Normalized hash bucket $\chi^2 \bmod P$

Hash Analysis

Analyze hash

Distribution analysis

Time analysis

Compare hashes...

Hash Debugging

Verify hash

Profile hash

Inspect buckets

Inspect large buckets

Profile HAT Use thorough buckets

Hash efficiency

Minimum hash time (mus):

Harmonic hash time mean (mus):

Arithmetic hash time mean (mus):

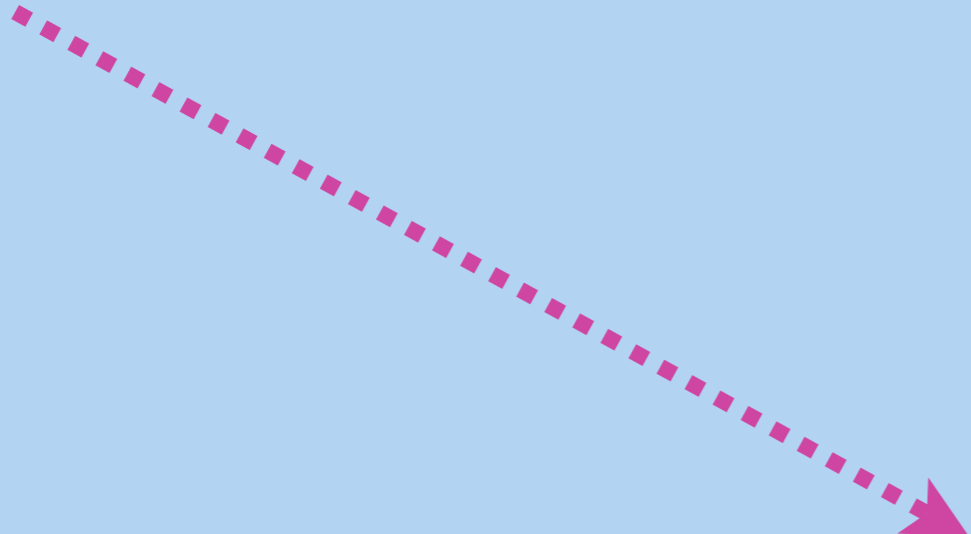
Approximate maximum hash time (mus):

Workspace [-] [□] [×]

Page Edit Smalltalk Options Help

Page 1 Variables

Smalltalk inspect



Text or Smalltalk code

a Namespace [-] [□] [×]

Object Edit Go History Explore Tools Help

- Refresh F5
- Browse Shift+Ctrl+B
- Inspect
- Inspect Owners
- Inspect Reference Paths
- Adopt in Hash Analysis Tool...**

MemoryGamePlayerStrategy .
PongMultiballStrategy ... Simp

AbstractBenchmarkTestCase
AbstractBitStream
AbstractBitStreamTests
AbstractBretMulveySBoxHash
AbstractChallengeMemoryGan
AbstractChallengeMemoryGan
AbstractCollectionDataSet
AbstractComplexFibonacciCal
AbstractComplexModFibonacc
AbstractCryptographicHashFu

Value at Key Key

Hash Analysis Tool

Ad-hoc data sets

- PluggableDataSetRootMetaclass
 - PluggableAllGeneralInstancesDataSetMetaclass
 - PluggableAllInstancesDataSetMetaclass
 - PluggableHashedCollectionDataSetMetaclass
 - PluggableDictionaryDataSet(Smalltalk)

Run details

Data set:

Hash function:

Ran:

Hash effectiveness

Number of objects:

Number of hash values:

Collision rate:

Hash quality %:

Normalized hash bucket χ^2 :

Avg normalized hash bucket $\chi^2 \bmod P$:

Normalized hash bucket $\chi^2 \bmod P$ values

Value of P	Load factor	Normalized hash bucket $\chi^2 \bmod P$

Hash Analysis

Analyze hash

Distribution analysis

Time analysis

Compare hashes...

Hash Debugging

Verify hash

Profile hash

Inspect buckets

Inspect large buckets

Profile HAT

Use thorough buckets

Hash efficiency

Minimum hash time (mus):

Harmonic hash time mean (mus):

Arithmetic hash time mean (mus):

Approximate maximum hash time (mus):

Hash Analysis Tool

Ad-hoc data sets

- PluggableDataSetRootMetaclass
 - PluggableAllGeneralInstancesDataSetMetaclass
 - PluggableAllInstancesDataSetMetaclass
 - PluggableHashedCollectionDataSetMetaclass
 - PluggableDictionaryDataSet(Smalltalk)

- Add adhoc dataset...
- Browse...
- Refresh

Run details

Data set:

Hash function:

Ran:

Hash effectiveness

Number of objects:

Number of hash values:

Collision rate:

Hash quality %:

Normalized hash bucket χ^2 :

Avg normalized hash bucket $\chi^2 \bmod P$:

Normalized hash bucket $\chi^2 \bmod P$ values

Value of P	Load factor	Normalized hash bucket $\chi^2 \bmod P$

Hash Analysis

Analyze hash

Distribution analysis

Time analysis

Compare hashes...

Hash Debugging

Verify hash

Profile hash

Inspect buckets

Inspect large buckets

Profile HAT

Use thorough buckets

Hash efficiency

Minimum hash time (mus):

Harmonic hash time mean (mus):

Arithmetic hash time mean (mus):

Approximate maximum hash time (mus):

Hash Analysis Tool

Ad-hoc data sets

- PluggableDataSetRootMetaclass
 - PluggableAllGeneralInstancesDataSetMetaclass
 - PluggableAllInstancesDataSetMetaclass
 - allInstancesOf(Association)
 - PluggableHashedCollectionDataSetMetaclass
 - PluggableDictionaryDataSet(Smalltalk)

Run details

Data set:

Hash function:

Ran:

Hash effectiveness

Number of objects:

Number of hash values:

Collision rate:

Hash quality %:

Normalized hash bucket χ^2 :

Avg normalized hash bucket $\chi^2 \bmod P$:

Normalized hash bucket $\chi^2 \bmod P$ values

Value of P	Load factor	Normalized hash bucket $\chi^2 \bmod P$

Hash Analysis

Analyze hash

Distribution analysis

Time analysis

Compare hashes...

Hash Debugging

Verify hash

Profile hash

Inspect buckets

Inspect large buckets

Profile HAT

Use thorough buckets

Hash efficiency

Minimum hash time (mus):

Harmonic hash time mean (mus):

Arithmetic hash time mean (mus):

Approximate maximum hash time (mus):

Hash Analysis Tool

Ad-hoc data sets

PluggableDataSetRootMetaclass
 PluggableAllGeneralInstancesDataSetMetaclass
 PluggableAllInstancesDataSetMetaclass
 allInstancesOf(Association)
 PluggableHashedCollectionDataSetMetaclass
 PluggableDictionaryDataSet(Smalltalk)

Hash(Object)

Hash Analysis

Analyze hash

Distribution analysis

Time analysis

Compare hashes...

Profile HAT

Hash Debugging

Verify hash

Profile hash

Inspect buckets

Inspect large buckets

Use thorough buckets

Run details

Data set: allInstancesOf(Association)

Hash function: Hash(Object)

Ran: 08/09/2008 5:44:35.367

Hash effectiveness

Number of objects: 27,942

Number of hash values: 5,752

Collision rate: 4.857789

Hash quality %: 20.585499

Normalized hash bucket χ^2 : 197.847684

Avg normalized hash bucket $\chi^2 \bmod P$: 198.094737

Normalized hash bucket $\chi^2 \bmod P$ values

Value of P	Load factor	Normalized hash bucket $\chi^2 \bmod P$
29,147	0.958658	198.322275
30,971	0.902199	198.096521
32,789	0.852176	198.033892

Hash efficiency

Minimum hash time (mus):

Harmonic hash time mean (mus):

Arithmetic hash time mean (mus):

Approximate maximum hash time (mus):

Hash Analysis Tool

Ad-hoc data sets

- PluggableDataSetRootMetaclass
 - PluggableAllGeneralInstancesDataSetMetaclass
 - PluggableAllInstancesDataSetMetaclass
 - allInstancesOf(Association)
 - PluggableHashedCollectionDataSetMetaclass
 - PluggableDictionaryDataSet(Smalltalk)

Run details

Data set: PluggableDictionaryDataSet(Smalltalk)
Hash function: Hash(Object)
Ran: 08/09/2008 5:45:14.534

Hash effectiveness

Number of objects: 1,358
Number of hash values: 1,304
Collision rate: 1.041411
Hash quality %: 96.023564
Normalized hash bucket χ^2 : 0.041237
Avg normalized hash bucket $\chi^2 \bmod P$: 0.147559

Normalized hash bucket $\chi^2 \bmod P$ values

Value of P	Load factor	Normalized hash bucket $\chi^2 \bmod P$
2,053	0.661471	0.368925
2,311	0.587624	0.324742
2,579	0.526561	0.311487

Hash Analysis

Analyze hash

Distribution analysis

Time analysis

Compare hashes...

