

Meðaltal talna

Lesum inn straum af tölum og skrifum út meðaltal

```
public class Average {  
    public static void main(String[] args) {  
        double sum = 0.0; // cumulative total  
        int n = 0; // number of values  
  
        while (!StdIn.isEmpty()) {  
            double x = StdIn.readDouble();  
            sum = sum + x;  
            n++;  
        }  
  
        StdOut.println(sum / n);  
    }  
}
```

```
% java Average  
10.0 5.0 6.0  
3.0 7.0 32.0  
<Ctrl-d>  
10.5
```

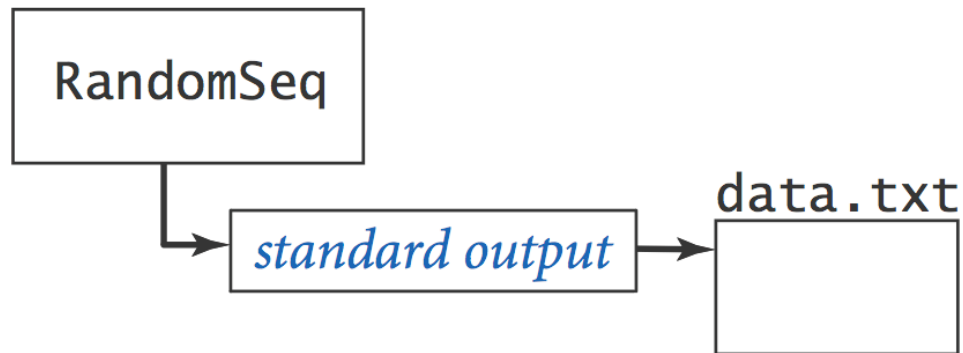
<Ctrl-d> fyrir mac/linux

<Ctrl-z> fyrir Windows

Ótakmarkaður fjöldi talna

Tenging forrita

Getum stýrt úttaki með pípum

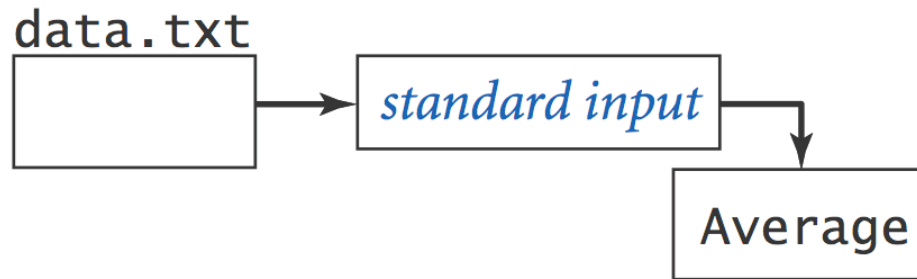


```
% java RandomSeq 1000 > data.txt
```

↑
pípum stdout yfir í
data.txt

Tenging forrita

Getum stýrt staðalinntaki inn í forrit



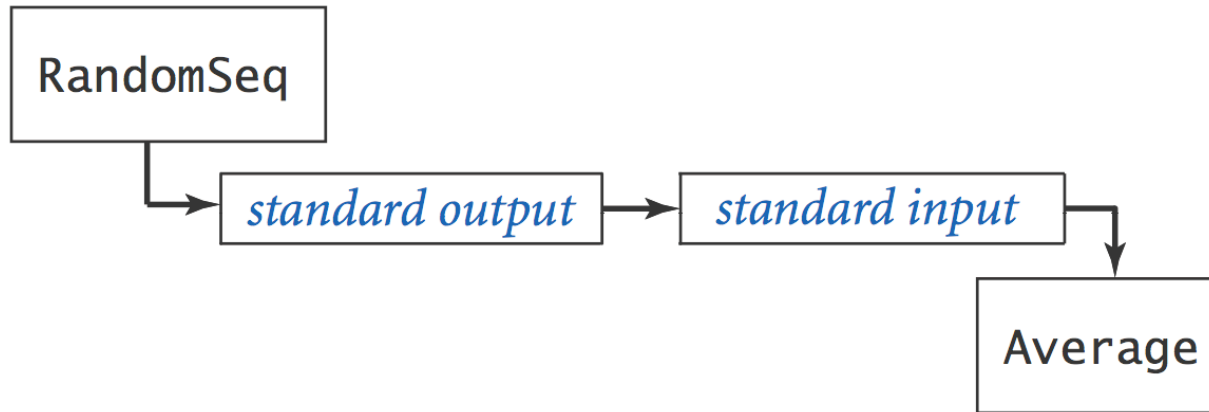
```
% less < data.txt  
0.5475375782884312  
0.4971087292684019  
0.23123808041753813  
...
```

