



# Tölvunarfræði 1

## Fyrirlestur 16: Forritasöfn í Java II

Hjálmtyr Hafsteinsson  
Haust 2015





# Í síðasta fyrirlestri

- Forritasöfn (*libraries*)
  - Skipulagning falla í forritasöfn
  - Forritaskil (*API*)
- Dæmi um forritasöfn
  - **Gaussian**
  - **StdRandom**

**Kafli 2.2**



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-  
OG TÖLVUNARFRÆÐIDEILD



# Í þessum fyrirlestri

- Einingaprófun (*unit testing*)
- Stærri dæmi um forritasöfn
  - Gagnagreining (`StdStats`)
  - Inntak/úttak fyrir fylki (`StdArrayIO`)
  - Ítruð föll (`IFS`)
- Einingaforritun (*modular programming*)

**Kaflí 2.2**



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-  
OG TÖLVUNARFRÆÐIDEILD



# Einingaprófun (*unit testing*)

- Prófun á því hvort einstakir klasar eða föll innan hans vinna eins og þeir eiga að gera
- Framkvæmd (í Java):
  - Höfum `main`-fall í klasanum sem prófar öll föll í honum
  - Keyrum fall á inntaki sem við vitum hvað á að koma út úr
  - Berum saman vænt úttak og úttak úr keyrslu

```
public static void main (String[] args) {  
    ...  
    if (annadveldi(2) != 4)  
        System.out.println("Galli í fallinu annadveldi!");  
    ...  
}
```

Ef fall skilar ekki því sem við búumst við, láta vita af því





# Einingaprófun í StdRandom

- Forritasöfn hafa `main`-fall sem prófar öll föllin:

```
public class StdRandom {  
    ...  
  
    public static void main(String[] args) {  
        int N = Integer.parseInt(args[0]);  
        for (int i = 0; i < N; i++) {  
            StdOut.printf(" %2d " , uniform(100));  
            StdOut.printf("%8.5f ", uniform(10.0, 99.0));  
            StdOut.printf("%5b " , bernoulli(.5));  
            StdOut.printf("%7.5f ", gaussian(9.0, .2));  
            StdOut.println();  
        }  
    }  
}
```

Viðfang á skipanalínu  
(`N`) segir hversu oft á  
að keyra öll föllin

```
% java StdRandom 5  
61 21.76541 true 9.30910  
57 43.64327 false 9.42369  
31 30.86201 true 9.06366  
92 39.59314 true 9.00896  
36 28.27256 false 8.66800
```

Ef við breytum einhverju falli  
í forritasafni þá keyrum við  
einingaprófunina til að sjá  
hvort við höfum nokkuð  
skemmt eitthvað



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-  
OG TÖLVUNARFRÆÐIDEILD



# Forritasafnið StdStats

- Forritasafn fyrir tölfræðireikninga og línu- og súlurit

```
public class StdStats
```

```
double max(double[] a)
```