Hash Debugging

Verify hash

Profile hash

Inspect buckets

Inspect large buckets

Profile HAT

Use thorough buckets

Hash efficiency

Minimum hash time (mus)

Harmonic hash time mean (mus)

Arithmetic hash time mean (mus)

Approximate maximum hash time (mus)

SUnit VM

- Refactored SUnit
 - SUnit Based Validation
 - SUnit Benchmarks

Implementors of suiteClass - AbstractTestCase class>>suiteClass

Browser Edit Find View Class Method Tools Help

Method Class

- △ AbstractBenchmarkTestCase class suiteClass {private}
- ▼ AbstractTestCase class suiteClass {building suites}
- △ AbstractValidator class suiteClass {private}

Source Rewrite Code Critic Bytecode

suiteClass

^TestSuite

Method: #suiteClass (building suites)

References to: suiteClass - AbstractTestCase class>>buildSuite

Browser Edit Find View Class Method Tools Help

Method Class

- △ AbstractBenchmarkTestCase class selector: {instance creation}
- AbstractTestCase class buildHierarchySuite {building suites}
- AbstractTestCase class buildSingleSuite {building suites}
- ▼ AbstractTestCase class buildSuite {building suites}
- AbstractTestCase class buildSuiteFromMethods: {building suites}

Source Rewrite Code Critic Bytecode

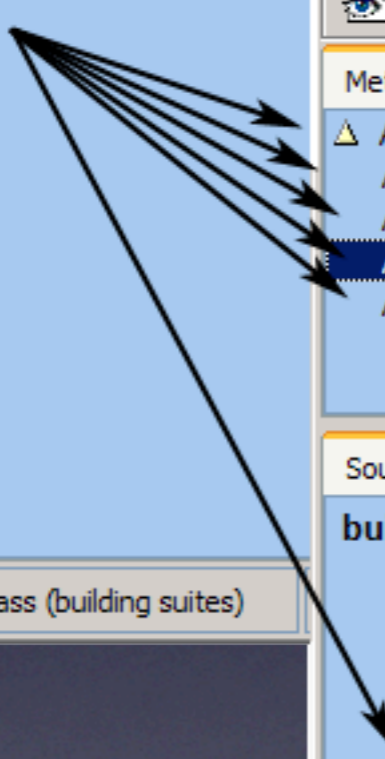
buildSuite

```

| suite |
self isAbstract iffFalse: [^self buildSuiteFromSelectors].
suite := self suiteClass named: self name asString.
^self
collectConcreteClassesWithin: self allSubclasses
with: [:aClass | aClass buildSuiteFromSelectors]
andAddNonEmptyResultsTo: suite

```

Method: #buildSuite (building suites) **Parcel:** none **Package:** SUnit



Implementors of selectorPattern - AbstractTestCase class>>selectorPattern

Browser Edit Find View Class Method Tools Help

Method Class

- AbstractBenchmarkTestCase class selectorPattern {private}
- AbstractTestCase class selectorPattern {accessing}**
- AbstractValidator class selectorPattern {private}

Source Rewrite Code Critic Bytecode

selectorPattern

^test*

Method: #selectorPattern (accessing)

References to: selectorPattern - AbstractTestCase class>>testSelectors

Browser Edit Find View Class Method Tools Help

Method Class

- AbstractTestCase class allTestSelectors {accessing}
- AbstractTestCase class testSelectors {accessing}**

Source Rewrite Code Critic Bytecode

testSelectors

```
^self sunitSelectors select:  
[:each | self selectorPattern sunitMatch: each]
```

Method: #testSelectors (accessing) **Parcel:** none **Package:** SUnit

TestResult>>runCase:

Browser Edit Find View Class Method Tools Help

Method Class

runCase:

Source Rewrite Code Critic Bytecode

```
runCase: aTestCase

[
  aTestCase runCase.
  self recordPass: aTestCase
]
on: self exceptionsToTrap
do:
  [:ex |
    ex
      occurredForTestCase: aTestCase
      inTheContextOf: self
  ]
```

Method: #runCase: (running) Parcel: SUnit Package: SUnit

Implementors of occurredForTestCase:inTheContextOf: - Error>>occurredForTestCase:inTheCont...

Browser Edit Find View Class Method Tools Help

Find:

Method Class

- △ BenchmarkMeasurementNotification occurredForTestCase:inTheContextOf: {SUnit}
- ▼ **Error occurredForTestCase:inTheContextOf: {SUnit}**
- ▽ GenericException occurredForTestCase:inTheContextOf: {SUnit}
- △ SUnitNoticeNotification occurredForTestCase:inTheContextOf: {SUnit}
- △ TestFailure occurredForTestCase:inTheContextOf: {SUnit}
- △ TestNoticeNotification occurredForTestCase:inTheContextOf: {SUnit}
- △ ValidationFailureException occurredForTestCase:inTheContextOf: {SUnit}

Source Rewrite Code Critic Bytecode

```
occurredForTestCase: aTestCase
inTheContextOf: aTestResult

    aTestResult recordError: aTestCase.
    self return: false
```

Method: #occurredForTestCase:inTheCo Parcel: none Package: SUnitPreload

SUnit Based Validation (brief)

AbstractValidator

Browser Edit Find View Package Class Protocol Method Tools Help

Find:

Package Hierarchy PartialHierarchy Instance Class Shared Variable Instance Variable

- BitStreams +
- ComplexConditions +
- Hash Analysis Tool
- ImageResizing +
- LexiSort
- Match +
- ReferenceFinder +
- SUnit Extensions +
 - ReThink Test Runner
 - SUnitBenchmarks +
 - SUnitValidation +**

- AbstractTestResultRecordingP
- DefaultTestResultRecordingP
- LogFileTestResultRecordingP
- AbstractValidator**
- AbstractDomainValidator
- AbstractUIValidator
- ValidationFailure
- ValidationFailureException
- ValidationResult
- ValidationSuite

- accessors
- converting
- exceptions
- message patterns
- object
- setup
- validation helpers
- validation helpers - adaptor values
- validation helpers - values
- private

- value
- valuesDate
- valuesDefined
- valuesNonEmptyString
- valuesNonFutureDate
- valuesNonFutureTimestamp
- valuesNonOldDate
- valuesNonOldTimestamp
- valuesNotZero
- valuesPositiveInteger
- valuesTimestamp

Source **Comment** Hierarchy Diagram Rewrite Code Critic

```
Smalltalk defineClass: #AbstractValidator
  superclass: #{XProgramming.SUnit.AbstractTestCase}
  indexedType: #none
  private: false
  instanceVariableNames: 'object originalObject prettyPrint reportFailureFor aspect '
  classInstanceVariableNames: ''
  imports: ''
  category: 'SUnitValidation'
```

Class: AbstractValidator **Parcel:** none **Package:** SUnitValidation

AbstractValidator>>failValidationBecause:

Browser Edit Find View Package Class Protocol Method Tools Help

Package Hierarchy PartialHierarchy Instance Class Shared Variable Instance Variable

BitStreams +
 ComplexConditions +
 Hash Analysis Tool
 ImageResizing +
 LexiSort
 Match +
 ReferenceFinder +
 SUnit Extensions +
 ReThink Test Runner
 SUnitBenchmarks +
 SUnitValidation +

AbstractTestResultRecordingP
 DefaultTestResultRecordingP
 LogFileTestResultRecordingP
AbstractValidator
 AbstractDomainValidator
 AbstractUIValidator
 ValidationFailure
 ValidationFailureException
 ValidationResult
 ValidationSuite

accessors
 converting
exceptions
 message patterns
 object
 setup
 validation helpers
 validation helpers - adaptor values
 validation helpers - values
 private

failValidationBecause:

Source Rewrite Code Critic Bytecode

failValidationBecause: aString

```

| exception validationFailure |
validationFailure := ValidationFailure new
  object: self object;
  aspect: self aspectForReporting;
  description: aString;
  yourself.
exception := ValidationFailureException new.
exception validationFailure: validationFailure.
exception raise

```

Method: #failValidationBecause: (exceptions) Parcel: none Package: SUnitValidation

SUnit Benchmarks



VW / VW+ VW+c VW+e Dc De [CI] VW / VW+ [CI] VW+c [CI] VW+e [CI] Dc [CI] De Sc Variables

```
Time millisecondsToRun: [1000000 timesRepeat: ['abc' match: 'abc']] 632  
Time millisecondsToRun: [1000000 timesRepeat: ['abc' inlinedCaseInsensitiveMatches: 'abc']] 531  
  
Time millisecondsToRun: [1000000 timesRepeat: ['ab*de#fg' match: 'abczdexfg']] 2198  
Time millisecondsToRun: [1000000 timesRepeat: ['ab*de#fg' inlinedCaseInsensitiveMatches: 'abczdexfg']] 1179  
  
Time millisecondsToRun: [1000000 timesRepeat: ['ab*' match: 'abczdexfg']] 537  
Time millisecondsToRun: [1000000 timesRepeat: ['ab*' inlinedCaseInsensitiveMatches: 'abczdexfg']] 529  
  
Time millisecondsToRun: [1000000 timesRepeat: ['ab#' match: 'abc']] 664  
Time millisecondsToRun: [1000000 timesRepeat: ['ab#' inlinedCaseInsensitiveMatches: 'abc']] 540  
  
Time millisecondsToRun: [1000000 timesRepeat: ['*ab*' match: 'xyzabczdexfg']] 1750  
Time millisecondsToRun: [1000000 timesRepeat: ['*ab*' inlinedCaseInsensitiveMatches: 'xyzabczdexfg']] 1803  
  
Time millisecondsToRun: [1000000 timesRepeat: ['*ab#' match: 'xyzabc']] 1984  
Time millisecondsToRun: [1000000 timesRepeat: ['*ab#' inlinedCaseInsensitiveMatches: 'xyzabc']] 661  
  
Time millisecondsToRun: [1000000 timesRepeat: ['*fg' match: 'abczdexfg']] 3326  
Time millisecondsToRun: [1000000 timesRepeat: ['*fg' inlinedCaseInsensitiveMatches: 'abczdexfg']] 513  
  
Time millisecondsToRun: [1000000 timesRepeat: ['#ab' match: 'xab']] 656  
Time millisecondsToRun: [1000000 timesRepeat: ['#ab' inlinedCaseInsensitiveMatches: 'xab']] 532  
  
632+2198+537+664+1750+1984+3326+656 11747  
531+1179+529+540+1803+661+513+532 6288  
  
Time millisecondsToRun: [1000000 timesRepeat: ['ab*de#fg' match: 'abczdexfgz']] 4169  
Time millisecondsToRun: [1000000 timesRepeat: ['ab*de#fg' inlinedCaseInsensitiveMatches: 'abczdexfgz']] 507
```