pípum staðalinntaki inn í forrit

```
% java Average < data.txt  
0.4947655567740991
```

Tenging forrita

Getum tengt úttak eins forrits við inntak annars



Pípum úttaki `RandomSeq` inn í inntak `Average`

```
% java RandomSeq 1000000 | java Average  
0.4997970473016028  
  
% java RandomSeq 1000000 | java Average  
0.5002071875644842
```

Teikningar

Í verkefni í næstu viku munum við láta forrit teikna. Notum `StdDraw.java` sem fylgir bókinni

- Sér um að búa til gluggaforrit
- Getur teiknað línur, punkta, texta, hringi
- Vistar myndir sem png/jpg
- Hægt að búa til einfaldar hreyfimyndir

Mörg dæmi í bókinni, þurfum að setja `StdDraw.java` í sömu möppu og forrit.

Teikningar

```
public class StdDraw
```

```
void line(double x0, double y0, double x1, double y1)
void point(double x, double y)
void text(double x, double y, String s)
void circle(double x, double y, double r)
void filledCircle(double x, double y, double r)
void square(double x, double y, double r)
void filledSquare(double x, double y, double r)
void polygon(double[] x, double[] y)
void filledPolygon(double[] x, double[] y)

void setXscale(double x0, double x1)    reset x range to (x0, x1)
void setYscale(double y0, double y1)    reset y range to (y0, y1)
void setPenRadius(double r)             set pen radius to r
void setPenColor(Color c)               set pen color to c
void setFont(Font f)                    set text font to f
void setCanvasSize(int w, int h)        set canvas to w-by-h window
void clear(Color c)                     clear the canvas; color it c
void show(int dt)                       show all; pause dt milliseconds
void save(String filename)              save to a .jpg or w.png file
```

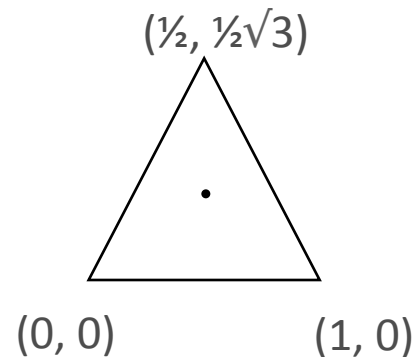
Note: Methods with the same names but no arguments reset to default values.

Teikningar

Dæmi: þríhyrningur

```
public class Triangle {  
    public static void main(String[] args) {  
        double t = Math.sqrt(3.0) / 2.0;  
        StdDraw.line(0.0, 0.0, 1.0, 0.0);  
        StdDraw.line(1.0, 0.0, 0.5, t);  
        StdDraw.line(0.5, t, 0.0, 0.0);  
        StdDraw.point(0.5, t/3.0);  
    }  
}
```

```
% java Triangle
```