*largest value*

```
double min(double[] a)
```

*smallest value*

```
double mean(double[] a)
```

*average*

```
double var(double[] a)
```

*sample variance*

```
double stddev(double[] a)
```

*sample standard deviation*

```
double median(double[] a)
```

*median*

Einföld  
teikniföll

```
void plotPoints(double[] a)
```

*plot points at (i, a[i])*

```
void plotLines(double[] a)
```

*plot lines connecting points at (i, a[i])*

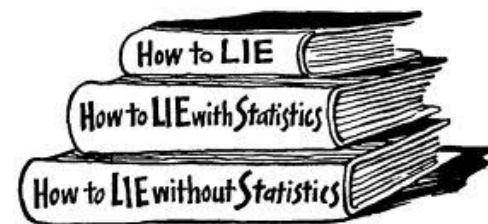
```
void plotBars(double[] a)
```

*plot bars to points at (i, a[i])*



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-  
OG TÖLVUNARFRÆÐIDEILD





# Einfalt súlurit

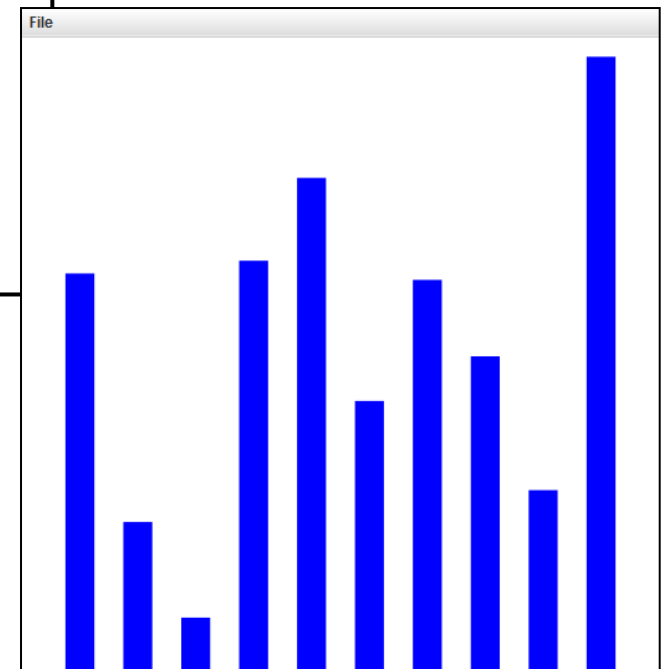
```
public class SuluDemo {  
    public static void main(String[] args) {  
        int N = Integer.parseInt(args[0]);  
  
        StdDraw.setYscale(0, 100);  
        StdDraw.setPenColor(StdDraw.BLUE);  
  
        double[] a = new double[N];  
        for (int i = 0; i < N; i++) {  
            a[i] = StdRandom.uniform(100);  
        }  
        StdStats.plotBars(a);  
    }  
}
```

```
% java SuluDemo 10
```

Kvarða y-ásinn rétt

Teikna bláar súlur

Setja slembigildi frá 0 til 100 í fylkið

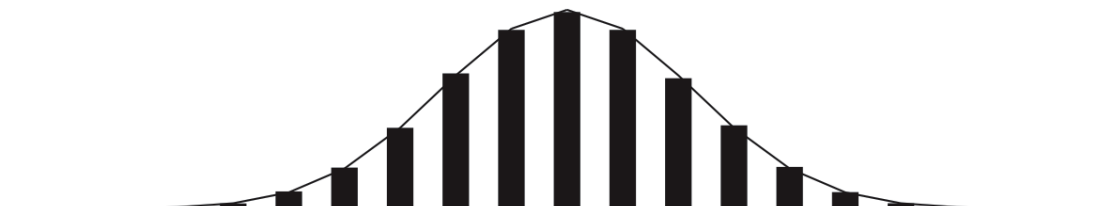




# Hermun á hlutkesti

- Köstum  $N$  krónum
- Hversu margar af þeim lenda með þorskinn upp (*heads*)?
- Stærðfræðin segir okkur að fjöldinn sé normaldreifður með meðaltal  $N/2$  og staðalfrávik  $\sqrt{N} / 2$
- Framkvæmum hermun til að athuga hvort það standist
- Teiknum síðan niðurstöðurnar saman

% java Bernoulli 20 100000



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-  
OG TÖLVUNARFRÆÐIDEILD





```
public class Bernoulli {  
    public static int binomial(int N) {  
        int heads = 0;  
        for (int j = 0; j < N; j++)  
            if (StdRandom.bernoulli(0.5)) heads++;  
        return heads;  
    }  
}
```

Kasta  $N$  krónum, skila  
fjölda þorska

```
public static void main(String[] args) {  
    int N = Integer.parseInt(args[0]);  
    int T = Integer.parseInt(args[1]);
```

Framkvæma  $T$  hermanir og  
geyma fjölda þorska sem  
komu upp í hvert sinn

```
    int[] freq = new int[N+1];  
    for (int i = 0; i < T; i++)  
        freq[binomial(N)]++;
```

```
    double[] normalized = new double[N+1];  
    for (int i = 0; i <= N; i++)  
        normalized[i] = (double) freq[i] / T;  
    StdStats.plotBars(normalized);
```

Teikna niðurstöðu  
hermunar sem súlurit

```
    double mean = N / 2.0, stddev = Math.sqrt(N) / 2.0;  
    double[] phi = new double[N+1];  
    for (int i = 0; i <= N; i++)  
        phi[i] = Gaussian.phi(i, mean, stddev);  
    StdStats.plotLines(phi);  
}
```

Finna fræðilega ferilinn  
og teikna hann líka



# Inntak og úttak fyrir fylki

- Vinnum mikið með fylki
  - Dágóður kóði í kringum að lesa inn og skrifa út fylki
  - Gott að hafa föll sem gera það fyrir okkur

```
public class StdArrayIO
```

---

<code>double[]</code>	<code>readDouble1D()</code>	<i>read a one-dimensional array of double values</i>
<code>double[][]</code>	<code>readDouble2D()</code>	<i>read a two-dimensional array of double values</i>
<code>void</code>	<code>print(double[] a)</code>	<i>print a one-dimensional array of double values</i>
<code>void</code>	<code>print(double[][] a)</code>	<i>print a two-dimensional array of double values</i>

Notes:

1. 1D format is an integer  $N$  followed by  $N$  values.
2. 2D format is two integers  $M$  and  $N$  followed by  $M \times N$  values in row-major order.
3. Methods for `int` and `boolean` are also included.