VW / VW+ VW+c VW+e Dc De [CI] VW / VW+ [CI] VW+c [CI] VW+e [CI] Dc [CI] De Sc Variables

```

Time millisecondsToRun: [1000000 timesRepeat: ['abc' match: 'abc']] 632
Time millisecondsToRun: [1000000 timesRepeat: ['abc' inlinedCaseInsensitiveMatches: 'abc']] 531

Time millisecondsToRun: [1000000 timesRepeat: ['ab*de#fg' match: 'abczdexfg']] 2198
Time millisecondsToRun: [1000000 timesRepeat: ['ab*de#fg' inlinedCaseInsensitiveMatches: 'abczdexfg']] 1179

Time millisecondsToRun: [1000000 timesRepeat: ['ab*' match: 'abczdexfg']] 537
Time millisecondsToRun: [1000000 timesRepeat: ['ab*' inlinedCaseInsensitiveMatches: 'abczdexfg']] 529

Time millisecondsToRun: [1000000 timesRepeat: ['ab#' match: 'abc']] 664
Time millisecondsToRun: [1000000 timesRepeat: ['ab#' inlinedCaseInsensitiveMatches: 'abc']] 540

Time millisecondsToRun: [1000000 timesRepeat: ['*ab*' match: 'xyzabczdexfg']] 1750
Time millisecondsToRun: [1000000 timesRepeat: ['*ab*' inlinedCaseInsensitiveMatches: 'xyzabczdexfg']] 1803

Time millisecondsToRun: [1000000 timesRepeat: ['*ab#' match: 'xyzabc']] 1984
Time millisecondsToRun: [1000000 timesRepeat: ['*ab#' inlinedCaseInsensitiveMatches: 'xyzabc']] 661

Time millisecondsToRun: [1000000 timesRepeat: ['*fg' match: 'abczdexfg']] 3326
Time millisecondsToRun: [1000000 timesRepeat: ['*fg' inlinedCaseInsensitiveMatches: 'abczdexfg']] 513

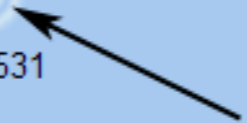
Time millisecondsToRun: [1000000 timesRepeat: ['#ab' match: 'xab']] 656
Time millisecondsToRun: [1000000 timesRepeat: ['#ab' inlinedCaseInsensitiveMatches: 'xab']] 532

632+2198+537+664+1750+1984+3326+656 11747
531+1179+529+540+1803+661+513+532 6288

Time millisecondsToRun: [1000000 timesRepeat: ['ab*de#fg' match: 'abczdexfgz']] 4169
Time millisecondsToRun: [1000000 timesRepeat: ['ab*de#fg' inlinedCaseInsensitiveMatches: 'abczdexfgz']] 507

```

7 x 56 = 392



duh: interj.

Used to express disdain for something deemed stupid or obvious, especially a self-evident remark.

SUnitBenchmarks - AbstractBenchmarkTestCase

Browser Edit Find View Class Method Tools Help

Find:

AbstractBenchmarkTestCase

- AbstractTestResultRecordingPlugin
- BenchmarkExpectation
- ⚡ BenchmarkExpectationNotification
- BenchmarkMeasurement
- ⚡ BenchmarkMeasurementNotification
- BenchmarkResult
- BenchmarkSuite
- DefaultTestResultRecordingPlugin
- LogFileTestResultRecordingPlugin
- ◆ ReThinkAbstractBenchmarkView
- ◆ ReThinkBenchmarkExpectation
- ◆ ReThinkBenchmarkMeasurement
- ◆ ReThinkBenchmarkRunner

Instance Class Shared Variable Instance Variable

accessing
comparing
constants
copying
exceptions
pretty print
printing
running
setup
stop watch
private - accessing
private - limits
private - measuring expectations

<=
flushNativeCodeCache
flushNewSpace
itTakesLessThan:messageSend
microsecondsPerSecond
microsecondThreshold
musToRun:times:
netMusToRun:times:
prettyPrintForMeasurement
prettyPrintForMeasurement:
▼ prettyPrintForSetUp
prettyPrintForSetUp:
▲ printOn:

Source **Comment** Hierarchy Diagram Rewrite Code Critic

```
Smalltalk defineClass: #AbstractBenchmarkTestCase
  superclass: #{XProgramming.SUnit.AbstractTestCase}
  indexedType: #none
  private: false
  instanceVariableNames: 'selectorForSetUp prettyPrintForSetUp prettyPrintForMeasurement '
  classInstanceVariableNames: ''
  imports: ''
  category: 'SUnitBenchmarks'
```

Class: AbstractBenchmarkTestCase **Parcel:** none **Package:** SUnitBenchmarks

SUnitBenchmarks - AbstractBenchmarkTestCase >> reportIterationsPerSecond:

Browser Edit Find View Class Method Tools Help

Find:

AbstractBenchmarkTestCase
 AbstractTestResultRecordingPlu
 BenchmarkExpectation
 ⚡ BenchmarkExpectationNotifica
 BenchmarkMeasurement
 ⚡ BenchmarkMeasurementNotifi
 BenchmarkResult
 BenchmarkSuite
 DefaultTestResultRecordingPlug
 LogFileTestResultRecordingPlug

Instance Class Shared Variable Instance Variable

reportIterationsPerSecond:

comparing
 constants
 copying
 exceptions
 pretty print
 printing
 running
 setup

Source Rewrite Code Critic Bytecode

reportIterationsPerSecond: aNumber

```

| notification measurement |
measurement := BenchmarkMeasurement new
  benchmarkTestCase: self class;
  setup: self selectorForSetUp;
  measurement: self selector;
  iterationsPerSecond: aNumber;
  prettyPrintForSetUp: self prettyPrintForSetUp;
  prettyPrintForMeasurement: self prettyPrintForMeasurement;
  yourself.
notification := BenchmarkMeasurementNotification new.
notification benchmarkMeasurement: measurement.
notification raise

```

Method: #reportIterationsPerSecond: (except) Parcel: none Package: SUnitBenchmarks

SUnitBenchmarks - AbstractBenchmarkTestCase>>reportExpectedIterationsPerSecond:andActualIteratio...

Browser Edit Find View Class Method Tools Help

Find:

AbstractBenchmarkTestCase

- AbstractTestResultRecordingPlus
- BenchmarkExpectation
- BenchmarkExpectationNotification
- BenchmarkMeasurement
- BenchmarkMeasurementNotification
- BenchmarkResult

Instance Class Shared Variable Instance Variable

- stop watch
- private - accessing
- private - limits
- private - measuring expectations
- private - memory spaces
- private - stop watch

reportExpectedIterationsPerSecond
simpleMessageSend

Source Rewrite Code Critic Bytecode

reportExpectedIterationsPerSecond: expectedIterationsPerSecond
andActualIterationsPerSecond: actualIterationsPerSecond

```

| expectation notification |
expectation := BenchmarkExpectation new
  benchmarkTestCase: self class;
  setup: self selectorForSetUp;
  measurement: self selector;
  iterationsPerSecond: actualIterationsPerSecond;
  expectedIterationsPerSecond: expectedIterationsPerSecond;
  prettyPrintForSetUp: self prettyPrintForSetUp;
  prettyPrintForMeasurement: self prettyPrintForMeasurement;
  yourself.
notification := BenchmarkExpectationNotification new.
notification benchmarkMeasurement: expectation.
notification raise

```

Method: #reportExpectedIterationsPerSecond Parcel: none Package: SUnitBenchmarks

AbstractMatchBenchmarkTestCase

Browser Edit Find View Package Class Protocol Method Tools Help

Find:

Package Hierarchy PartialHierarchy Instance Class Shared Variable Instance Variable

Object
AbstractTestCase
AbstractBenchmarkTest
AbstractMatchBench
ClassBasedCharacte
CondensedCharacte
MatchBenchmark
MatchBenchmark
MatchBenchmark
EnumeratedCharac
MatchBenchmark
MatchBenchmark
MatchBenchmark
StringMatchBenchma
MatchBenchmarkTe
MatchBenchmarkTe

Match benchmarks

accessing
pretty print
setup
setup - mixed case A
setup - mixed case B
setup - mixed case C
setup - mixed case D
setup - same case A
setup - same case B
setup - same case C
setup - same case D

basicPattern:
basicString:
makeStringUppercase
pattern:
prettyPrintForSetUp
setUpMixedCaseA1
setUpMixedCaseA2
setUpMixedCaseA3
setUpMixedCaseA4
setUpMixedCaseA5
setUpMixedCaseA6
setUpMixedCaseA7
setUpMixedCaseA8
setUpMixedCaseB1
setUpMixedCaseB2

Source **Comment** Hierarchy Diagram Rewrite Code Critic

```
Smalltalk defineClass: #AbstractMatchBenchmarkTestCase  
  superclass: #{Smalltalk.AbstractBenchmarkTestCase}  
  indexedType: #none  
  private: false  
  instanceVariableNames: 'pattern string'  
  classInstanceVariableNames: ''  
  imports: ''  
  category: 'Match Benchmarks'
```

Class: AbstractMatchBenchmarkTestCase **Parcel:** none **Package:** Match benchmarks

AbstractMatchBenchmarkTestCase>>setUpSameCaseA1

Browser Edit Find View Package Class Protocol Method Tools Help

Package Hierarchy PartialHierarchy Instance Class Shared Variable Instance Variable

Object
AbstractTestCase
AbstractBenchmarkTest
AbstractMatchBenchmarkTest
ClassBasedCharacter
CondensedCharacter
MatchBenchmark
MatchBenchmark
MatchBenchmark
EnumeratedCharacter
MatchBenchmark
MatchBenchmark
MatchBenchmark
StringMatchBenchmark
MatchBenchmarkTe
MatchBenchmarkTe

Match benchmarks

accessing
pretty print
setup
setup - mixed case A
setup - mixed case B
setup - mixed case C
setup - mixed case D
setup - same case A
setup - same case B
setup - same case C
setup - same case D

setUpMixedCaseD3
setUpMixedCaseD4
setUpMixedCaseD5
setUpMixedCaseD6
setUpMixedCaseD7
setUpSameCaseA1
setUpSameCaseA2
setUpSameCaseA3
setUpSameCaseA4
setUpSameCaseA5
setUpSameCaseA6
setUpSameCaseA7
setUpSameCaseA8
setUpSameCaseB1
setUpSameCaseB2
setUpSameCaseB3

