Advanced Programming

Andrew P. Black & Tim Sheard

Where did all the time go?

First make it right, then make it fast.

How long does it take?

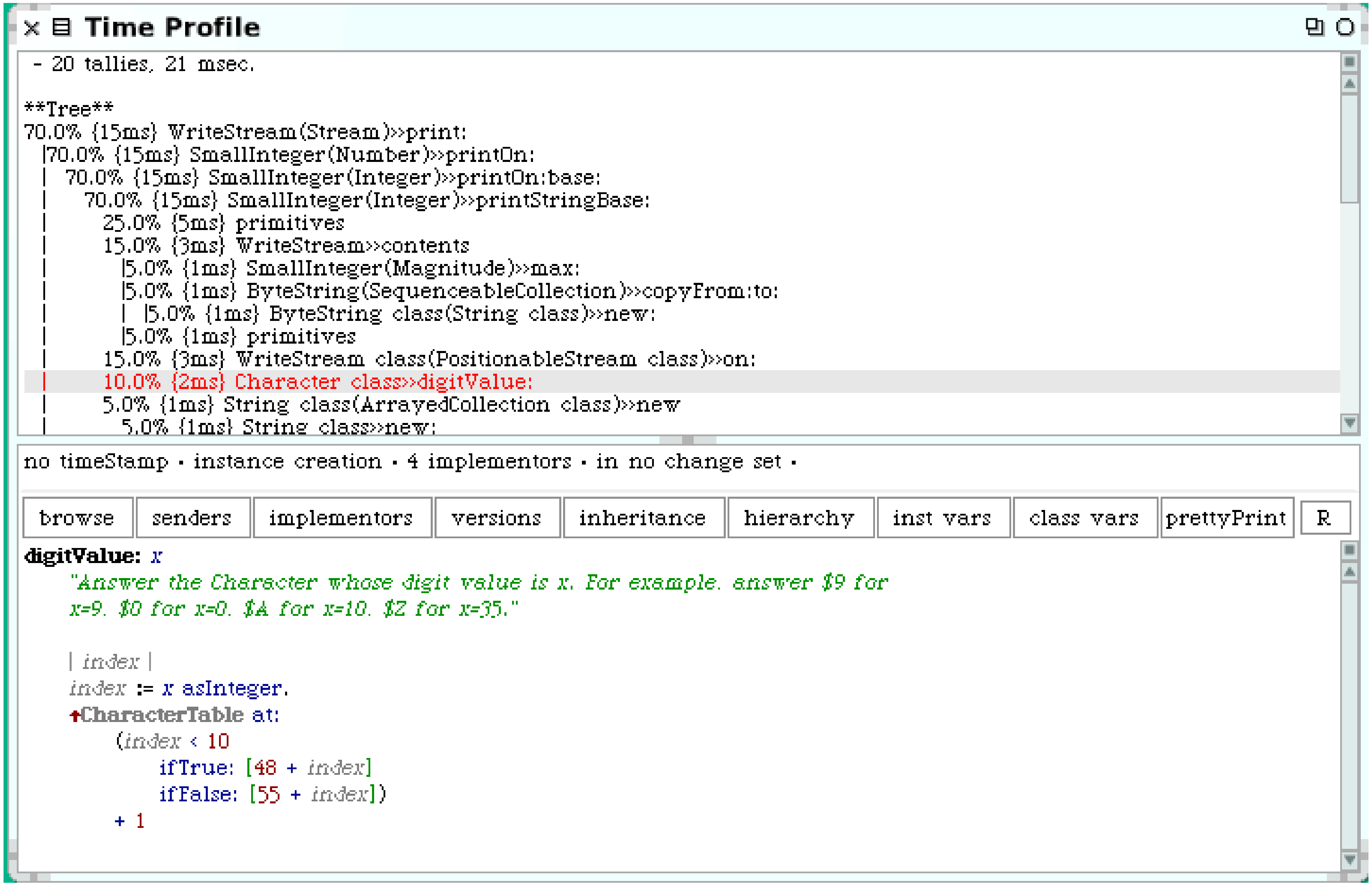
- several methods on blocks let you find out how long the block takes to execute
  - [data detectMax: [:each | each]] timeToRun
    ➠ 7
  - [data detectMax: [:each | each]] durationToRun
    ➠ 0:00:00:00.007
  - [data detectMax: [:each | each]] bench
    ➠ '153.78 per second.'