HÁSKÓLI ÍSLANDS

ÍDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-  
OG TÖLVUNARFRÆÐIDEILD

Þurfum að ákveða hvernig  
gögnin eru geymd



# Fallið `readDouble1D()`

`StdArrayIO.java`

```
public class StdArrayIO {  
    ...  
  
    public static double[] readDouble1D() {  
        int N = StdIn.readInt();  
        double[] a = new double[N];  
        for (int i = 0; i < N; i++) {  
            a[i] = StdIn.readDouble();  
        }  
        return a;  
    }  
}
```

Fyrst kemur lengd  
einvíða fylkisins

Búa til fylki af réttri  
stærð og lesa inni það

`ProfaStdArrayIO.java`

```
public class ProfaStdArrayIO {  
    public static void main(String[] args) {  
  
        double[] a = StdArrayIO.readDouble1D();  
        StdArrayIO.print(a);  
        StdOut.println();  
    }  
}
```



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-  
OG TÖLVUNARFRÆÐIDEILD



# Fallið `print(double[] a)`

[StdArrayIO.java](#)

```
public class StdArrayIO {  
    ...  
  
    public static void print(double[] a) {  
        int N = a.length;  
        StdOut.println(N);  
        for (int i = 0; i < N; i++) {  
            StdOut.printf("%9.5f ", a[i]);  
        }  
        StdOut.println();  
    }  
    ...  
}
```

Prentum fyrst út  
lengd fylkisins

Prentum svo út öll  
stökin í sömu línu

```
% more fylki1.txt  
4  
5.3  2.1  8  6.9  
% java ProfasStdArrayIO <fylki1.txt  
4  
5.30000  2.10000  8.00000  6.90000
```

Getum notað úttakið sem inntak í  
annað forrit sem notar þessi föll



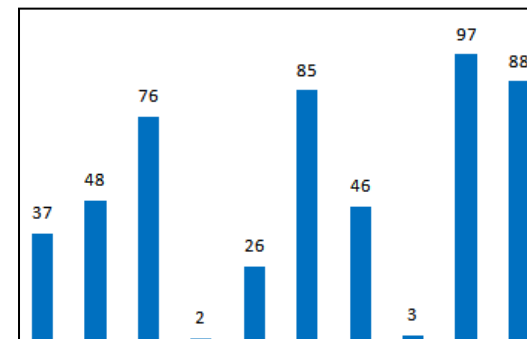
HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-  
OG TÖLVUNARFRÆÐIDEILD



# Fyrirlestraræfing

1. Hvernig væri einfalt einingapróf fyrir `min`-fallið í `stdStats` forritasafninu?
2. Hvernig myndum við setja gildin ofaná súlurnar í súluritinu frá `stdStats.plotBars`?  
(Bara útskýra, ekki forrita)



3. Hvers vegna þarf að lesa fyrst stærð fylkisins í föllum forritasafnsins `stdArrayIO`? Væri mögulegt að skrifa föllin þannig að þess þyrfti ekki?



