



Tölvunarfræði 1

Fyrirlestur 13: Föll í Java I

Hjálmtyr Hafsteinsson
Haust 2015





Í síðasta fyrirlestri

- Teikning með `stdDraw`
 - Línur og punktar
 - Kassar, hringir, texti
 - Hreyfing
- Hljóð með `stdAudio`

Kafli 1.5



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Í þessum fyrirlestri

- Föll (*static methods*) í Java
 - Tilgangur falla
 - Skilgreining og notkun
 - Dæmi
- Helstu eiginleikar falla

Kafli 2.1



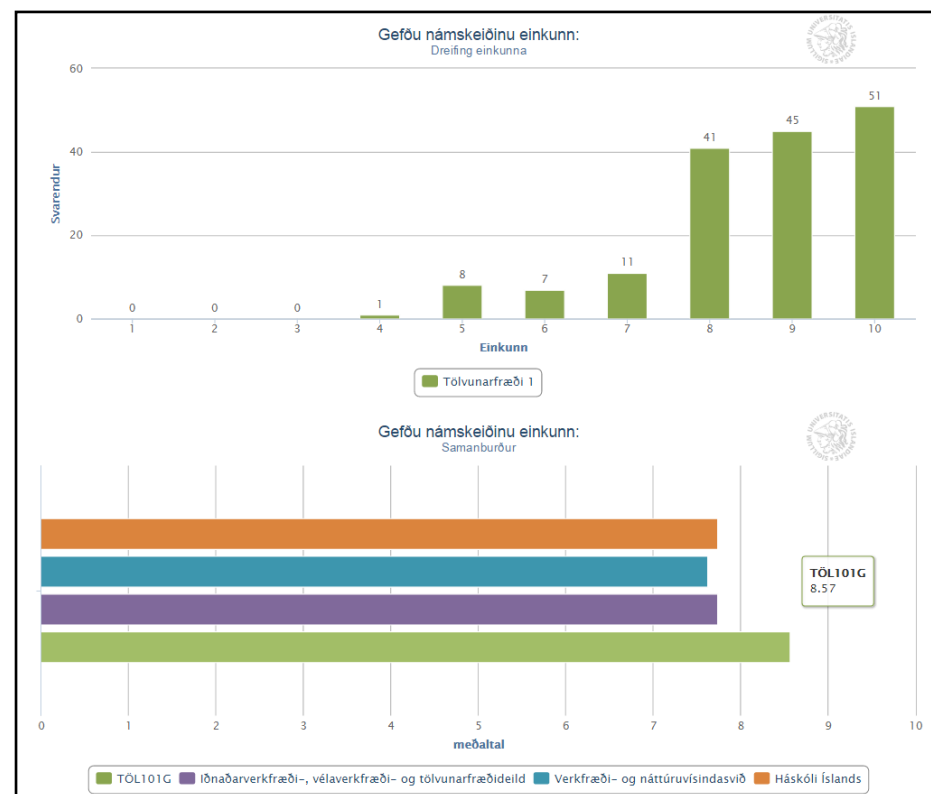
HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Miðmisseriskönnun

- Þokkaleg þátttaka:
 - 164 af 333 skráðum nemendum (~49%)
- Ánægja með:
 - Fyrirlestraæfingar
 - Glærur
- Ábendingar:
 - Heimadæmi að þyngjast
 - Fleiri léttari æfingar
 - Of þungar fyrirlestraæfingar
 - Stoðtíma
 - Sýna hvernig forrit verða til



HÁSKÓLI ÍSLANDS

ÍÐNAÐARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Viðbrögð við könnun

- Breytingar:
 - Meiri breidd í þyngd einstakra heimadæma
 - Benda á dæmi í bók til að gera aukalega
 - Hugsanlegt að hafa stoðtíma
- Ekki breyta:
 - Dagsetning heimadæma (koma á fös., skila á fös.)
 - Tímadæmi birtast í dæmatímum
 - Einkunnagjöf fyrir heimadæmi
 - Ekki farið yfir fyrirlestraæfingar

Byrja strax að reyna við dæmin!

Virka eins og mini-próf

Heimadæmi eru æfingadæmi - þau hafa ekki sjálfstæðan tilgang!