Source Rewrite Code Critic Bytecode

setUpSameCaseA1

```
self pattern: 'abc'.  
self string: 'abc'
```

Method: #setUpSameCaseA1 (setup - same cas) **Parcel:** none **Package:** Match benchmarks

Package	Hierarchy	PartialHierarchy	Instance	Class	Shared Variable	Instance Variable
Object	Match benchmarks		accessing			pattern:
AbstractTestCase			pretty print			prettyPrintForSetUp
AbstractBenchmarkTest			setup			setUpMixedCaseA1
AbstractMatchBenchmarkTestCase			setup - mixed case A			setUpMixedCaseA2
ClassBasedCharacter			setup - mixed case B			setUpMixedCaseA3
CondensedCharacter			setup - mixed case C			setUpMixedCaseA4
MatchBenchmark			setup - mixed case D			setUpMixedCaseA5
MatchBenchmark			setup - same case A			setUpMixedCaseA6
MatchBenchmark			setup - same case B			setUpMixedCaseA7
EnumeratedCharacter			setup - same case C			setUpMixedCaseA8
MatchBenchmark			setup - same case D			setUpMixedCaseB1
MatchBenchmark						setUpMixedCaseB2
MatchBenchmark						setUpMixedCaseB3
StringMatchBenchmark						setUpMixedCaseB4
MatchBenchmarkTe						setUpMixedCaseB5
MatchBenchmarkTe						setUpMixedCaseB6

```

setUpMixedCaseA1

  self setUpSameCaseA1.
  self makeStringUppercase
  
```

MatchBenchmarkTestCaseVW >> measureVW

Browser Edit Find View Package Class Protocol Method Tools Help

Find:

Package Hierarchy PartialHierarchy Instance Class Shared Variable Instance Variable

Object
 AbstractTestCase
 AbstractBenchmarkTest
AbstractMatchBench
 ClassBasedCharacter
 CondensedCharacter
 MatchBenchmark
 MatchBenchmark
 MatchBenchmark
 EnumeratedCharac
 MatchBenchmark
 MatchBenchmark
 MatchBenchmark
 MatchBenchmark
 StringMatchBenchma
MatchBenchmarkTe
 MatchBenchmarkTe

Match benchmarks
 measuring
 measureVW

Source Rewrite Code Critic Bytecode

measureVW

```

| localPattern localString |
self prettyPrintForMeasurement: 'VW'.
localPattern := self pattern.
localString := self string.
self stopWatch: [localPattern match: localString]

```

Method: #measureVW (measuring) **Parcel:** none **Package:** Match benchmarks

MatchBenchmarkTestCaseVWPlus >> measureVWPlus

Browser Edit Find View Package Class Protocol Method Tools Help

Find:

Package Hierarchy PartialHierarchy Instance Class Shared Variable Instance Variable

Object
 AbstractTestCase
 AbstractBenchmarkTest
AbstractMatchBench
 ClassBasedCharacter
 CondensedCharacter
 MatchBenchmark
 MatchBenchmark
 MatchBenchmark
 EnumeratedCharac
 MatchBenchmark
 MatchBenchmark
 MatchBenchmark
 StringMatchBenchma
 MatchBenchmarkTe
MatchBenchmarkTe

Match benchmarks
 measuring
 measureVWPlus

Source Rewrite Code Critic Bytecode

measureVWPlus

```

| localPattern localString |
self prettyPrintForMeasurement: 'VW+'.
localPattern := self pattern.
localString := self string.
self stopWatch: [localPattern inlinedCaseInsensitiveMatches: localString]

```

Method: #measureVWPlus (measuring) **Parcel:** none **Package:** Match benchmarks

Match in SUnit Benchmarks

$8 \times 56 = 448$ do its

vs

$8 + 56 = 64$ methods

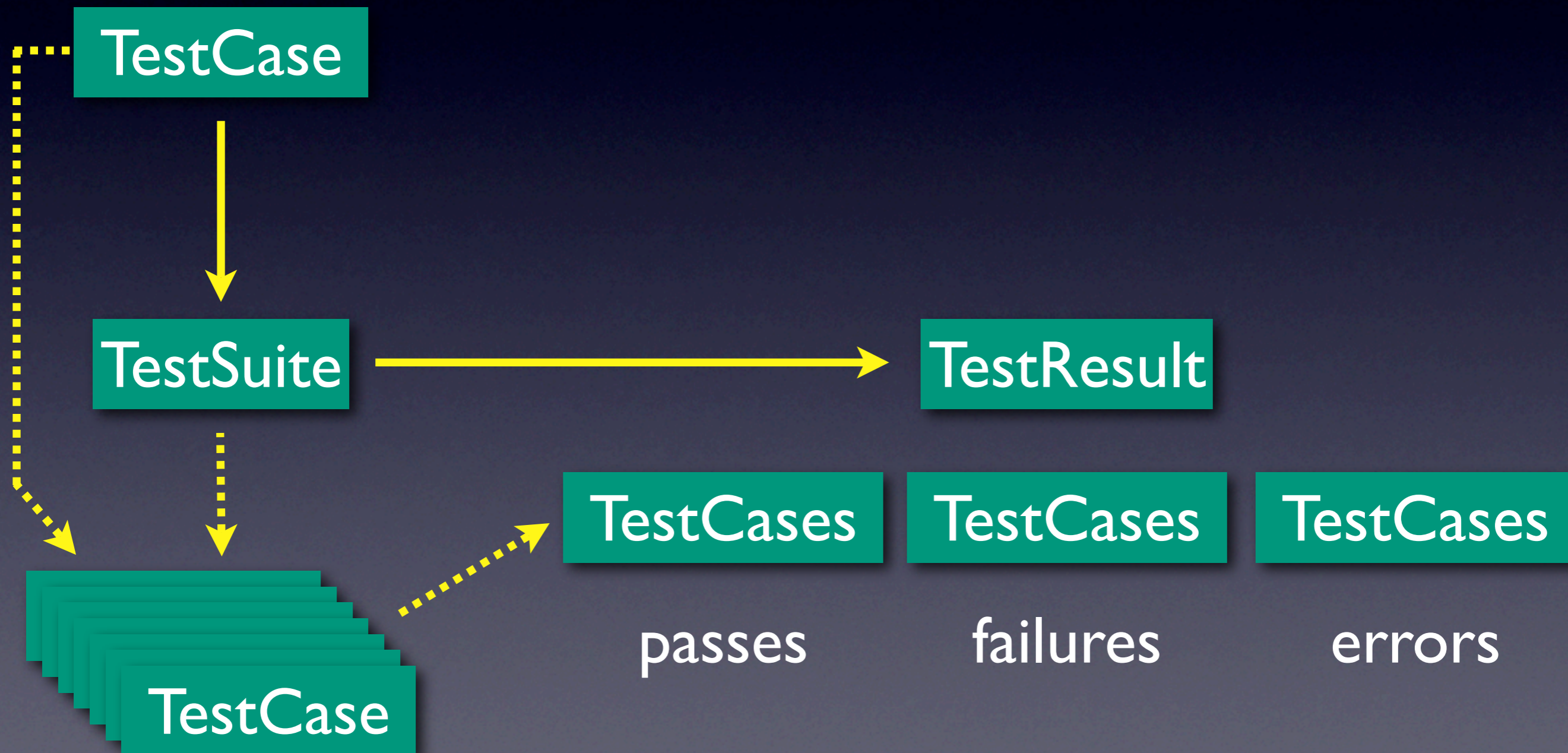
However... SUnit is deficient

- Extending it can be difficult.
- How many implementations of logging exist?
- Are the different variations cross compatible?
- SUnit VM and HAT point to serious limitations.

... therefore...

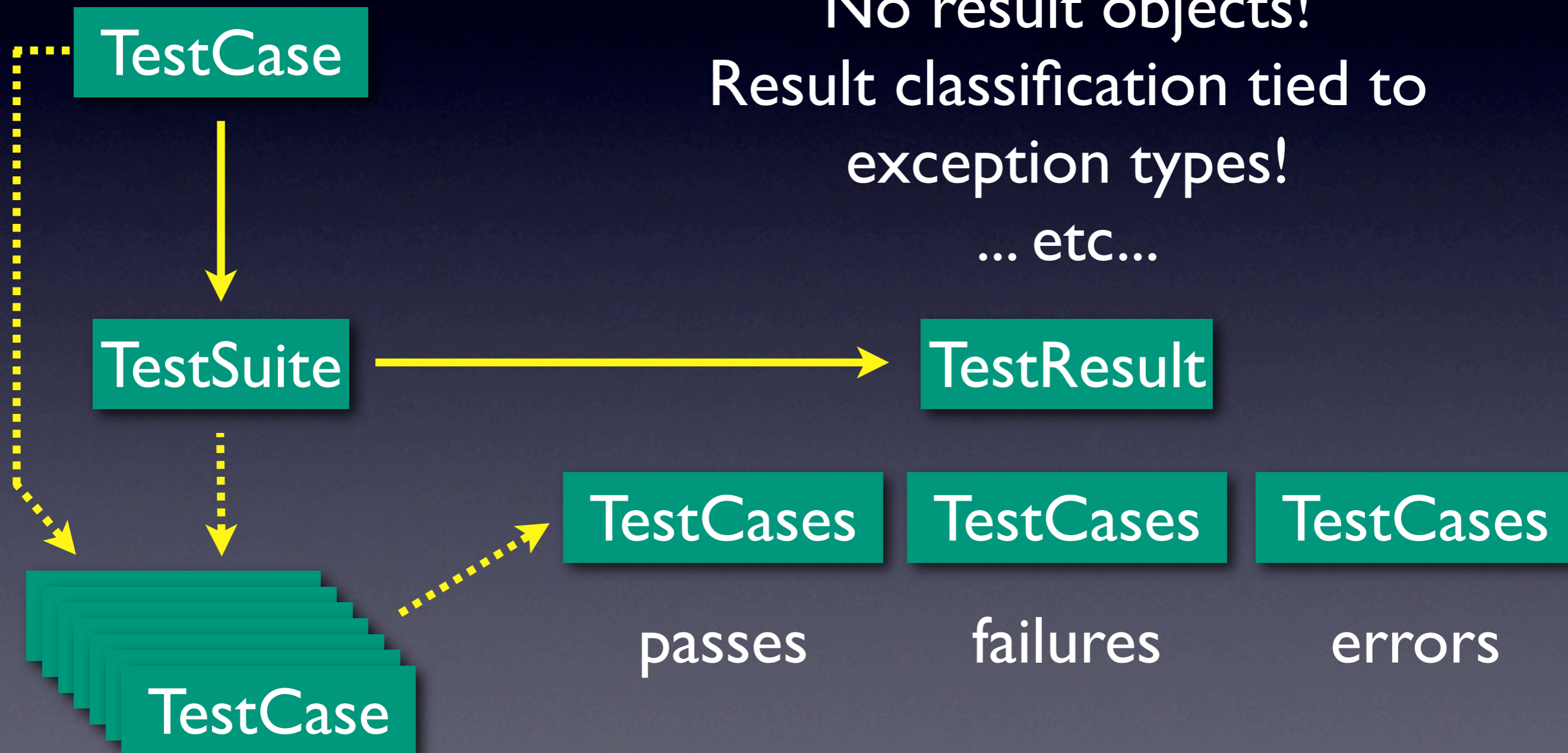
Assessments

Remember SUnit?