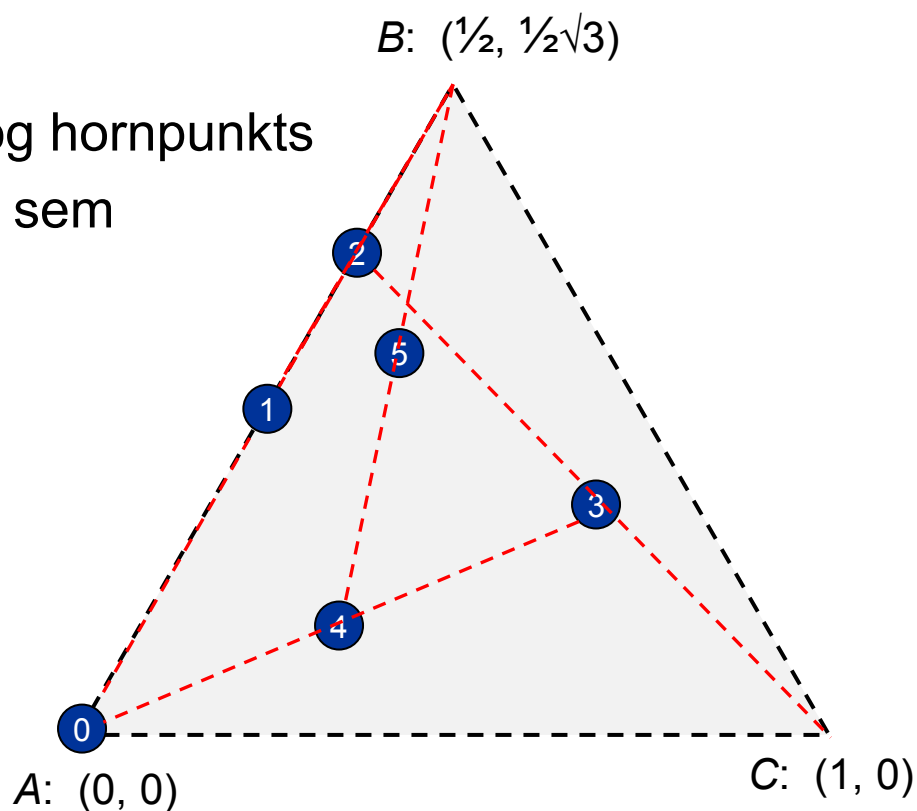
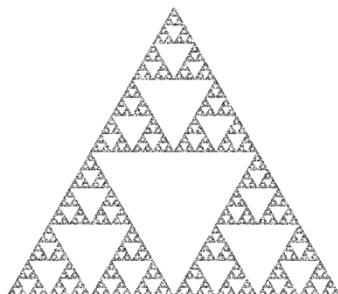


# Óreiðuleikurinn (*Chaos game*)

- Spila á jafnarma þríhyrningi með hornpunkta  $A$ ,  $B$ ,  $C$ 
  - Byrja í  $A$  (núverandi punktur)
  - Endurtaka eftirfarandi  $N$  sinnum:
    - velja hornpunkt af handahófi
    - fara miðja vegu milli núv. punkts og hornpunkts
    - teiknar punkt á þeim stað og setja sem núverandi punkt

- Hvaða mynd kemur?

$B \ B \ C \ A \ B \ \dots$





# Útfærsla í Java

```
public class Sierpinski {  
    public static void main(String[] args) {  
        int N = Integer.parseInt(args[0]);  
        double[] cx = { 0.000, 1.000, 0.500 };  
        double[] cy = { 0.000, 0.000, 0.866 };  
  
        double x = 0.0, y = 0.0;  
  
        for (int i = 0; i < N; i++) {  
            int r = StdRandom.uniform(3);  
            x = (x + cx[r]) / 2.0;  
            y = (y + cy[r]) / 2.0;  
            StdDraw.point(x, y);  
        }  
    }  
}
```

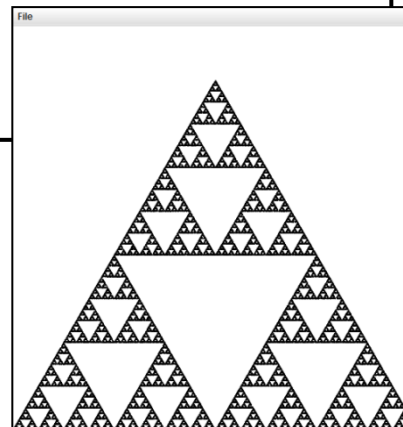
Lesað fjölda ítrana

Hornpunktir  
þríhyrningsins

Byrjum í  $A = (0, 0)$

Velja einn hornpunktinn  
af handhófi

Finna punktinn mitt á  
milli hans og núv. punkts



HÁSKÓLI ÍSLANDS

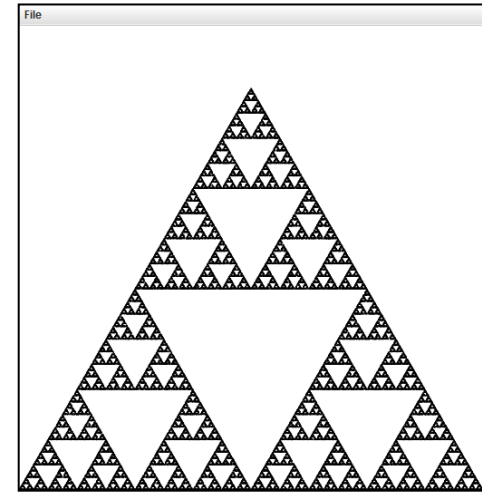
IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-  
OG TÖLVUNARFRÆÐIDEILD