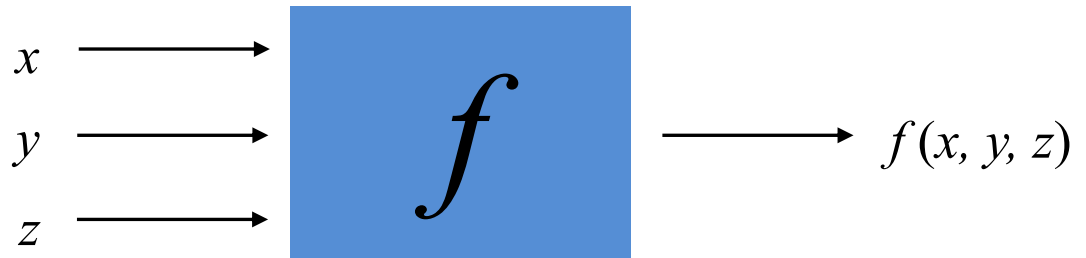
HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Föll (*functions*)

- Í stærðfræði:
 - Vensl (*relation*) á milli inntaksmengis og úttaksmengis
 - Varpar inntaksgildum yfir í eitt úttaksgildi
 - Dæmi: $f(x) = x^2$
 - Forritunarmál líkja eftir þessu fræðilega hugtaki





Föll í Java

- Eiginleikar:
 - Hafa núll eða fleiri viðföng (*arguments*)
 - Skila einu (eða engu) skilagildi (*return value*)
 - Geta hafa hliðaráhrif (*side effects*), t.d. teikning í teikniglugga, gefa hljóðmerki, skrifa í staðalúttak, ...
- Notkun:
 - Til að skilgreina og reikna úr formúlum
 - Til að skipuleggja forrit í einingar
 - Til að forðast endurtekningar (*redundancy*)

Ekki í samræmi
við stærðfræði-
útgáfuna





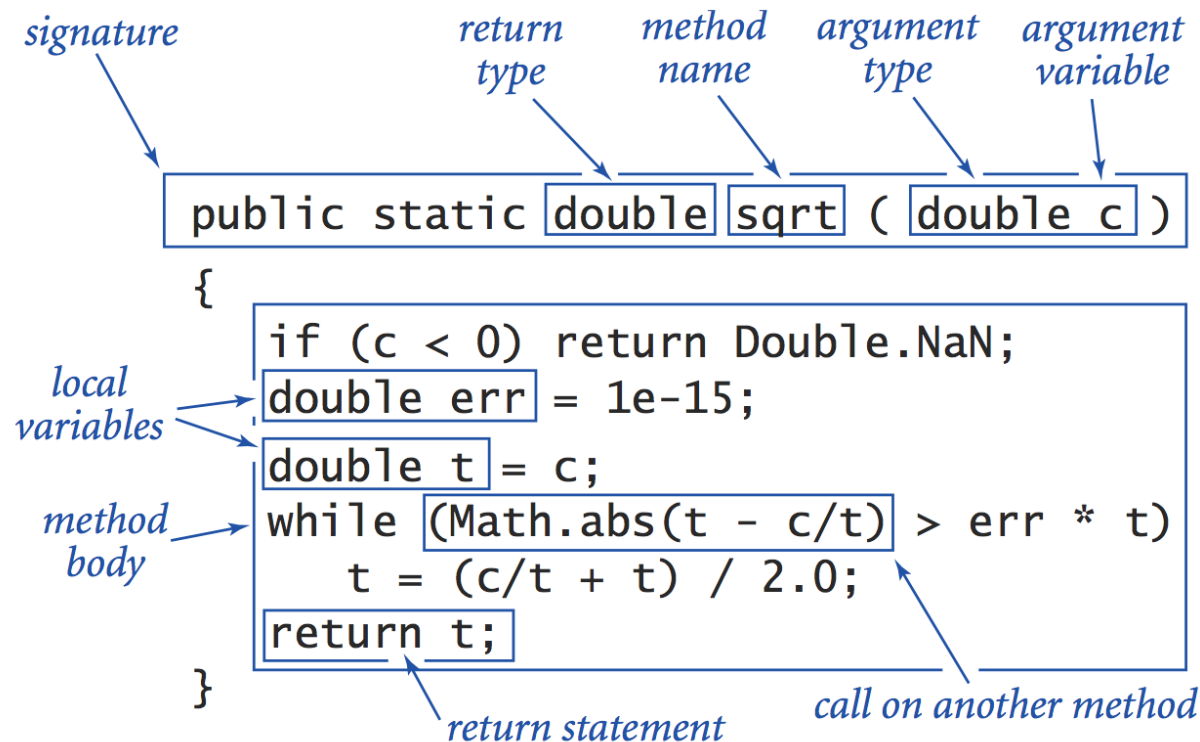
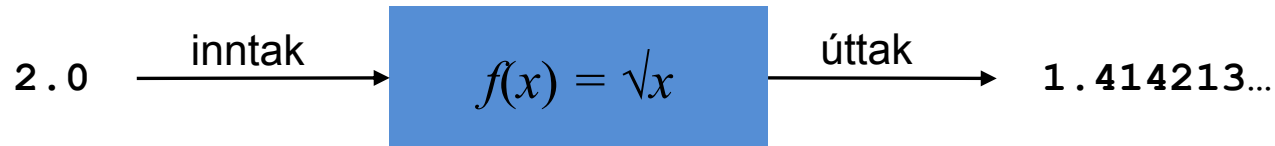
Dæmi um Java föll