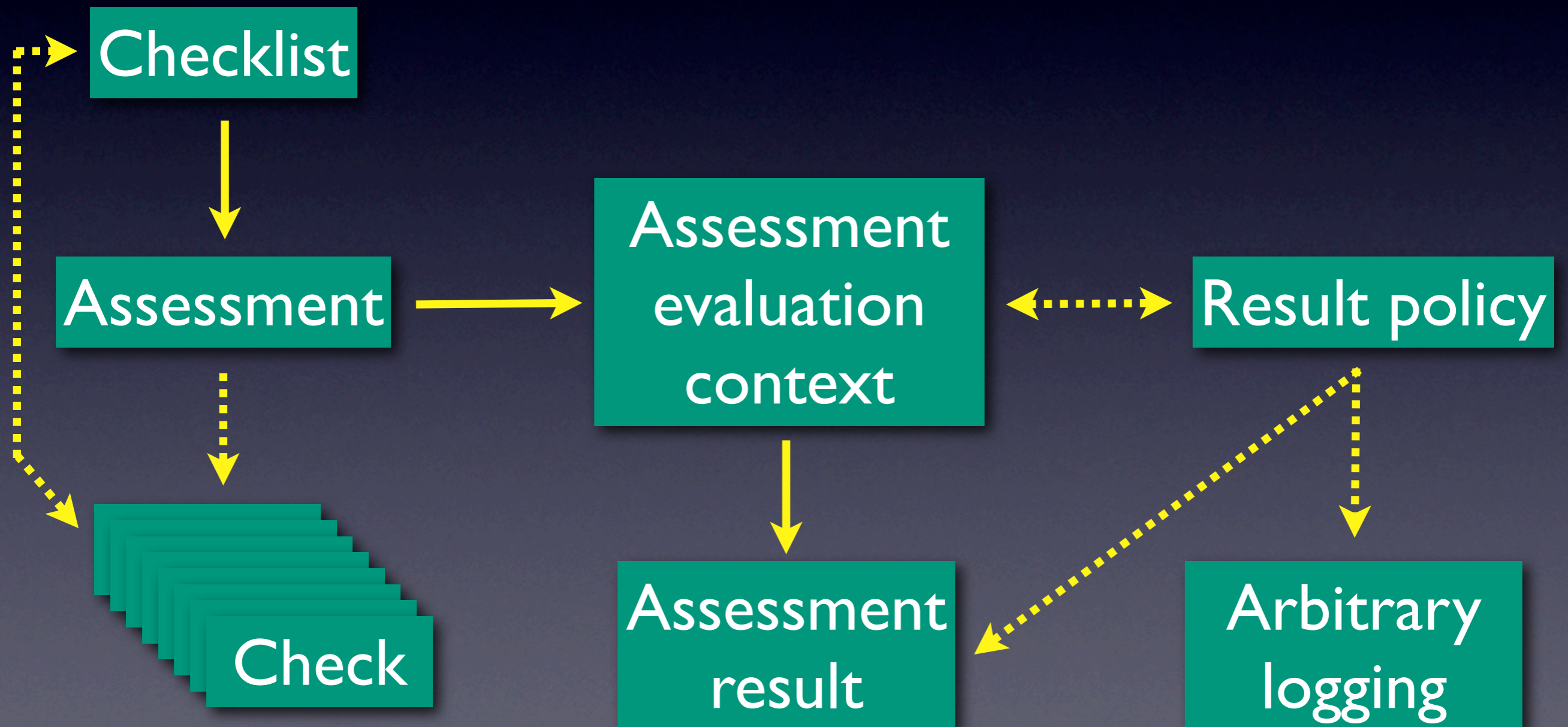


Remember SUnit?

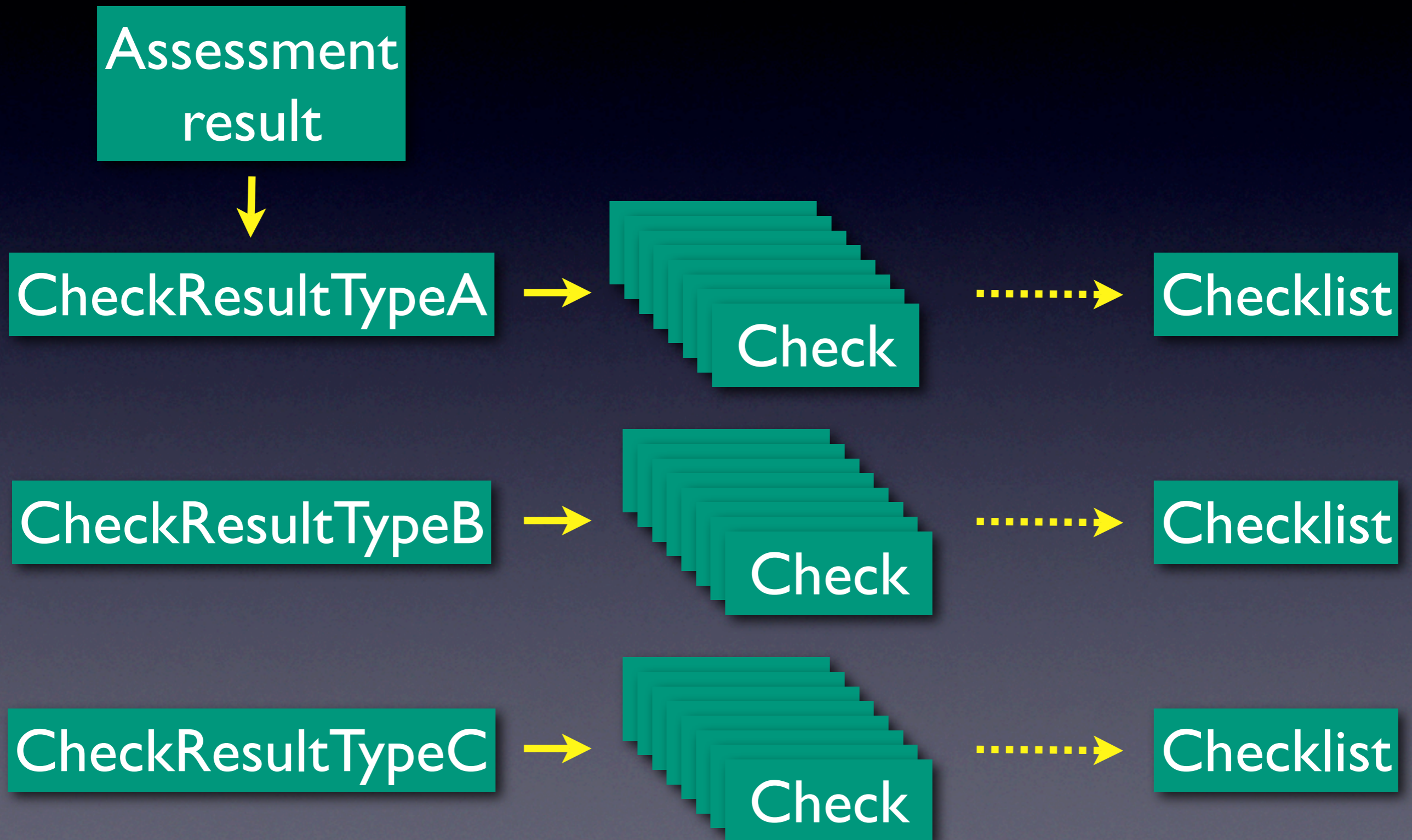
No evaluation context!
No result objects!
Result classification tied to
exception types!
... etc...