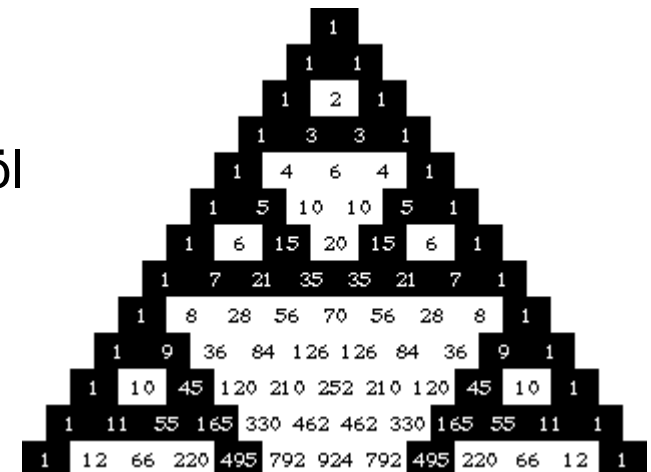
# Þríhyrningur Sierpinski

- Hægt að búa hann til á nokkra aðra vegu:
  - Endurkvæmt (*recursive*):
    - Hver þríhyrningur samanstendur af 4 þríhyrningum, þrír fylltir og einn tómur (á hvolfi)

Sjáum í næstu viku



- Með Pascal þríhyrningi:
  - Búa til Pascal þríhyrning og lita allar oddatöl svartar, en jafnar tölur hvítar



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-  
OG TÖLVUNARFRÆÐIDEILD

([Meira um þríhyrning Sierpinski](#))

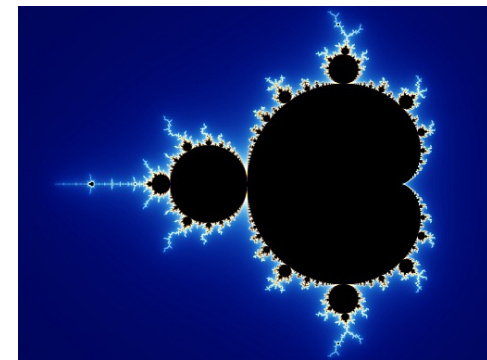




# Ítruð fallakerfi (*IFS*)

- Ítruð fallakerfi (*iterated functions systems*) er safn falla sem geta gefið okkur fallegar brotamyndir
  - Höfum föll  $f_i^x(x, y)$  og  $f_i^y(x, y)$
  - Setjum  $(x, y) = (f_i^x(x, y), f_i^y(x, y))$
- Brotamyndir (*fractals*) eru myndir sem eru á einhvern hátt eins á mismunandi kvörðum
  - Dæmi: Mandelbrot mengi

*i*-ið er valið af handahófi með tiltekinni líkindadreifingu





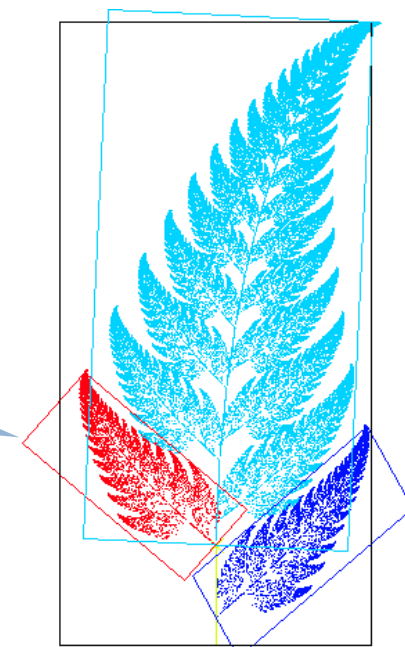
# Dæmi um fallakerfi

- Barnsley burkni (*Barnsley fern*)
  - Hefur 4 föll með mismunandi líkum



probability	$x$ -update		$y$ -update	
1%	$x =$	.500	$x =$	.160 $y$
85%	$x =$	$.85x + .04y + .075$	$y =$	$.04x + .85y + .180$
7%	$x =$	$.20x - .26y + .400$	$y =$	$.23x + .22y + .045$
7%	$x =$	$.15x + .28y + .575$	$y =$	$.26x + .24y - .086$

Þegar við þysjum (zoom) inn í myndina þá er þetta alltaf sama formið





# Forrit fyrir ítruð fallakerfi

- Okkar föll eru öll fyrsta stigs föll af tveimur breytistærðum ( $x$  og  $y$ )
  - Skilgreina þau með gagnaskrá
  - Forrit les gagnaskránnu og teiknar myndina

