- [illegible]

[illegible]:

[illegible]:

- [illegible]:

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]:

- [illegible]

O[illegible] [illegible] [illegible] S[illegible] S[illegible]:

- A [illegible] [illegible] [illegible] [illegible]

S[illegible] F[illegible]:

- E[illegible] [illegible] [illegible] [illegible];

J[illegible] [illegible] [illegible] [illegible] [illegible]:

.class

B[illegible] → J[illegible] → [illegible]:

Snnnn nn nnnnnnnnn Jnnnnn:

Snnn 1: Pnnnnnn nnn≋nnn, nnnnnnn Jnnnnn.

Snnn 2: Cnnnnnnn nnnnn nnnnn. nnn nnn nnnn, nnn

nnnn nnnn nnnnnnnnnnn.

Bnnnnn Annnn

Snnnnn nnnnnnn nn nnnnn Snnn:

Annnnn Pnnnnnnn Cnnnnnnn:

```
StmtList stmtList = body.getStmtList();
ListIterator stmtIt = stmtList.listIterator();
```

[illegible] 4: [illegible]

- [illegible]

```
public int stepPoly(int x)
{ if (x < 0)
    { System.out.println("error"); return -1; }
  else if (x <= 5)
```

[illegible]:

[illegible] 0 [illegible] 2

[illegible] 3: [illegible] U/D [illegible]
D/U [illegible].

[illegible]:

- [illegible]

[illegible]:

- [illegible]

soft [illegible]

hard [illegible]

[illegible]

- [illegible]
- [illegible] *hard* [illegible]

[illegible]:

```
import java.awt.*;

class Test
```

[illegible]:

[illegible] 1: [illegible]

[illegible] 2: [illegible]

[illegible] 3: [illegible]

[illegible]:

```
import java.awt.*;

class Test
```

[illegible]

[illegible]	[illegible] (#)

[illegible]

lines per	methods	avg. execution

[illegible]

- [illegible]

[illegible]

[illegible]:

- [illegible]

- [illegible]

```
class A extends Object {
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

- [illegible]

[illegible]

- [illegible]

B[illegible] a type propagation graph, [illegible]:

[illegible]

[illegible]); [illegible]

[illegible]