Assessments



Assessments



- Object
- AbstractResult**
- CheckResult
- CheckBenchmarkResult
- CheckBenchmarkAbstractExpectationNotice
- CheckBenchmarkExpectationFailed
- CheckBenchmarkExpectationMet
- CheckError
- CheckFailure
- CheckValidationFailure
- CheckNotice
- CheckPass
- CheckBenchmarkPass
- PrerequisiteResult
- PrerequisiteStartFailure
- PrerequisiteStopFailure

Assessments - Atomic results *

- accessing
- classification
- printing
- private
- private - accessing

- checkExecutionCount
- classificationTag
- initialize
- prettyPrint
- prettyPrint:
- printOn:
- printPrettyPrintDetailsO
- printResultDetailsOn:
- summaryPrintOn:

```
Smalltalk.Assessments defineClass: #AbstractResult
  superclass: #{Core.Object}
  indexedType: #none
  private: false
  instanceVariableNames: 'prettyPrint '
  classInstanceVariableNames: "
  imports: "
  category: 'Assessments'
```

- Object
- AbstractResult**
- CheckResult
- CheckBenchmarkResult
- CheckBenchmarkAbstractExpectationNotice
- CheckBenchmarkExpectationFailed
- CheckBenchmarkExpectationMet
- CheckError
- CheckFailure
- CheckValidationFailure
- CheckNotice
- CheckPass
- CheckBenchmarkPass
- PrerequisiteResult
- PrerequisiteStartFailure
- PrerequisiteStopFailure

Assessments - Atomic results *

accessing
 classification
 printing
 private
 private - accessing

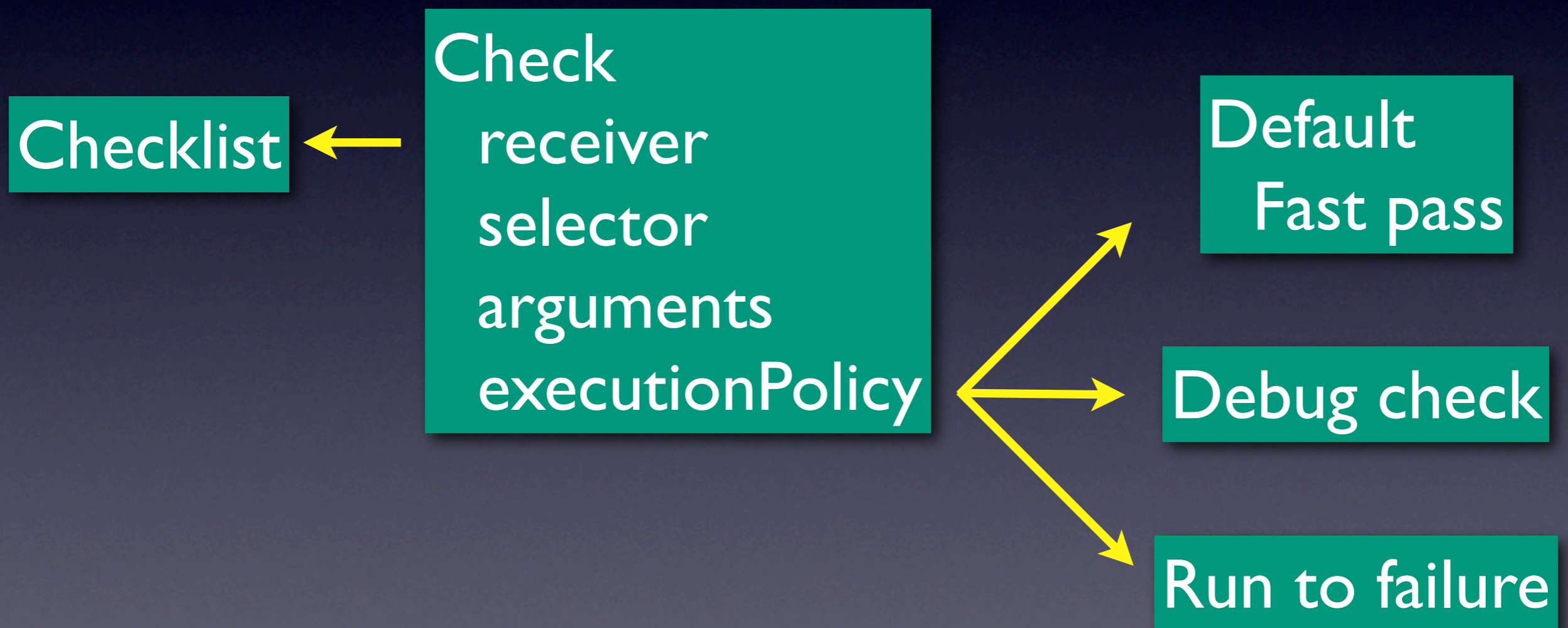
checkExecutionCount
classificationTag
 initialize
 prettyPrint
 prettyPrint:
 printOn:
 printPrettyPrintDetailsO
 printResultDetailsOn:
 summaryPrintOn:

```

classificationTag

^self class
  
```

Checks

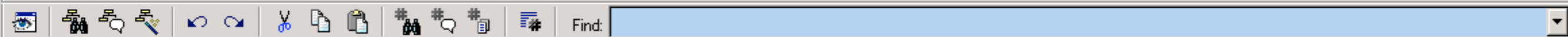


Package	Hierarchy	PartialHierarchy	Instance	Class	Shared Variable	Instance Variable
Local Image	GenericException		(compatibilit		handlerActionForEvaluation	
Andres VW * +	AssessmentNotification		(initialize-re		handlerActionForPrerequisiteEvaluation	
Andres VW.pvt * +	AssessmentMalfunctionNotification		(accessing)		handlerActionForRunningToFailure	
Andres Math * +	CheckAbstractFailureNotification		(actions)		messageTextForHandling	
Andres Math Tests *	CheckFailureNotification		assessment		occurredWhileEvaluating:inTheContextOf:	
Andres Tools * +	CheckValidationFailureNotification		(backward c		occurredWhileEvaluatingPrerequisiteAction:inTheContextOf:	
AGC *	CheckResumableFailureNotification		(backward c		occurredWhileRunning:toFailureInTheContextOf:	
AmigaRecompression *	CheckBenchmarkMeasurementNotification		(compatibilit		preventsAssessmentSuccess	
AllSubCollections * +	CheckBenchmarkAbstractExpectationNotifi		(constants)		resignalAsAssessmentMalfunctionNotification	
AndresCollections *	CheckBenchmarkExpectationFailedNotifi		(printing)			
Assessments * +	CheckBenchmarkExpectationMetNotificat		(signalling)			
Assessments Framework * +	CheckNoticeNotification		(SUnit)			
Assessments *	CheckPassNotification		(testing)			
Assessments - Atomic results *	CheckBenchmarkPassNotification		(tracing)			
Assessments - Custom checklists *	PrerequisiteFailureNotification		(private)			
Assessments - Evaluation contexts *	ResultAdoptionNotification		(private - ac			
Assessments - Exceptions * +	Error		(private - de			

Polymorphic exception handlers

```
Smalltalk.Core defineClass: #GenericException
  superclass: #{Core.Object}
  indexedType: #none
  private: false
  instanceVariableNames: 'messageText originator initialContext firstUnwindContext handlerContext parameter proceedBlock isResumable signal searchContext '
  classInstanceVariableNames: 'notifierString '
  imports: ''
  category: 'Kernel-Exception Handling'
```


... but if checks can have execution policies...



Package Hierarchy PartialHierarchy

- Andres VW * +
- Andres VW.pvt * +
- Andres Math * +
- Andres Math Tests *
- Andres Tools * +
- AGC *
 - AmigaRecompression *
 - AllSubCollections * +
 - AndresCollections *
 - Assessments +
 - Assessments Framework +
 - Assessments
 - Assessments - Atomic results
 - Assessments - Custom checklists
 - Assessments - Evaluation contexts
 - Assessments - Exceptions +
 - Assessments - Protected message sends

CheckExecutionPolicy

- AbstractSUnitCheckExecutionPolicy
- SUnitCheckExecutionPolicy
- SUnitVMCheckExecutionPolicy
- SUnitVMBenchmarkExecutionPolicy
- SUnitVMValidationExecutionPolicy
- SUnitTooCheckExecutionPolicy
- AssessmentsCheckExecutionPolicy
- AssessmentsBenchmarkCheckExecutionPolicy
- AssessmentsValidationCheckExecutionPolicy
- ProtectedMessageSend
- AttemptToStartPrerequisite
- AttemptToStopPrerequisite
- Check
 - AbstractDebugCheck
 - DebugCheck
 - RunToFailureCheck

Instance	Class	Shared Variable	Instance Variable
accessing			aboutToEvaluateCheck:inTheContextOf:
execution - checks			aboutToExecuteOn:
perform - checks			basicDebug:
perform - prerequisite			basicPerform:
private			basicRunToFailure:
private - accessing			benchmarkExpectationFailedNotification
private - errors			benchmarkExpectationMetNotification
private - exceptions			benchmarkMeasurementNotification
private - exceptions to			checkExecutionCountFor:
private - execution - c			compiledMethodForEvaluationOf:
private - perform			debugPerformCheck:inTheContextOf:
private - perform debu			doneEvaluatingCheck:inTheContextOf:
private - perform run t			doneExecutingOn:
			exceptionsToTrap
			exceptionsToTrap:
			fastPerformCheck:inTheContextOf:
			fastPerformCheckPassResultFor:

```
Smalltalk.Assessments defineClass: #CheckExecutionPolicy
  superclass: #(Core.Object)
  indexedType: #none
  private: false
  instanceVariableNames: 'exceptionsToTrap'
  classInstanceVariableNames: 'current'
  imports: ''
  category: 'Assessments'
```

Class: Assessments.CheckExecutionPolicy

Parcel: none

Package: Assessments - Protected message sends

Package	Hierarchy	PartialHierarchy	Instance	Class	Shared Variable	Instance Variable
<ul style="list-style-type: none"> Andres VW * + Andres VW.pvt * + Andres Math * + Andres Math Tests * Andres Tools * + <ul style="list-style-type: none"> AGC * <ul style="list-style-type: none"> AmigaRecompression * AllSubCollections * + AndresCollections * Assessments + <ul style="list-style-type: none"> Assessments Framework + <ul style="list-style-type: none"> Assessments Assessments - Atomic results Assessments - Custom checklists Assessments - Evaluation contexts Assessments - Exceptions + Assessments - Protected message sends 	<ul style="list-style-type: none"> CheckExecutionPolicy AbstractSUnitCheckExecutionPolicy SUnitCheckExecutionPolicy SUnitVMCheckExecutionPolicy SUnitVMBenchmarkExecutionPolicy SUnitVMValidationExecutionPolicy SUnitTooCheckExecutionPolicy AssessmentsCheckExecutionPolicy AssessmentsBenchmarkCheckExecutionPolicy AssessmentsValidationCheckExecutionPolicy ProtectedMessageSend AttemptToStartPrerequisite AttemptToStopPrerequisite Check AbstractDebugCheck DebugCheck RunToFailureCheck 	<ul style="list-style-type: none"> accessing execution - checks perform - checks perform - prerequisite private private - accessing private - errors private - exceptions private - exceptions to private - execution - c private - perform private - perform debu private - perform run t 	<ul style="list-style-type: none"> doneExecutingOn: exceptionsToTrap exceptionsToTrap: fastPerformCheck:inTheContextOf: fastPerformCheckPassResultFor: firstExceptionsToTrap initialize malfunctionNotification maximumStepsWhenDebugging newActualReceiverForCheck:inTheContext newExceptionsToTrap newProcessToDebugEvaluationOf:signallir newSuspendedProcessToDebugEvaluation notifyMalfunction: performAttemptToStartPrerequisite: performAttemptToStopPrerequisite: performCheck:inTheContextOf: 			

```
performCheck: aCheck inTheContextOf: anEvaluationContext

| checkPassed |
checkPassed := self basicPerform: aCheck.
checkPassed ifFalse: [^self].
self raisePassNotificationFor: aCheck
```