$y$ -föllin sjálf

$$f_0^y(x, y) = 0.16y$$

$$f_1^y(x, y) = -0.04x + 0.85y + 0.180$$

$$f_2^y(x, y) = 0.23x + 0.22y + 0.045$$

$$f_3^y(x, y) = 0.26x + 0.24y - 0.086$$

barnsley.txt

4				
	0.01	0.85	0.07	0.07
4	3			
	0.00	0.00	0.500	
	0.85	0.04	0.075	
	0.20	-0.26	0.400	
	-0.15	0.28	0.575	
4	3			
	0.00	0.16	0.000	
	-0.04	0.85	0.180	
	0.23	0.22	0.045	
	0.26	0.24	-0.086	

Fjöldi falla (hér 4)

Líkur á hverju falli

Stuðlar á  $x$ -föllum

Stuðlar á  $y$ -föllum



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-  
OG TÖLVUNARFRÆÐIDEILD



# Forritið IFS.java

```
public class IFS {  
    public static void main(String[] args) {  
  
        int T = Integer.parseInt(args[0]);  
  
        double[] dist = StdArrayIO.readDouble1D();  
        double[][] cx = StdArrayIO.readDouble2D();  
        double[][] cy = StdArrayIO.readDouble2D();  
  
        double x = 0.0, y = 0.0;  
  
        for (int t = 0; t < T; t++) {  
            int r = StdRandom.discrete(dist);  
            double x0 = cx[r][0]*x + cx[r][1]*y + cx[r][2];  
            double y0 = cy[r][0]*x + cy[r][1]*y + cy[r][2];  
            x = x0;  
            y = y0;  
  
            StdDraw.point(x, y);  
        }  
    }  
}
```

Fjöldi ítrana kemur  
af skipanalínu

Lesi líkindafylki

Lesi fallastuðla -  
tvívíð fylki

Upphafsgildi (0,0)

Velja fall af handahófi

Beita fallinu á (x,y)

Teikna nýjan punkt



# Dæmi um *IFS*

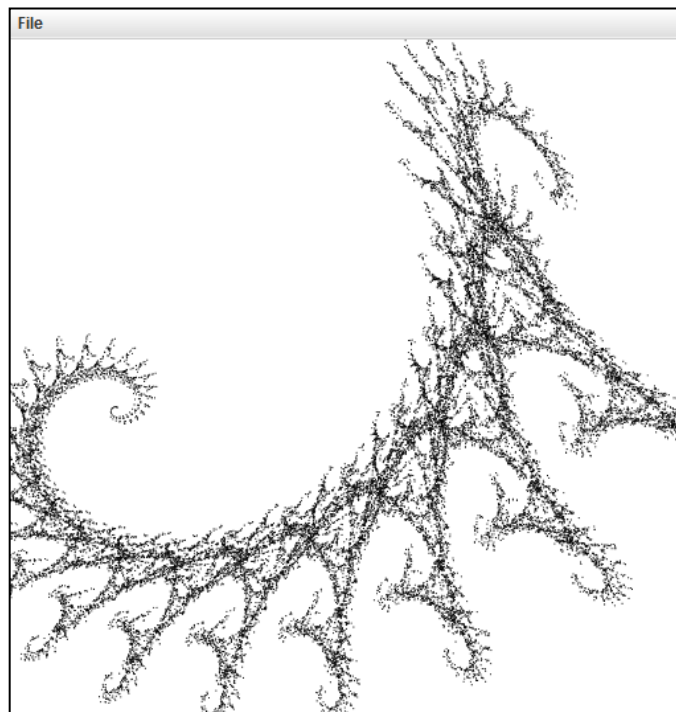
dragon.txt

```
2
    0.787473  0.212527

2 3
    0.824074  0.281482 -0.1002660
    0.088272  0.520988  0.5344000

2 3
    -0.212346  0.864198  0.0951123
    -0.463889 -0.377778  1.0415240
```

```
% java IFS 30000 <dragon.txt
```



Hægt að skoða keyrslu  
forritsins í [Java Visualizer](#)  
Kemur engin grafík

Útgáfa af [drekaferli](#) (*dragon curve*)



HÁSKÓLI ÍSLANDS

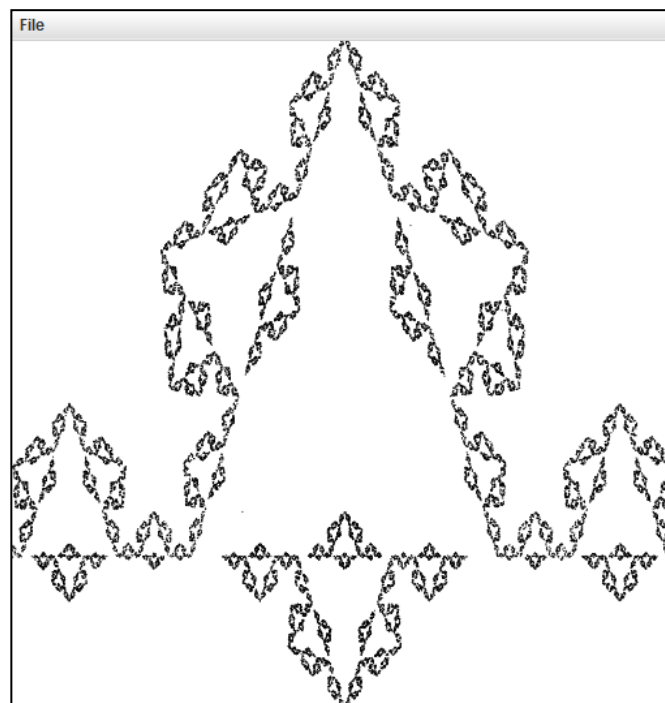
IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-  
OG TÖLVUNARFRÆÐIDEILD



# Dæmi um *IFS*

koch.txt

```
5
    0.151515  0.253788  0.253788
0.151515  0.189394
5 3
    0.307692 -0.000000  0.7580704
    0.192308 -0.205882  0.3349620
    0.192308  0.205882  0.4707040
    0.307692 -0.000000 -0.0674990
    0.307692 -0.000000  0.3453822
5 3
    0.000000  0.294118  0.1604278
    0.653846  0.088235  0.2709686
   -0.653846  0.088235  0.9231744
    0.000000  0.294118  0.1604278
    0.000000 -0.294118  0.2941176
```



```
% java IFS 30000 <koch.txt
```

Eftirlíking af [Koch snjókorni](#)



HÁSKÓLI ÍSLANDS

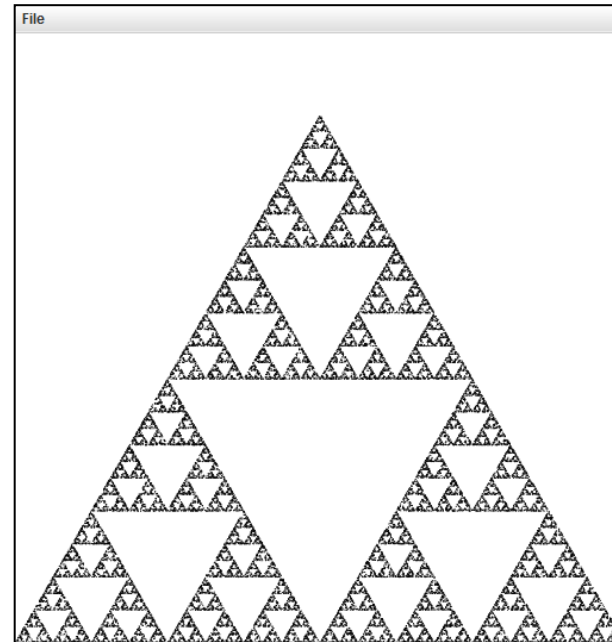
IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-  
OG TÖLVUNARFRÆÐIDEILD



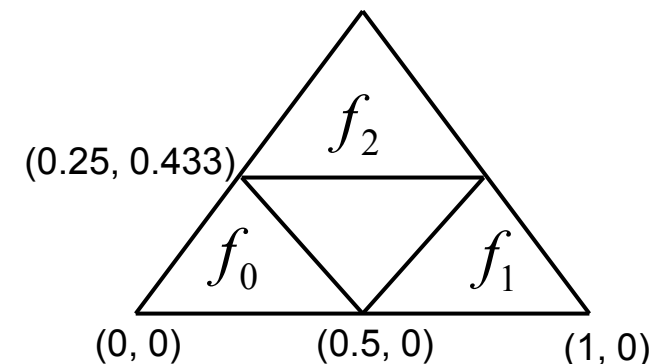
# Sierpinski þríhyrningur sem *IFS*

sierpinski.txt

```
3
  .33 .33 .34
3 3
  .50 .00 .00
  .50 .00 .50
  .50 .00 .25
3 3
  .00 .50 .00
  .00 .50 .00
  .00 .50 .433
```