- Höfum notað mikið af Java föllum:
 - Innbyggð föll:
 - `Math.random()`, `System.out.println()`,
`Integer.parseInt()`, ...
 - Inntaks/úttaks forritasöfnin:
 - `StdIn.readInt()`, `StdDraw.line()`, `StdAudio.play()`, ...
 - Við höfum meira að segja skilgreint eitt fall:
 - `main()`
- Nú ætlum við að byrja að skrifa fleiri föll sjálf!





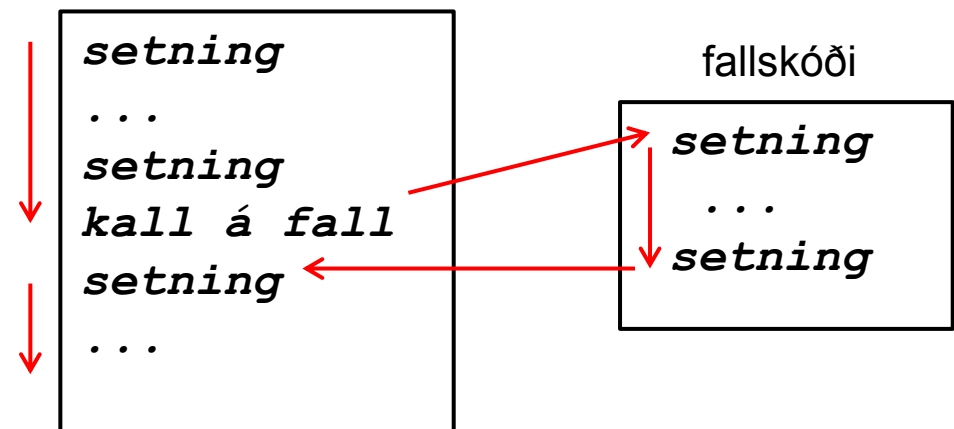
Skipulag Java falla





Stýring framkvæmdar (*flow of control*)

- Höfum séð aðferðir til að breyta framkvæmdaröð skipana:
 - Stýriskipanir: `if`-setning, `switch`-setning
 - Lykkjur: `for`-setning, `while`-setning
- Föll gefa nýja leið til að breyta framkvæmdaröð skipana
 - Köllum á föllin (*function call*)
 - Þau eru framkvæmd
 - Föllum síðan aftur í forritið sem kallaði





Dæmi um Java fall

([Java Visualizer](#))

Athugið:

Keyrsla hefst alltaf í `main`-falli klasa

Fall sem reiknar kvaðratrót með aðferð Newton-Raphson

Aðalfallið fær inn kommutölur á skipanalínu

Breyta gildum í `args`-fylkinu yfir í `double` fylki

Kalla á `sqrt`-fallið fyrir hvert stak og skrifa út gildið

```
public class Newton
{
    public static double sqrt(double c)
    {
        if (c < 0) return Double.NaN;
        double err = 1e-15;
        double t = c;
        while (Math.abs(t - c/t) > err * t)
            t = (c/t + t) / 2.0;
        return t;
    }

    public static void main(String[] args)
    {
        int N = args.length;
        double[] a = new double[N];
        for (int i = 0; i < N; i++)
            a[i] = Double.parseDouble(args[i]);
        for (int i = 0; i < N; i++)
        {
            double x = sqrt(a[i]);
            StdOut.println(x);
        }
    }
}
```



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Hvað gerist?