Package	Hierarchy	PartialHierarchy	Instance	Class	Shared Variable	Instance Variable
Andres VW * +			accessing			aboutToEvaluateCheck:inTheContextOf:
Andres VW.pvt * +			execution			allScenarioSelectorsFor:
Andres Math * +			perform - checks			allScenariosPerform:withCheck:inTheCont
Andres Math Tests *			private - exceptions			checkExecutionCountFor:
Andres Tools * +			private - perform			debugPerformCheck:inTheContextOf:
AGC *			private - SUnit except			fastPerformCheck:inTheContextOf:
AmigaRecompression *			private - SUnit except			handleKnownSUnitException:
AllSubCollections * +						handleSUnitVMBenchmarkExpectationNo
AndresCollections *						handleSUnitVMBenchmarkMeasurementN
Assessments +						performCheck:inTheContextOf:
Assessments Framework +						resumableFailureNotifications
Assessments						runToFailurePerformCheck:inTheContextO
Assessments - Atomic results						sunitVMBenchmarkExpectationNotificati
Assessments - Custom checklists						sunitVMBenchmarkMeasurementNotificati
Assessments - Evaluation contexts						
Assessments - Exceptions +						
Assessments - Protected message sends						

```
performCheck: aCheck inTheContextOf: anEvaluationContext
self
  allScenariosPerform:
    [:check |
      super
        performCheck: check
        inTheContextOf: anEvaluationContext
    ]
  withCheck: aCheck
  inTheContextOf: anEvaluationContext
```

SUnitTooCheckExecutionPolicy >> handleKnownSUnitException:

Browser Edit Find View Package Class Protocol Method Tools Help

Find: []

Package Hierarchy PartialHierarchy

- Andres VW * +
 - Andres VW.pvt * +
 - Andres Math * +
 - Andres Math Tests * +
 - Andres Tools * +
 - AGC *
 - AmigaRecompression *
 - AllSubCollections * +
 - AndresCollections *
 - Assessments +
 - Assessments Framework +
 - Assessments
 - Assessments - Atomic results
 - Assessments - Custom checklists
 - Assessments - Evaluation contexts
 - Assessments - Exceptions +
 - Assessments - Protected message sends

Instance Class Shared Variable Instance Variable

- private - SUnit except
- private - SUnit except
- handleKnownSUnitException:
 - sunitTooFailureNotification

CheckExecutionPolicy
 AbstractSUnitCheckExecutionPolicy
 SUnitCheckExecutionPolicy
 SUnitVMCheckExecutionPolicy
 SUnitVMBenchmarkExecutionPolicy
 SUnitVMValidationExecutionPolicy
SUnitTooCheckExecutionPolicy
 AssessmentsCheckExecutionPolicy
 AssessmentsBenchmarkCheckExecutionPolicy
 AssessmentsValidationCheckExecutionPolicy
 ProtectedMessageSend
 AttemptToStartPrerequisite
 AttemptToStopPrerequisite
 Check
 AbstractDebugCheck
 DebugCheck
 RunToFailureCheck

Source Rewrite Code Critic Bytecode

```

handleKnownSUnitException: anException
  "No polymorphism because direct reference to exception classes is not possible"

  super handleKnownSUnitException: anException.
  (anException isKindOf: self sunitTooFailureNotification)
    ifTrue: [self handleSUnitFailureNotification: anException]
  
```

Remapping SUnitToo's exceptions into Assessments exceptions

Method: #handleKnownSUnitException: (private - SUnit exception handling) Parcel: none Package: Assessments - Protected message sends

... but how are the disparate test cases bridged?...

Assessments - Test bridge

Browser Edit Find View Package Class Protocol Method Tools Help

Package Hierarchy PartialHierarchy

Instance Class Shared Variable Instance Variable

Andres VW.pvt * +
 Andres Math * +
 Andres Math Tests * +
 Andres Tools * +
 AGC *
 AmigaRecompression *
 AllSubCollections * +
 AndresCollections *
 Assessments +
 Assessments Framework +
 Assessments
 Assessments - Atomic results
 Assessments - Custom checklists
 Assessments - Evaluation contexts
 Assessments - Exceptions +
 Assessments - Protected message sends
Assessments - Test bridge
 Assessments - VW extensions +

AbstractSUnitChecklistBridge
 SUnitTestCase
 SUnitTooTestCase
 SUnitVMAbstractTestCase
 SUnitVMBenchmarkTestCase
 SUnitVMTestCase
 SUnitVMValidatorTestCase
 CheckRepositoryMetaclass
 AbstractSUnitCheckRepositoryMetaclass
 SUnitCheckRepositoryMetaclass
 SUnitCheckRepositoryRootMetaclass
 SUnitVMAbstractCheckRepositoryMetaclass
 SUnitVMBenchmarkCheckRepositoryMetaclas
 SUnitVMCheckRepositoryMetaclass
 SUnitVMValidatorCheckRepositoryMetaclass
 SUnitTooCheckRepositoryMetaclass
 SUnitTooCheckRepositoryRootMetaclass
 SUnitPrerequisite

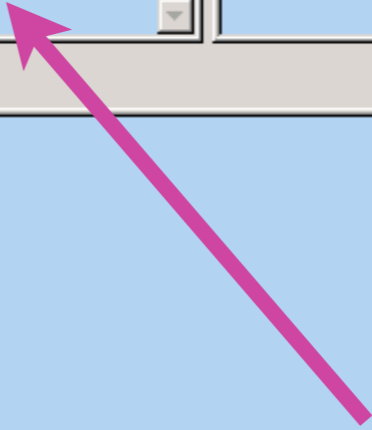
Source Properties Comment Rewrite Code Critic

```

Smalltalk defineClass: #NameOfClass
  superclass: #{NameOfSuperclass}
  indexedType: #none
  private: false
  instanceVariableNames: 'instVarName1 instVarName2'
  classInstanceVariableNames: ''
  imports: ''
  category: 'Assessments - Test bridge'
  
```

Package: Assessments - Test bridge

Version: (1.0 beta 13,valloud)=



Metaclasses!

CheckRepositoryMetaclass

Browser Edit Find View Package Class Protocol Method Tools Help

Package Hierarchy PartialHierarchy

Instance Class Shared Variable Instance Variable

Andres VW.pvt * +
 Andres Math * +
 Andres Math Tests * +
 Andres Tools * +
 AGC *
 AmigaRecompression *
 AllSubCollections * +
 AndresCollections * +
 Assessments +
 Assessments Framework +
 Assessments
 Assessments - Atomic results
 Assessments - Custom checklists
 Assessments - Evaluation contexts
 Assessments - Exceptions +
 Assessments - Protected message sends
 Assessments - Test bridge
 Assessments - VW extensions +

AbstractSUnitChecklistBridge
 SUnitTestCase
 SUnitTooTestCase
 SUnitVMAbstractTestCase
 SUnitVMBenchmarkTestCase
 SUnitVMTestCase
 SUnitVMValidatorTestCase
CheckRepositoryMetaclass
 AbstractSUnitCheckRepositoryMetaclass
 SUnitCheckRepositoryMetaclass
 SUnitCheckRepositoryRootMetaclass
 SUnitVMAbstractCheckRepositoryMetaclas
 SUnitVMBenchmarkCheckRepositoryMetaclas
 SUnitVMCheckRepositoryMetaclass
 SUnitVMValidatorCheckRepositoryMetaclass
 SUnitTooCheckRepositoryMetaclass
 SUnitTooCheckRepositoryRootMetaclass
 SUnitPrerequisite

instance creation
 accessing
 assessments
 check lookup
 check services
 checks
 classification
 comparing
 constants
 custom checklists
 lookup
 prerequisites
 printing
 selector lookup
 testing
 private
 private - accessing

=
 assessment
 assessmentClass
 atSelector:
 browse
 checkClass
 checklistWithCheckSelectorsNamed:
 checklistWithCheckSelectorsNamed:toBe
 checks
 checksDo:
 checkSelectorLookupRoot
 checkSelectorLookupSubclasses
 checkSelectorLookupSuperclass
 checkSelectorPrefix
 checkSelectorsDo:
 classToImpersonate
 classToImpersonate:
 evaluationResult

Source Comment Hierarchy Diagram Rewrite Code Critic

```
Smalltalk.Assessments defineClass: #CheckRepositoryMetaclass
  superclass: #{Core.Object}
  indexedType: #none
  private: false
  instanceVariableNames: 'classToImpersonate superclass subclasses '
  classInstanceVariableNames: ''
  imports: ''
  category: 'Assessments - SUnit bridge'
```

Class: Assessments.CheckRepositoryMetaclass Parcel: none Package: Assessments - Test bridge



Instance side polymorphic
with class side of Checklist

- Andres VW.pvt * +
 - Andres Math * +
 - Andres Math Tests * +
 - Andres Tools * +
 - AGC *
 - AmigaRecompression *
 - AllSubCollections * +
 - AndresCollections * +
 - Assessments +
 - Assessments Framework +
 - Assessments
 - Assessments - Atomic results
 - Assessments - Custom checklists
 - Assessments - Evaluation contexts
 - Assessments - Exceptions +
 - Assessments - Protected message sends
 - Assessments - Test bridge
 - Assessments - VW extensions +

- AbstractSUnitChecklistBridge
- SUnitTestCase
- SUnitTooTestCase
- SUnitVMAbstractTestCase
- SUnitVMBenchmarkTestCase
- SUnitVMTestCase
- SUnitVMValidatorTestCase
- CheckRepositoryMetaclass**
- AbstractSUnitCheckRepositoryMetaclass
- SUnitCheckRepositoryMetaclass
- SUnitCheckRepositoryRootMetaclass
- SUnitVMAbstractCheckRepositoryMetaclass
- SUnitVMBenchmarkCheckRepositoryMetaclass
- SUnitVMCheckRepositoryMetaclass
- SUnitVMValidatorCheckRepositoryMetaclass
- SUnitTooCheckRepositoryMetaclass
- SUnitTooCheckRepositoryRootMetaclass
- SUnitPrerequisite

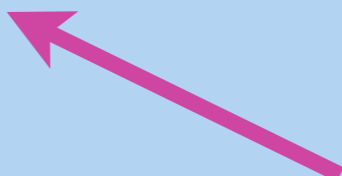
instance creation

- accessing
- assessments
- check lookup
- check services
- checks
- classification
- comparing
- constants
- custom checklists
- lookup
- prerequisites
- printing
- selector lookup
- testing
- private
- private - accessing

=

- assessment
- assessmentClass
- atSelector:
- browse
- checkClass
- checklistWithCheckSelectorsNamed:
- checklistWithCheckSelectorsNamed:toBe
- checks
- checksDo:
- checkSelectorLookupRoot
- checkSelectorLookupSubclasses
- checkSelectorLookupSuperclass
- checkSelectorPrefix
- checkSelectorsDo:
- classToImpersonate
- classToImpersonate:
- evaluationResult

```
Smalltalk.Assessments defineClass: #CheckRepositoryMetaclass
  superclass: #(Core.Object)
  indexedType: #none
  private: false
  instanceVariableNames: 'classToImpersonate superclass subclasses '
  classInstanceVariableNames: "
  imports: "
  category: 'Assessments - SUnit bridge'
```



Instance side polymorphic with classes too

- Andres VW.pvt * +
 - Andres Math * +
 - Andres Math Tests * +
 - Andres Tools * +
 - AGC *
 - AmigaRecompression *
 - AllSubCollections * +
 - AndresCollections * +
 - Assessments +
 - Assessments Framework +
 - Assessments
 - Assessments - Atomic results
 - Assessments - Custom checklists
 - Assessments - Evaluation contexts
 - Assessments - Exceptions +
 - Assessments - Protected message sends
 - Assessments - Test bridge
 - Assessments - VW extensions +

- AbstractSUnitChecklistBridge
- SUnitTestCase
- SUnitTooTestCase
- SUnitVMAbstractTestCase
- SUnitVMBenchmarkTestCase
- SUnitVMTestCase
- SUnitVMValidatorTestCase
- CheckRepositoryMetaclass**
- AbstractSUnitCheckRepositoryMetaclass
- SUnitCheckRepositoryMetaclass
- SUnitCheckRepositoryRootMetaclass
- SUnitVMAbstractCheckRepositoryMetaclass
- SUnitVMBenchmarkCheckRepositoryMetaclass
- SUnitVMCheckRepositoryMetaclass
- SUnitVMValidatorCheckRepositoryMetaclass
- SUnitTooCheckRepositoryMetaclass
- SUnitTooCheckRepositoryRootMetaclass
- SUnitPrerequisite

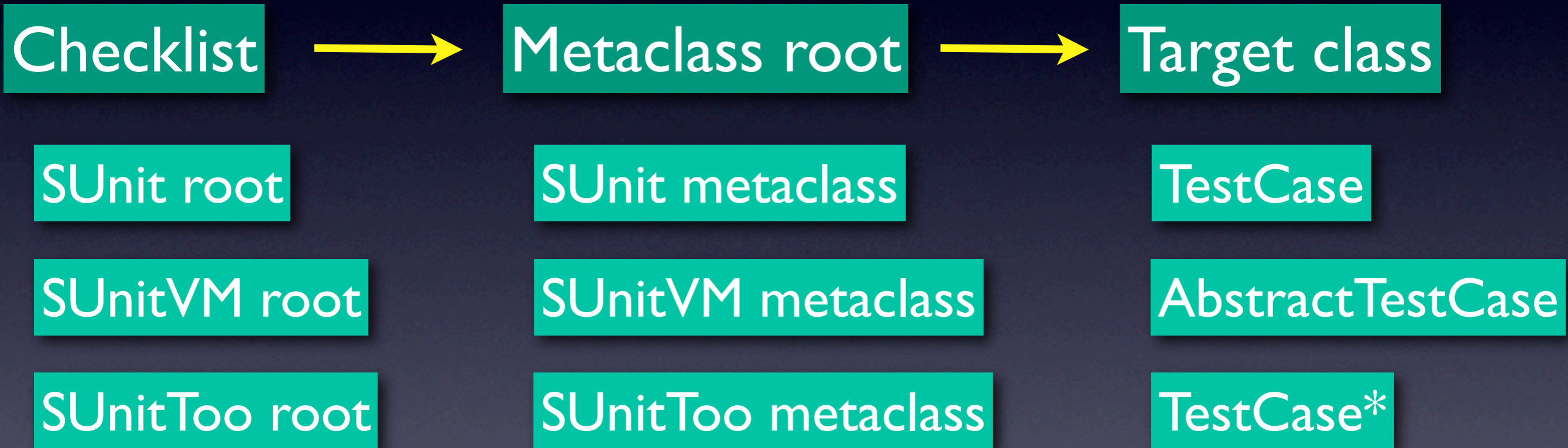
- instance creation
- accessing
- assessments
- check lookup
- check services
- checks
- classification
- comparing
- constants
- custom checklists
- lookup
- prerequisites
- printing
- selector lookup
- testing
- private
- private - accessing

- new

```
new

^self classToImpersonate new
```

Test bridge metaclasses



Package Hierarchy PartialHierarchy Instance Class Shared Variable Instance Variable

- Andres VW.pvt * +
 - Andres Math * +
 - Andres Math Tests * +
 - Andres Tools * +
 - AGC *
 - AmigaRecompression *
 - AllSubCollections * +
 - AndresCollections *
 - Assessments +
 - Assessments Framework +
 - Assessments - Test bridge
 - Assessments - VW extensions +

AbstractSUnitChecklistBridge
 SUnitTestCase
 SUnitTooTestCase
 SUnitVMAbstractTestCase
 SUnitVMBenchmarkTestCase
 SUnitVMTestCase
 SUnitVMValidatorTestCase
CheckRepositoryMetaclass
 AbstractSUnitCheckRepositoryMetaclass
 SUnitCheckRepositoryMetaclass
 SUnitCheckRepositoryRootMetaclass
 SUnitVMAbstractCheckRepositoryMetaclass
 SUnitVMBenchmarkCheckRepositoryMetaclas
 SUnitVMCheckRepositoryMetaclass
 SUnitVMValidatorCheckRepositoryMetaclass
 SUnitTooCheckRepositoryMetaclass
 SUnitTooCheckRepositoryRootMetaclass
 SUnitPrerequisite

instance creation
 pretendToBe:subclassedFrom:

```
pretendToBe: aClass
subclassedFrom: aSuperclass

^self new
  pretendToBe: aClass subclassedFrom: aSuperclass;
  yourself
```

References to: pretendToBe:subclassedFrom: - AbstractSUnitChecklistBridge class>>newMetacla...

Browser Edit Find View Class Method Tools Help

Method Class

- Assessments.AbstractSUnitChecklistBridge class newMetaclassForTestCaseRoot {SUnit}
- Assessments.CheckRepositoryMetaclass newSubclasses {private}
- Assessments.CheckRepositoryMetaclass class pretendToBe:subclassedFrom: {instance creation}

Source Rewrite Code Critic Bytecode

newMetaclassForTestCaseRoot

```
^self metaclassClass
  pretendToBe: self testCaseRoot
  subclassedFrom: self
```

Method: #newMetaclassForTestCaseRo **Parcel:** none **Package:** Assessments - Test bridge

References to: pretendToBe:subclassedFrom: - CheckRepositoryMetaclass>>newSubclasses

Browser Edit Find View Class Method Tools Help

Method Class

- Assessments.AbstractUnitChecklistBridge class newMetaclassForTestCaseRoot {SUnit}
- Assessments.CheckRepositoryMetaclass newSubclasses {private}
- Assessments.CheckRepositoryMetaclass class pretendToBe:subclassedFrom: {instance creation}

Source Rewrite Code Critic Bytecode

newSubclasses

```
^self classToImpersonate subclasses collect:  
  [:each |  
    self subclassClass  
    pretendToBe: each  
    subclassedFrom: self  
  ]
```

Method: #newSubclasses (private) **Parcel:** none **Package:** Assessments - Test bridge

Assessments v1.x demo

Questions?



Smalltalks 2008

The Conference

You are hereby invited to participate in the second edition of one of the most important Smalltalk conferences in recent years.

Smalltalkers from all around the globe will be attending this international event. Outstanding guest community personalities will deliver the conference keynotes.

There will be both a Scientific Research and a Software Industry track for presentations, as well as tutorial sessions for hands-on experience.

A coding contest will be held live on stage.

Once again, entry, coffee and pastries are free.

See you there!

Scientific Research track submission deadline: August 31
Software Industry track submission deadline: September 30
Coding contest registration deadline: September 30
Conference registration deadline: November 7

November 13-14-15, 2008
Universidad Abierta Interamericana
Avenida Montes de Oca 745
Buenos Aires, Argentina

<http://neuquina.lifia.info.unlp.edu.ar:8001/Smalltalks-2008>
smalltalks2008@gmail.com

Thank you, and see
you in Buenos Aires!