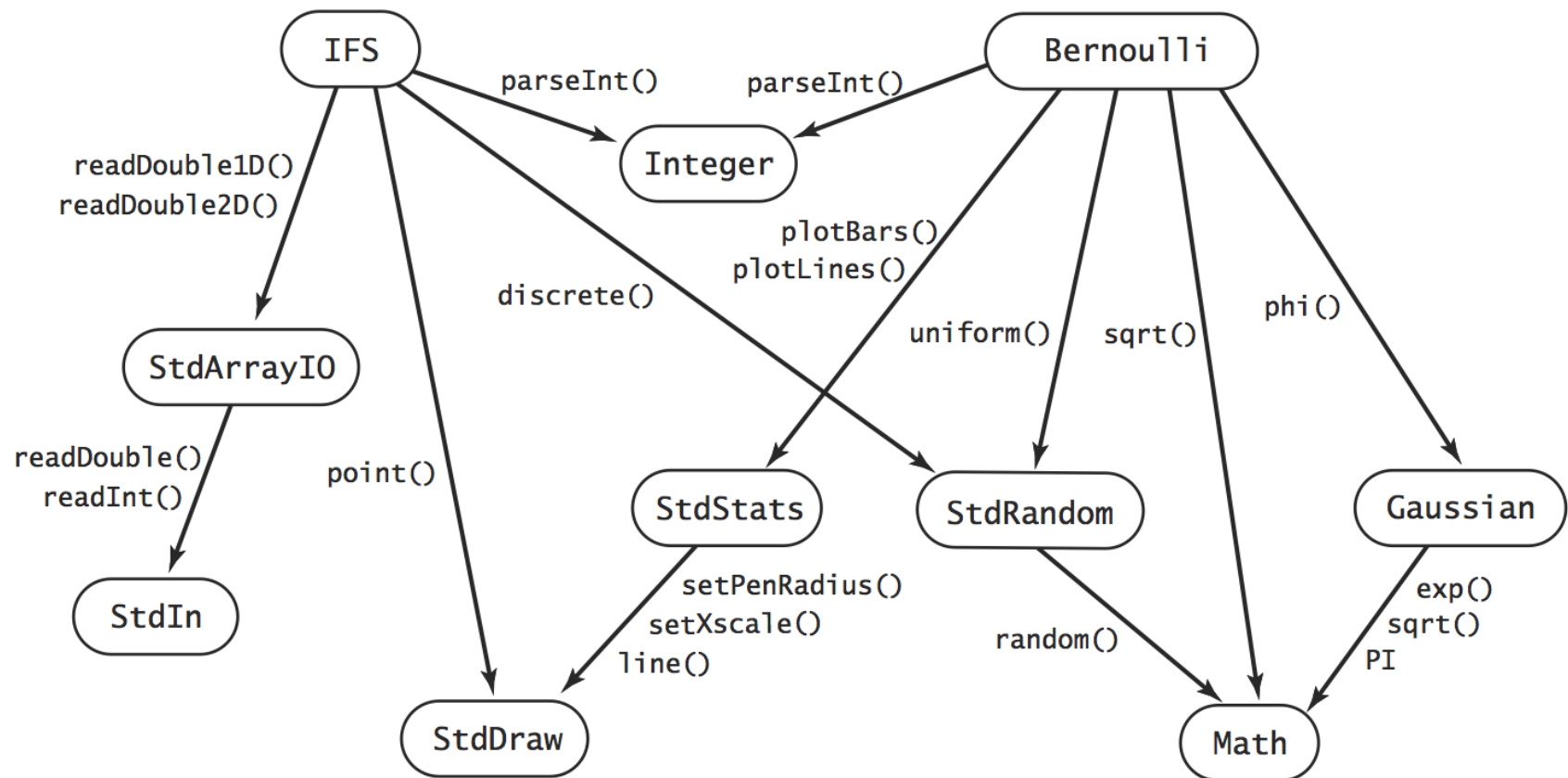


Samanstendur af 3 föllum, sem öll eru jafnlíkleg  
Hvert fall býr til einn af hinum þremur fylltu þríhyrningum





# Einingaforritun (*modular programming*)







## Fyrirlestraræfing

4. Í Java útfærslunni á Óreiðuleiknum, sem teiknar upp Sierpinski þríhyrninginn, er byrjað í punktinum  $(0, 0)$ . Hvað gerist ef við byrjum á einhverjum öðrum stað?
5. Það er til þrívíð útgáfa af Sierpinski þríhyrningnum sem samanstendur af pýramíðum. Hve mörgum?
6. Skrifið út föllins sem IFS notar til að búa til Sierpinski þríhyrninginn?





# Samantekt

- Í þessum tíma:
  - Stærri dæmi um forritasöfn **Kaflí 2.2**
  - Einingaforritun
- Í næsta tíma:
  - Endurkvæmni (*recursion*) **Kaflí 2.3**
  - Endurkvæm Java föll