Why did it take so long?

- TimeProfileBrowser is your friend
    - ▸ probabalistic sampling of pc (roughly once per ms)
- Really easy to use:
    - TimeProfileBrowser onBlock: [<your code here>]
- Let's use it on some examples:
    - ▸ TimeProfileBrowser onBlock:

```
TimeProfileBrowser onBlock:
  [String streamContents:
    [ :str | data do: [ : each | str print: each; space ]]]
```

Portland State UNIVERSITY



```
 - 20 tallies, 21 msec.

**Tree**
70.0% {15ms} WriteStream(Stream)>>print:
 |70.0% {15ms} SmallInteger(Number)>>printOn:
 |  70.0% {15ms} SmallInteger(Integer)>>printOn:base:
 |    70.0% {15ms} SmallInteger(Integer)>>printStringBase:
 |      25.0% {5ms} primitives
 |      15.0% {3ms} WriteStream>>contents
 |        |5.0% {1ms} SmallInteger(Magnitude)>>max:
 |        |5.0% {1ms} ByteString(SequenceableCollection)>>copyFrom:to:
 |        |  |5.0% {1ms} ByteString class(String class)>>new:
 |        |5.0% {1ms} primitives
 |      15.0% {3ms} WriteStream class(PositionableStream class)>>on:
 |      10.0% {2ms} Character class>>digitValue:
 |      5.0% {1ms} String class(ArrayedCollection class)>>new
 |        5.0% {1ms} String class>>new:
```

```
no timeStamp · instance creation · 4 implementors · in no change set ·
```

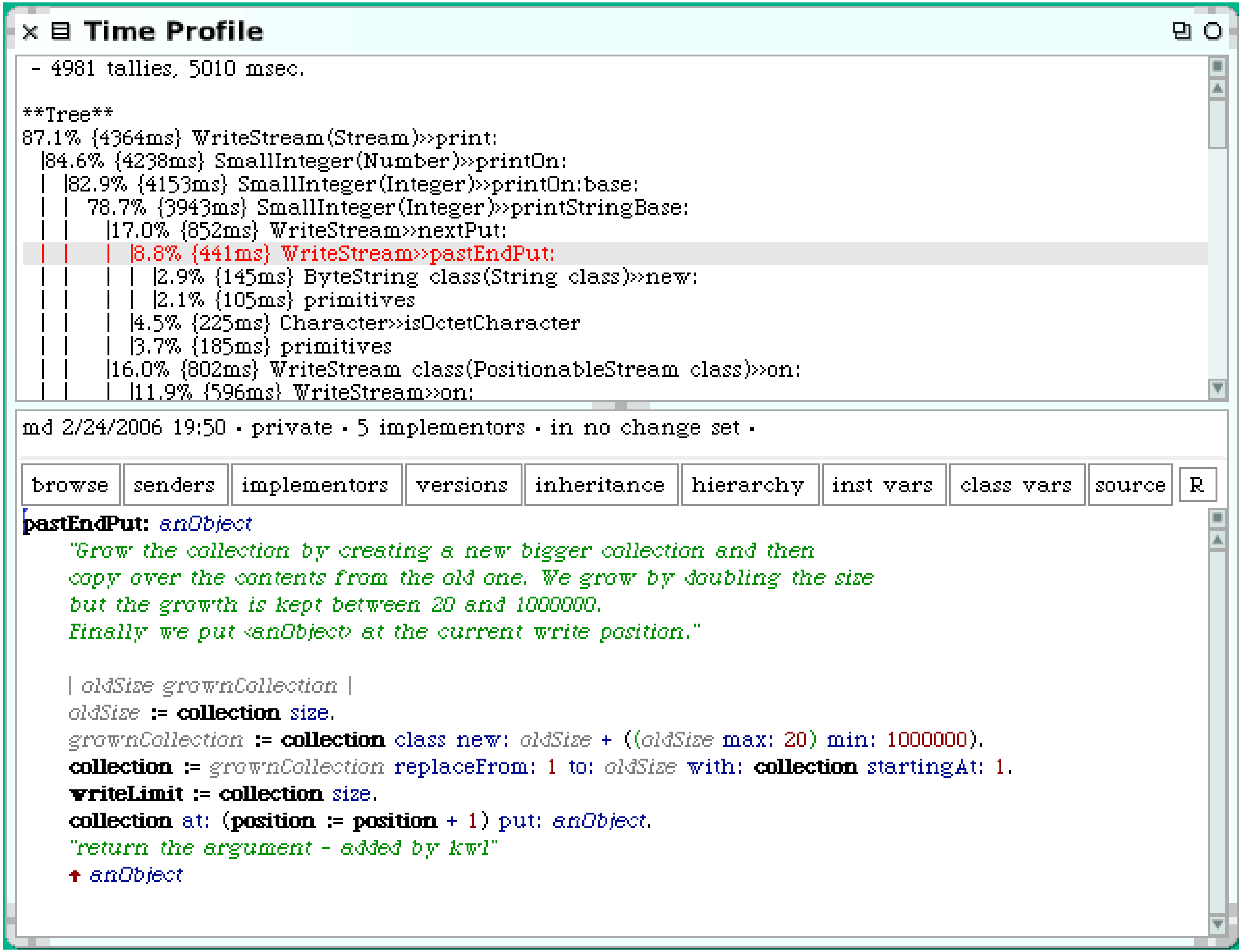
```
digitValue: x
    "Answer the Character whose digit value is x. For example, answer $9 for
    x=9, $0 for x=0, $A for x=10, $Z for x=35."

    | index |
    index := x asInteger.
    ↑CharacterTable at:
        (index < 10
            ifTrue: [48 + index]
            ifFalse: [55 + index])
        + 1
```

20 tallies: barely enough samples to be valid

Portland State UNIVERSITY

- Let’s try:
  - TimeProfileBrowser onBlock:
    [[String streamContents:
    [:str | data do: [: each | str print: each; space]]]
    bench]
  - [< code >] bench does 5 seconds’-worth of computation (or 1 iteration)



```
 - 4981 tallies, 5010 msec.

**Tree**
87.1% {4364ms} WriteStream(Stream)>>print:
  |84.6% {4238ms} SmallInteger(Number)>>printOn:
  |  |82.9% {4153ms} SmallInteger(Integer)>>printOn:base:
  |  |  78.7% {3943ms} SmallInteger(Integer)>>printStringBase:
  |  |    |17.0% {852ms} WriteStream>>nextPut:
  |  |    |  |8.8% {441ms} WriteStream>>pastEndPut:
  |  |    |  |  |2.9% {145ms} ByteString class(String class)>>new:
  |  |    |  |  |2.1% {105ms} primitives
  |  |    |  |4.5% {225ms} Character>>isOctetCharacter
  |  |    |  |3.7% {185ms} primitives
  |  |    |16.0% {802ms} WriteStream class(PositionableStream class)>>on:
  |  |    |  |11.9% {596ms} WriteStream>>on:
```

md 2/24/2006 19:50 • private • 5 implementors • in no change set •

```
pastEndPut: anObject
	"Grow the collection by creating a new bigger collection and then
	copy over the contents from the old one. We grow by doubling the size
	but the growth is kept between 20 and 1000000.
	Finally we put <anObject> at the current write position."

	| oldSize grownCollection |
	oldSize := collection size.
	grownCollection := collection class new: oldSize + ((oldSize max: 20) min: 1000000).
	collection := grownCollection replaceFrom: 1 to: oldSize with: collection startingAt: 1.
	writeLimit := collection size.
	collection at: (position := position + 1) put: anObject.
	"return the argument - added by kwl"
	↑ anObject
```