- Nokkrir mikilvægir hlutir að gerast í kalli á fall:
 - Keyrsla forritsins færist efst í fallið
 - Viðfangsbreyturnar fá gildi úr kallinu
 - Forritskafli fallsins er keyrður
 - Skilagildi fallsins er skilað þangað sem kallað var
 - Keyrsla heldur áfram þar sem frá var horfið

Þetta er kall með gildi (*call by value*)

```
...  
double x = sqrt(a[i]);  
...
```

```
public static double sqrt(double c) {  
    ...  
}
```



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Einfalt dæmi

([Java Visualizer](#))

```
public class MinFall {  
  
    public static int min(int a, int b) {  
        if ( a < b )  
            return a;  
        else  
            return b;  
    }  
  
    public static void main(String[] args) {  
        int x = Integer.parseInt(args[0]);  
        int y = Integer.parseInt(args[1]);  
  
        int m = min( x, y );  
        StdOut.printf("Lággildið er %d\n", m);  
    }  
}
```

Skilgreinum okkar
eigið lággildisfall

Staðsetning fallsins
í klasanum skiptir
ekki máli

Galli:
Fallið er aðeins
skilgreint fyrir `int`



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Fyrirlestraræfing

1. Í stærðfræði gefur $f(4)$ alltaf sama gildið. Gildir það sama um Java föll?
2. Skrifið Java fallið `mult2(double x)`, sem skilar `2.0*x`
3. Skrifið Java fallið `oddatala(int i)`, sem skilar `true` ef `i` er oddatala, en `false` annars



Gildissvið breytu (*scope*)

Sá hluti forritsins þar sem hægt er að vinna með breytuna

Viðfangsbreyta falls er staðvær (*local*) innan þess

Aðeins hægt að vinna með breytur falls innan þess

Gildir líka um lykkjubreytu í `for`-lykkju. Gildissvið hennar er aðeins lykkjan sjálf

*scope of
c, err, and t*

*scope of
args[], N, and a[]*

```
public class Newton
{
    public static double sqrt(double c)
    {
        if (c < 0) return Double.NaN;
        double err = 1e-15;
        double t = c;
        while (Math.abs(t - c/t) > err * t)
            t = (c/t + t) / 2.0;
        return t;
    }
}
```

*this code cannot refer to
args[], N, or a[]*

```
public static void main(String[] args)
{
    int N = args.length;
    double[] a = new double[N];
    for (int i = 0; i < N; i++)
        a[i] = Double.parseDouble(args[i]);

    for (int i = 0; i < N; i++)
    {
        double x = sqrt(a[i]);
        StdOut.println(x);
    }
}
```

*this code cannot refer to
c[], err, or t*

scope of i

scope of i and x

*two different
variables*



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Dæmi um gildissvið

- Hvað gerist þegar þetta forrit er þýtt og keyrt?

Staðvæð breyta j

```
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));  
    }  
}
```

([Java Visualizer](#))

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



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Dæmi um gildissvið

- Hvað gerist þegar þetta forrit er þýtt og keyrt?

i er þegar skilgreind sem viðfangsbreyta

```
public class Cubes2 {  
  
    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));  
    }  
}
```

Error: Duplicate local variable i



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Dæmi um gildissvið

- Hvað gerist þegar þetta forrit er þýtt og keyrt?

Ekki nóg að setja
bara skilagildið í
breytuna `i`

```
public class Cubes3 {  
  
    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));  
    }  
}
```

Error: This method must return a result of type int





Dæmi um gildissvið

- Hvað gerist þegar þetta forrit er þýtt og keyrt?

i-ið í fallinu cube
og i-ið í main eru
ólíkar breytur

```
public class Cubes4 {  
  
    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));  
    }  
}
```

```
% javac Cubes4.java  
% java Cubes4 6  
1 1  
2 8  
3 27  
4 64  
5 125  
6 216
```



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Dæmi um gildissvið

- Hvað gerist þegar þetta forrit er þýtt og keyrt?