Teikningar

Dæmi: Punktasafn, lesum inn (x,y) hnit af staðalinntaki og teiknum punkt

```
public class PlotFilter {
    public static void main(String[] args) {
        double xmin = StdIn.readDouble();
        double ymin = StdIn.readDouble();
        double xmax = StdIn.readDouble();
        double ymax = StdIn.readDouble();
        StdDraw.setXscale(xmin, xmax);
        StdDraw.setYscale(ymin, ymax);

        while (!StdIn.isEmpty()) {
            double x = StdIn.readDouble();
            double y = StdIn.readDouble();
            StdDraw.point(x, y);
        }
    }
}
```

← skölum hnitakerfi

← Lesum inn punkta
og teiknum

Teikningar

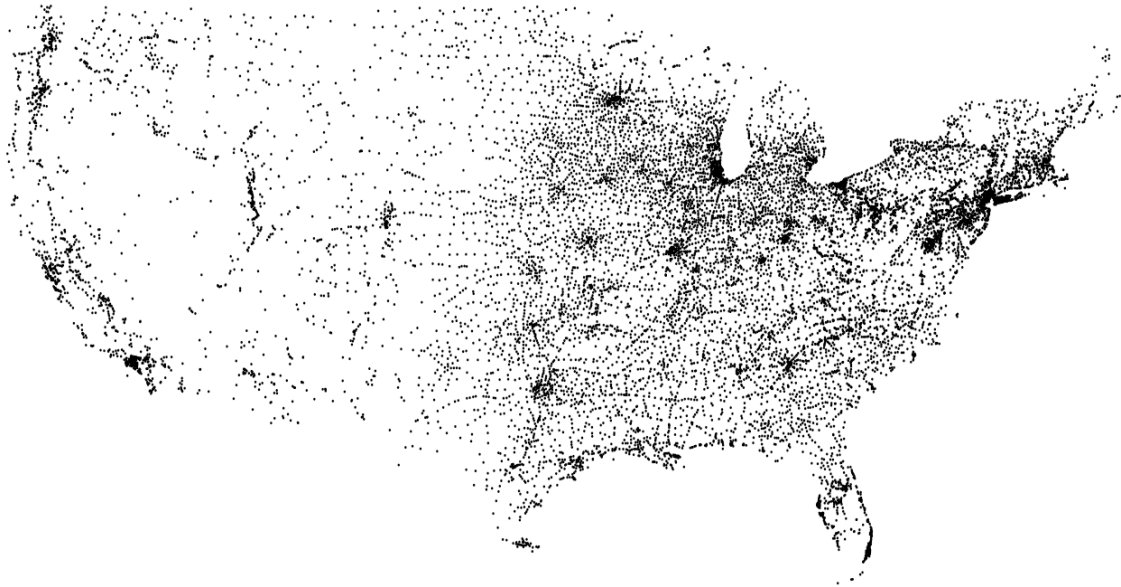
Allar borgir í BNA

```
% more < USA.txt
669905.0 247205.0 1244962.0 490000.0
1097038.8890 245552.7780
1103961.1110 247133.3330
1104677.7780 247205.5560
...

% java PlotFilter < USA.txt
```

Ytri mörk hnita

Hnit fyrir 13509 borgir



Föll

Java föll (function/static method)

- Taka 0 eða fleiri viðföng
- Skila 0 eða einu gildi
- Geta haft hliðarverkanir (t.d. prenta á skjá)

Föll í stærðfræði

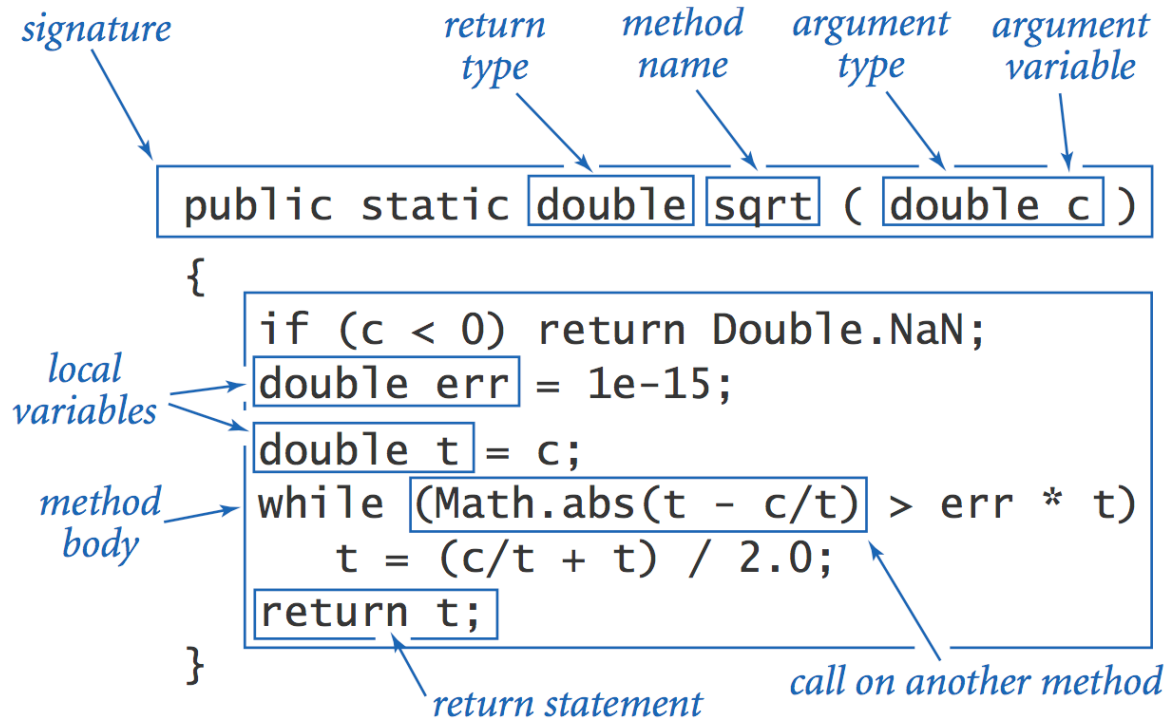
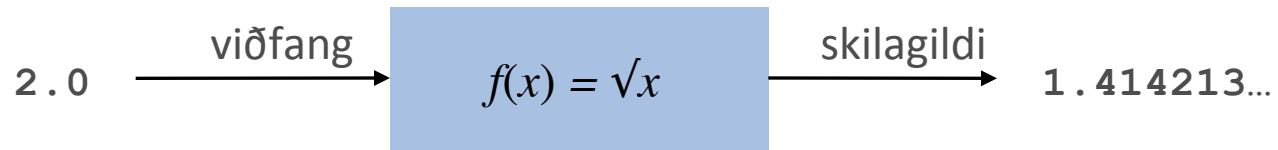
- Taka 1 eða fleiri viðföng
- Skila einu gildi
- Engar hliðarverkanir!

Dæmi

- Innbyggð föll: `Math.random()`, `Math.abs()`, `Integer.parseInt()`
- I/O föll: `System.out.println()`, `StdDraw.line()`
- Föll sem við skrifum: `main()`

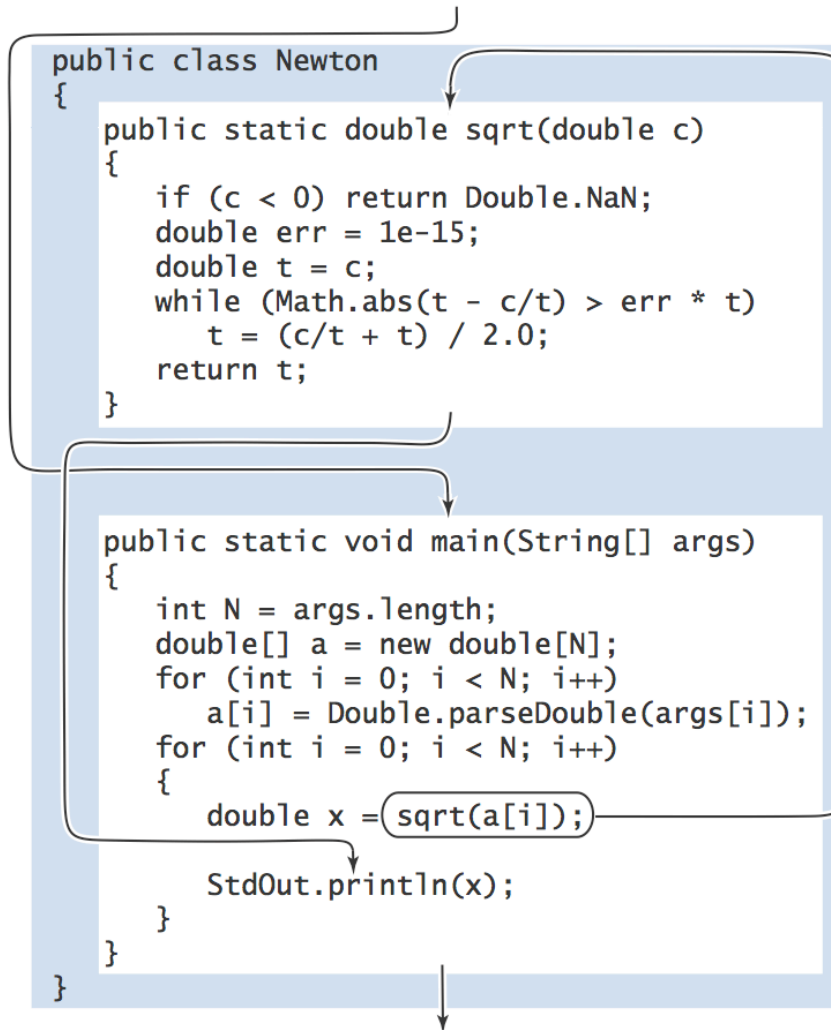
Föll

Auðvelt að skrifa föll í Java



Föll

Föll leyfa okkur að stýra flæði forrits



Föll

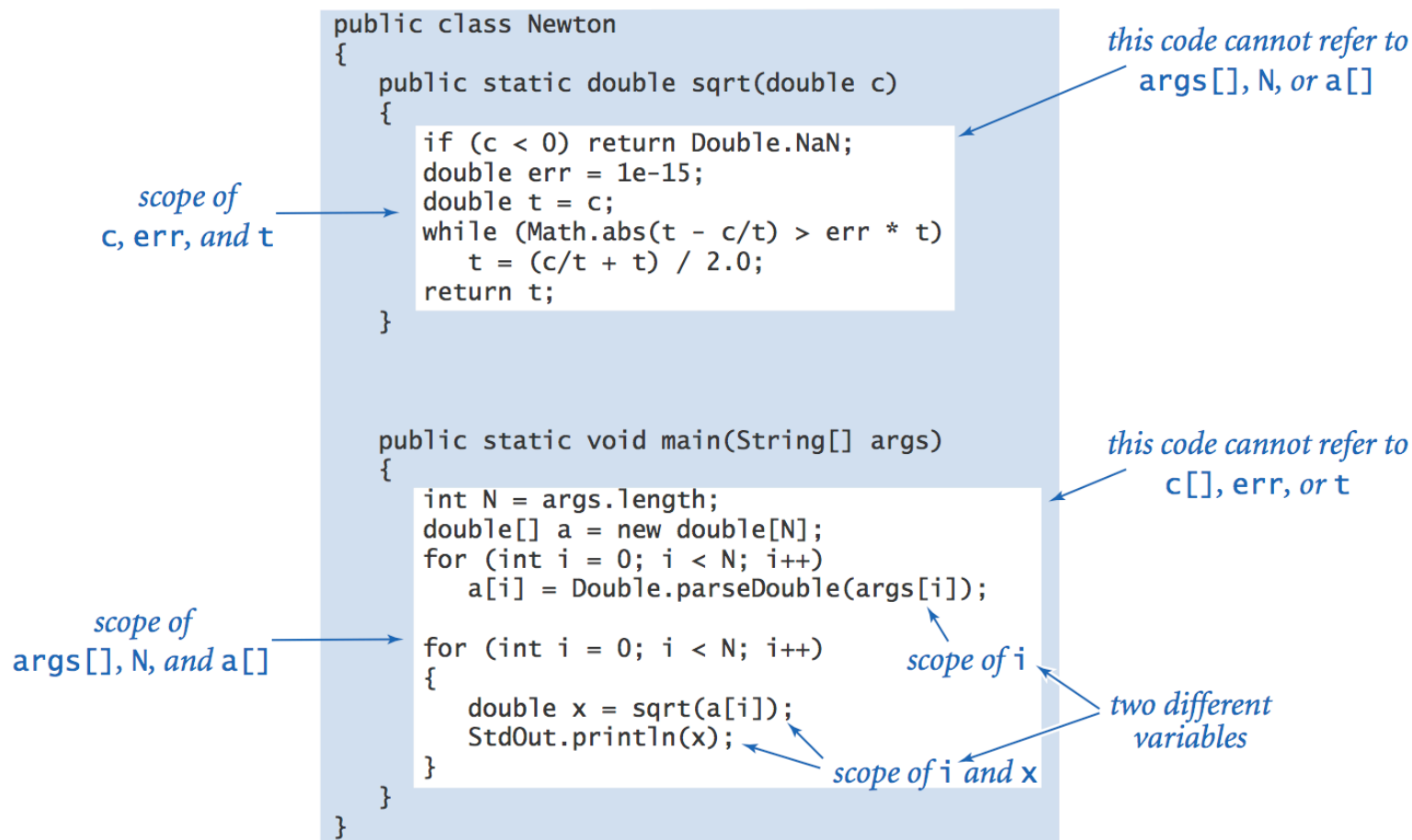
Föll leyfa okkur að stýra flæði forrits

Þegar við köllum á fall

1. Keyrsla forrits færast efst í fallið
2. Viðfangsbreytur fá upphafsgildi
3. Forritskafli fallsins er keyrður
4. Skilagildi falls er skilað þar sem kallað var frá
5. Keyrsla forrits heldur áfram þar sem frá var horfið

Gildissvið

Gildissvið breytu er sá hluti forrits sem getur vísað í hana (sjá betur bls. 188-193).



Hvað gerir þetta forrit?

```
public class Cubes1 {  
    public static int cube(int i) {  
        int j = i * i * i;  
        return j;  
    }  
  
    public static void main(String[] args) {  
        int N = Integer.parseInt(args[0]);  
        for (int i = 1; i <= N; i++)  
            StdOut.println(i + " " + cube(i));  
    }  
}
```

```
% javac Cubes1.java  
% java Cubes1 6  
1 1  
2 8  
3 27  
4 64  
5 125  
6 216
```

Hvað gerir þetta forrit?

```
public class Cubes1 {  
    public static int cube(int i) {  
        int i = i * i * i;  
        return i;  
    }  
  
    public static void main(String[] args) {  
        int N = Integer.parseInt(args[0]);  
        for (int i = 1; i <= N; i++)  
            StdOut.println(i + " " + cube(i));  
    }  
}
```


Hvað gerir þetta forrit?

```
public class Cubes1 {  
    public static int cube(int i) {  
        i = i * i * i;  
    }  
  
    public static void main(String[] args) {  
        int N = Integer.parseInt(args[0]);  
        for (int i = 1; i <= N; i++)  
            StdOut.println(i + " " + cube(i));  
    }  
}
```

Hvað gerir þetta forrit?

```
public class Cubes1 {  
    public static int cube(int i) {  
        i = i * i * i;  
        return i;  
    }  
  
    public static void main(String[] args) {  
        int N = Integer.parseInt(args[0]);  
        for (int i = 1; i <= N; i++)  
            StdOut.println(i + " " + cube(i));  
    }  
}
```

Hvað gerir þetta forrit?

```
public class Cubes1 {  
    public static int cube(int i) {  
        return i * i * i;  
    }  
  
    public static void main(String[] args) {  
        int N = Integer.parseInt(args[0]);  
        for (int i = 1; i <= N; i++)  
            StdOut.println(i + " " + cube(i));  
    }  
}
```

Gildissvið

Fall getur aðeins vísað í viðfangsbreytur og breytur skilgreindar í eigin forritskóða

```
public class Cubes1 {  
    public static int cube(int i) {  
        return i * i * i;  
    }  
  
    public static void main(String[] args) {  
        int N = Integer.parseInt(args[0]);  
        for (int i = 1; i <= N; i++)  
            StdOut.println(i + " " + cube(i));  
    }  
}
```

Viðfangsbreyta í falli getur heitið sama nafni og breyta annars staðar

Föll og rökstudd forritun

Í rökstuddri forritun lýsum við föllum með fastayrðingum

1. Notkun: sýnum hvernig á að nota fallið, gefum breytum heiti
2. Fyrir: Hvað þarf að gilda um viðföngin
3. Eftir: Hvað gildir um skilagildið og mögulega hliðarverkanir

```
// Notkun: x = sqrt(c);
// Fyrir: c >= 0
// Eftir: x * x = c með nákvæmni 1e-15
public static double sqrt(double c) {
    if (c < 0) return Double.NaN;
    double EPS = 1E-15;
    double t = c;
    while (Math.abs(t - c/t) > EPS*t)
        t = (c/t + t) / 2.0;
    return t;
}
```

Föll og gildissvið

Forrit og ástand þess verður einfaldara ef við brjótum niður afmörkuð verkefni í föll

Gildissvið breytu er sá forritskóði sem getur lesið og breytt henni

Við getum endurnýtt föll í öðrum forritum, rétt eins og `Integer.parseInt()` nýtist aftur og aftur. Mun betra en copy/paste

Föll og gildissvið

```
public class D {  
    public static int f(int a) {  
        int x,c;  
        // ... reiknar ...  
        return x;  
    }  
    public static int g(int a) {  
        int b,c;  
        // ... reiknar ...  
        b = f(c);  
    }  
}
```

Hvert er gildissvið a,b,c,x?

Hvaða breytur getur f haft áhrif á?

Hvaða breytur getur g haft áhrif á?

Föll og hliðarverkanir

Það er auðveldara að skilja og sanna fall með engar hliðarverkanir.

Dæmi um hliðarverkanir:

- Skrifna út á stdout
- Lesa af stdin
- `Math.random()` (vægar hliðarverkanir)
- Breyta fylkjum
- Breyta hlutum (* seinna í námskeiðinu)

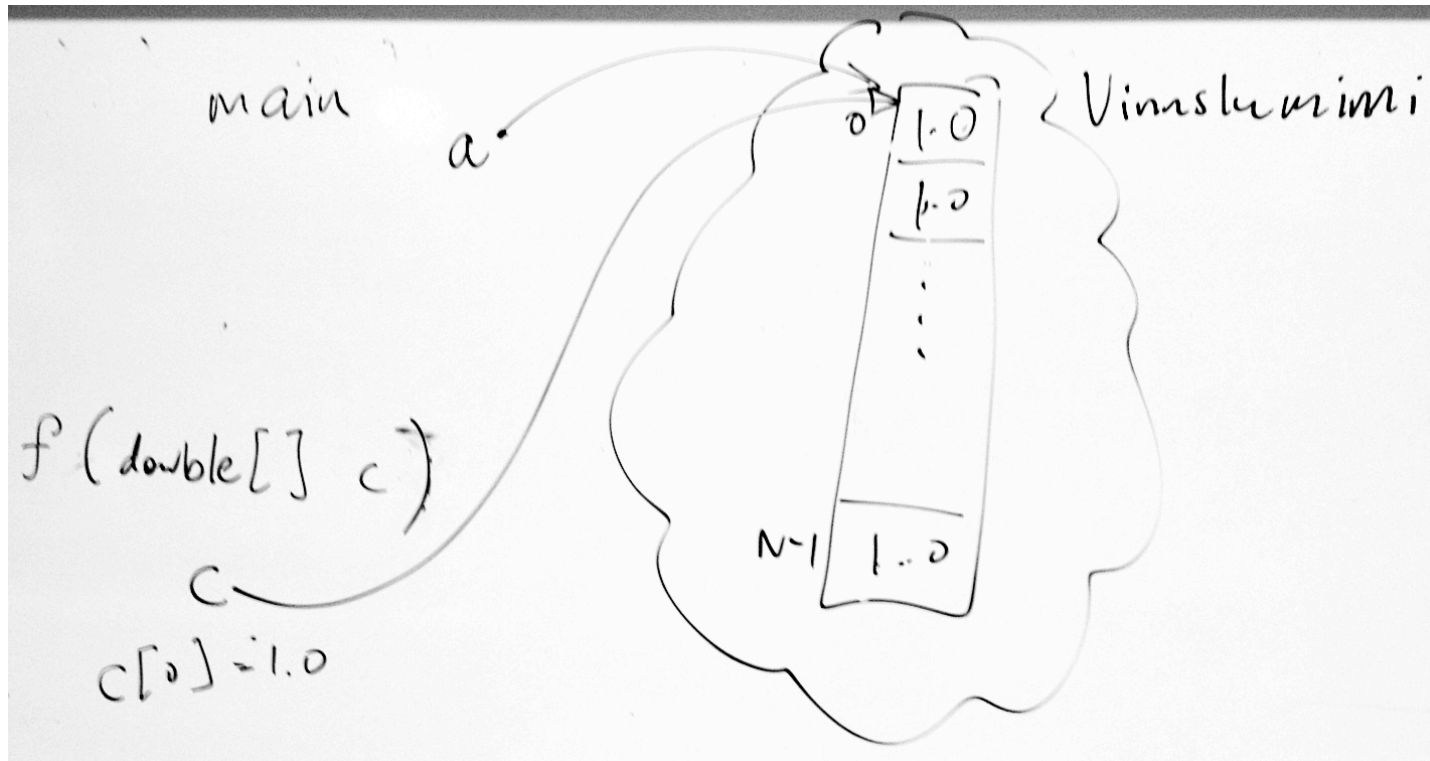
Hvað gerir eftirfarandi kóði?

```
public static void f(double[] c) {
    for (int i = 0; i < c.length; i++) {
        c[i] = 1.0;
    }
}

public static void main (String[] args) {
    double[] a = new double[N];
    / ... prentum a, 0.0 N sinnum
    f(a);
    // ... prentum a, ???
}
```

Gildi fylkja

Gildið á fylkinu a er geymt sem tilvísun í minnishólfuð þar sem gögnin eru geymd. Fylkið c í fallinu f fær því sama gildi, þ.e. bendir á sama stað í minni.



Föll og fylki

Þegar fylki er notað sem viðfang í falli hafa breytingar áhrif fyrir utan fallið, ólíkt því sem gerist með `int`, `double` og `boolean` viðföng.

Ef fylki breytist í falli þá þarf að taka það fram í útskýringu á fallinu.

```
// Notkun: f(c)
// Fyrir:  ekkert
// Eftir:  c[i] = 1 fyrir öll i
public static void f(double[] c) {
    for (int i = 0; i < c.length; i++) {
        c[i] = 1.0;
    }
}
```