Óþarfi að setja skilagildið í aðra breytu

Ef útreikningur í falli er einfaldur þá setjum við hann oft beint í `return`-setninguna

```
public class Cubes5 {  
  
    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));  
    }  
}
```

```
% javac Cubes5.java  
% java Cubes5 6  
1 1  
2 8  
3 27  
4 64  
5 125  
6 216
```



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Yfirhleðsla (*overloading*)

- Hvert fall hefur mynsturfar (*signature*):
 - Nafn fallsins ásamt fjölda og tagi viðfanga

Þetta fall ræður aðeins við `int`-viðföng

- Dæmi:

```
public static int min(int x, int y) {  
    if ( a < b ) return a;  
    else      return b;  
}
```

Þetta er annað fall,
því mynsturförin
eru ólík

```
public static double min(double x, double y) {  
    if ( a < b ) return a;  
    else      return b;  
}
```

Kallað **yfirhleðsla** að nota sama nafn
fyrir tvö föll með ólík mynsturför

Algengt í Java forritun



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Eitt skilagildi

- Java föll geta haft mörg viðföng (*parameters*)
- Hvert Java fall getur aðeins skilað einu gildi
 - Getur reyndar skilað fylki
 - Sjáum síðar að það er hægt að skilgreina gagnatög sem geta innihaldið mikið af upplýsingum
 - Föll með skilalagið `void` skila reyndar engu gildi
 - Þau eru notuð vegna hliðaráhrifa (útprintun, teikning, ...)

```
public static void prenta(int x) {  
    System.out.println(x);  
}
```



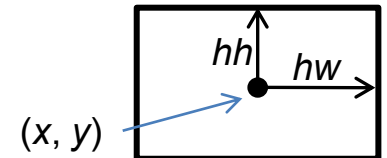


Dæmi um Java föll

- Skilar slembiheiltölum á bilinu $[0, N-1]$:

```
public static int uniform(int N) {  
    return (int) (Math.random()*N);  
}
```

- Teiknar rétthyrning (*rectangle*):
 - Miðja í (x, y) , hálf breidd hw og hálf hæð hh



```
public static void rectangle(double x, double y,  
                             double hw, double hh) {  
    rx = {x-hw, x+hw, x+hw, x-hw};  
    ry = {y+hh, y+hh, y-hh, y-hh};  
    StdDraw.polygon(rx, ry);  
}
```



Margar skilaskipanir

- Föll geta haft fleiri en eina skilaskipun (**return**)
 - Þegar keyrsla lendir í skilaskipun þá hættir fallið keyrslu og keyrsla heldur áfram hjá þeim sem kallaði
 - Allar mögulegar keyrslu þurfa að enda í skilaskipun
 - Ef fallið á að skila gildi

```
public static boolean vandamal(int n) {  
    if (n > 0)  
        return true;  
    else if (n <= 0)  
        return false;  
}
```

Error: This method must return a result of type boolean



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD

Það er ein möguleg leið í gegnum forritið sem hefur ekki **return**-setningu



Fleiri dæmi

- Ákvarða hvort viðfangið N sé frumtala (*prime*)

```
public static boolean isPrime(int N)
{
    if (N < 2) return false;

    for (int i=2; i<=N/i; i++)
        if (N%i == 0) return false;

    return true;
}
```

Skila **ósatt** ef
minni en 2

Skila **ósatt** ef
einhver tala
gengur uppí N

Lykkjan er keyrð þar til $i > N/i$
eða $i^2 > N$,
sem er jafngilt og $i > \sqrt{N}$

Þegar búið er að prófa
allar tölur uppí \sqrt{N} , þá
skila **satt**





Finna N -tu þýðtölu

```
public static double H(int N)
{
    double sum = 0.0;
    for (int i=1; i<=N; i++)
        sum += 1.0/i;

    return sum;
}
```

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} = \sum_{k=1}^n \frac{1}{k}.$$

Eigum nú fall til að finna N -tu þýðtölu

- Getum nú brotið forritin okkar upp eftir virkni
 - Afmarka tiltekna aðgerðir
 - Endurnýtum kóða með því að búa til nytsöm föll
- Sjáum meira af þessu í næsta tíma!





Fyrirlestraræfing

4. Hvaða gildi prentast út í kóðanum hér til hliðar?

```
public class Leyndarmal {  
    public static int fall(int i) {  
        i++;  
        return i;  
    }  
    public static void main(String[] args) {  
        int i = 4;  
        int j = fall(i);  
        StdOut.println(i + ", " + j);  
    }  
}
```

5. Sýnið Java fallið `slembi(m, n)` sem skilar slembiheiltölum á bilinu `[m, n)`

6. Hvað gerist ef það er forritskóði á eftir `return`-skipun í falli?





Samantekt

- Í þessum tíma:
 - Föll í Java
 - Skilgreining og notkun falla
- Í næsta tíma:
 - Stærri dæmi um föll
 - Notkun fylkja í föllum

Kaflí 2.1

Kaflí 2.1



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD