



Tölvunarfræði 1

Fyrirlestur 7: Lykkjur I

Hjálmtyr Hafsteinsson
Haust 2015



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Í síðasta fyrirlestri

- Val á milli margra möguleika
- Hreiðraðar (*nested*) `if`-setningar
- `switch`-setningin

Kafli 1.3



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Í þessum fyrirlestri

- Helstu gerðir lykkja í Java
- **while**-lykkjan
 - Lykkja á meðan skilyrði satt
- **for**-lykkjan
 - Lykkja í tiltekinn fjölda skipta

Kafli 1.3



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Ítrun (*iteration*)

- Viljum stundum gera sama hlutinn aftur og aftur
 - Gætum þá skrifað sömu skipanirnar mörgum sinnum:

```
System.out.println("Ég ætla ekki að blóta!");  
System.out.println("Ég ætla ekki að blóta!");  
System.out.println("Ég ætla ekki að blóta!");  
System.out.println("Ég ætla ekki að blóta!");  
System.out.println("Ég ætla ekki að blóta!");
```

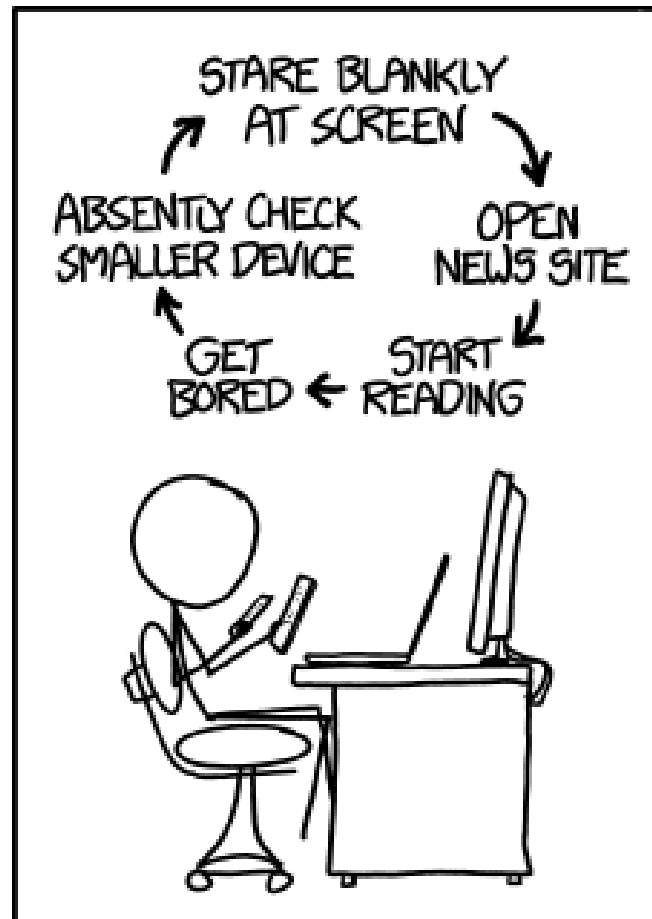
- eða sett skipanirnar inn í lykkju:

```
for (int i=0; i<5; i++)  
    System.out.println("Ég ætla ekki að blóta!");
```





Endalaus lykkja?



xkcd.com



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Tegundir lykkja í Java

- Java hefur þrjár gerðir lykkja
 - Við munum aðallega skoða tvær þeirra:

while-lykkja:

```
while (röksegð) {  
    setning;  
}
```

for-lykkja:

```
for (upphafsstilling; röksegð; upphækkun) {  
    setning;  
}
```





while-lykkjan

- Svipuð uppsetning og á `if`-setningu (án `else`)
 - `while` í stað `if`
- Virkni er ólík:
 - Reikna röksegðina
 - Ef hún er sönn þá framkvæma setningu
 - Endurtaka (þ.e. reikna röksegðina aftur, o.s.frv.)
 - Hættum ekki fyrr en röksegðin er ósönn

```
while (röksegð) {  
    setning;  
}
```

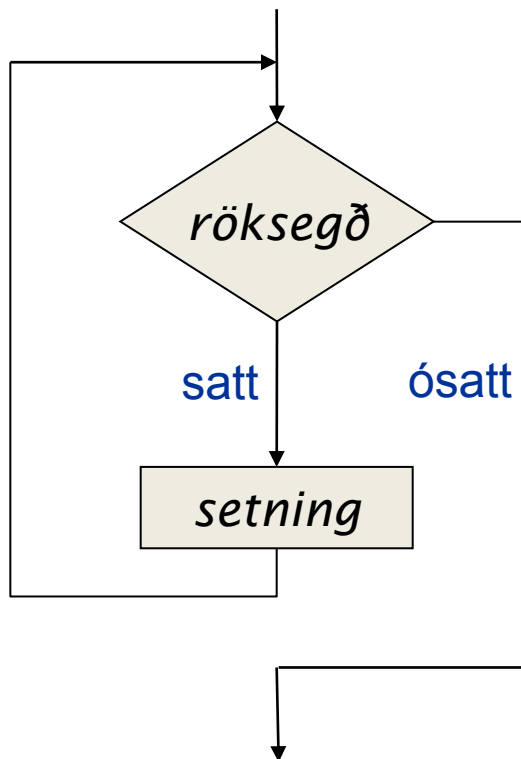
Virknin er svipuð og röð `if`-setninga:

```
if (röksegð) { setning; }  
if (röksegð) { setning; }  
...
```





Flæðirit `while`-lykkju



```
while (röksegð) {  
    setning;  
}
```

Getum sleppt mengjasvigum ef aðeins ein setning í lykkjunni

Ef skilyrðið er ósatt í upphafi þá er lykkjan aldrei framkvæmd

Það er enginn `else`-hluti. Ef skilyrði er ósatt þá hættum við í lykkjunni



Einföld `while`-lykkja

- Skrifa út 10 eintök af "Halló"

([Java Visualizer](#))

```
int i = 1;
while (i <= 10) {
    System.out.println(i + "th Hello");
    i = i + 1;
}
```

Breytan `i` er notuð sem teljari (*counter*) og hækkar í hverri ítrun

Þar sem breytan `i` er notuð í `println`-skipuninni þá er úttakið ekki alltaf það sama

```
1th Hello
2th Hello
3th Hello
4th Hello
5th Hello
6th Hello
7th Hello
8th Hello
9th Hello
10th Hello
```



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Skilyrðið verður að breytast

- Rökyrðingin í `while`-setningunni verður að breytast
 - Annars fáum við endalaus lykkju (eða enga ítrun)

```
int i = 1;
while (i <= 10) {
    System.out.println(i + "th Hello");
    //    i = i + 1;
}
```

Ef breytan `i` er aldrei hækkuð þá er skilyrðið alltaf það sama, þ.e. `1 <= 10`, sem er **satt**



Flottari útfærsla

Ekki öll tilfellið passa inní lykkjuna. Skrifum þau sérstaklega

```
public class TenHellos {  
    public static void main(String[] args) {  
  
        System.out.println("1st Hello");  
        System.out.println("2nd Hello");  
        System.out.println("3rd Hello");  
  
        int i = 4;  
        while (i <= 10) {  
            System.out.println(i + "th Hello");  
            i = i + 1;  
        }  
    }  
}
```



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Prenta út veldi af 2

- Prenta út töflu með i og 2^i
 - Stærð töflunnar, N , kemur sem skipanalínuviðfang

([Java Visualizer](#))

```
public class PowersOfTwo {  
    public static void main(String[] args) {  
  
        int N = Integer.parseInt(args[0]);  
        int i = 0;  
        int powerOfTwo = 1;  
  
        while (i <= N) {  
            System.out.println(i + " " + powerOfTwo);  
            powerOfTwo = 2 * powerOfTwo;  
            i = i + 1;  
        }  
    }  
}
```

Lesið N af skipanalínu

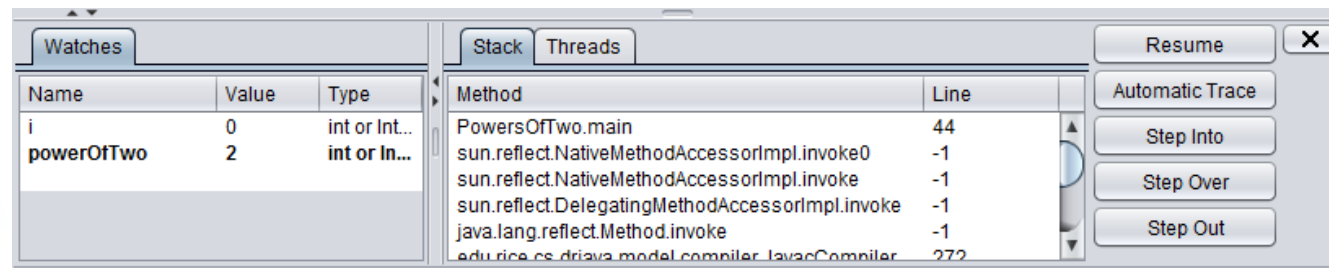
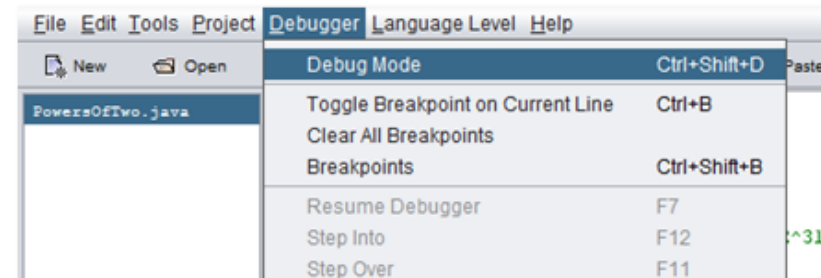
Breytið `powerOfTwo`
inniheldur alltaf 2^i





Rekja keyrslu forrits

- *DrJava* hefur leið til að rekja forrit í keyrslu:
 - Setja rofstað (*breakpoint*) í einhverja línu
 - Hægri-smella í línu og velja "*Toggle Breakpoint*"
 - Velja "*Debug mode*" í "*Debugger*" valmynd
 - Keyra forrit eins og venjulega
 - Fáum þá upp glugga þar sem hægt að fylgjast með breytum og að rekja sig í gegnum forritið



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Prenta út breytur

- Önnur leið til að rekja keyrslu forrits:

Setja inn `println`-skipanir!

Fyrir þá sem vilja ekki nota DrJava

- Virkar í skipanalínukeyrslu
- Neyðir mann til að reyna að hugsa um framkvæmd forritsins
- Þurfum svo að muna eftir að eyða þessum úttaksskipunum í lokin





Fyrirlestraræfing

1. Hvað myndi gerast í lykkjunni sem á að prenta út 10 "Halló" ef við settum `"i = i - 1"` í stað `"i = i + 1"`?
2. Hvert er vandamálið við eftirfarandi lykkju?

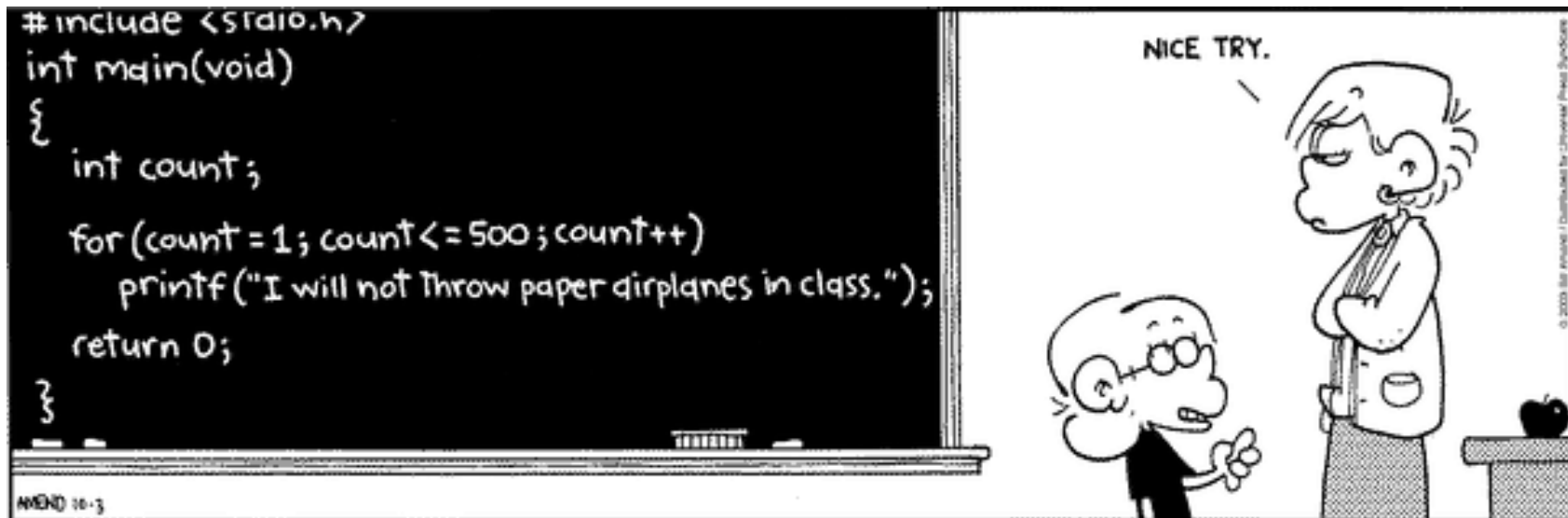
```
int i = 1;
while (i <= 10)
    System.out.println(i + "th Hello");
    i = i + 1;
```

3. Skrifið lykkju sem prentar út *sin*-fallið á bilinu 0.0 til 1.0 með upphækkun 0.1?





for-lykkja



Copyright 2004, FoxTrot by Bill Amend
www.ucomics.com/foxtrot/2003/10/03

Stundum er gott að kunna að forrita



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



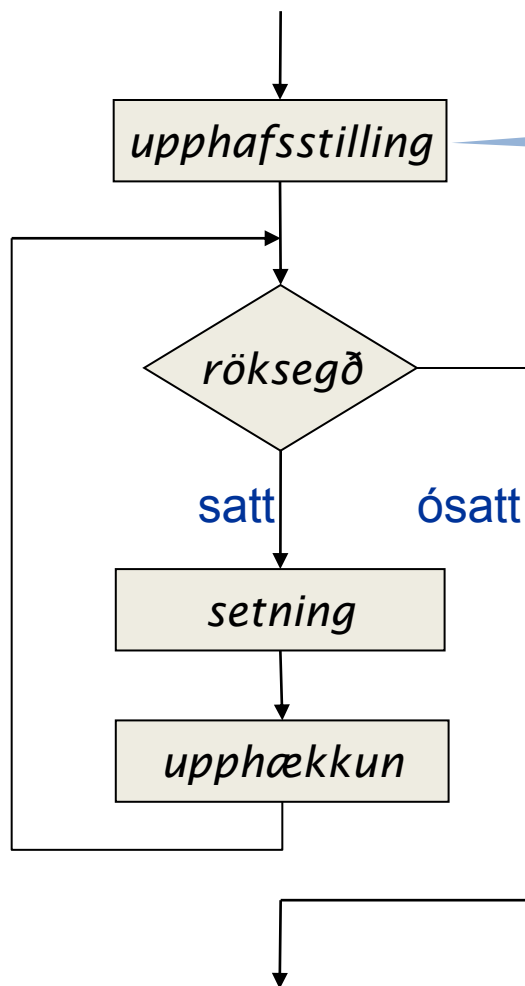
for-lykkjan

- Oftast notuð fyrir fastan fjölda ítrana
- Virkni:
 - Framkvæma *upphafsstillingu*
 - Ef *röksegð* er sönn þá framkvæma *setningu*
 - Framkvæma *upphækkun*
 - Fara aftur í að reikna *röksegð*

```
for (upphafsstilling; röksegð; upphækkun) {  
    setning;  
}
```



Flæðirit for-lykkju



Upphafsstilling er framkvæmd nákvæmlega einu sinni

Upphækkun er notuð til þess að skilyrðið verði einhvern tíman ósatt

```
for (upphafsstilling; röksegð; upphækkun) {  
    setning;  
}
```

Jafngild
while-lykkja

```
upphafsstilling;  
while (röksegð) {  
    setning;  
    upphækkun;  
}
```

(Ekki fleiri flæðirit!)

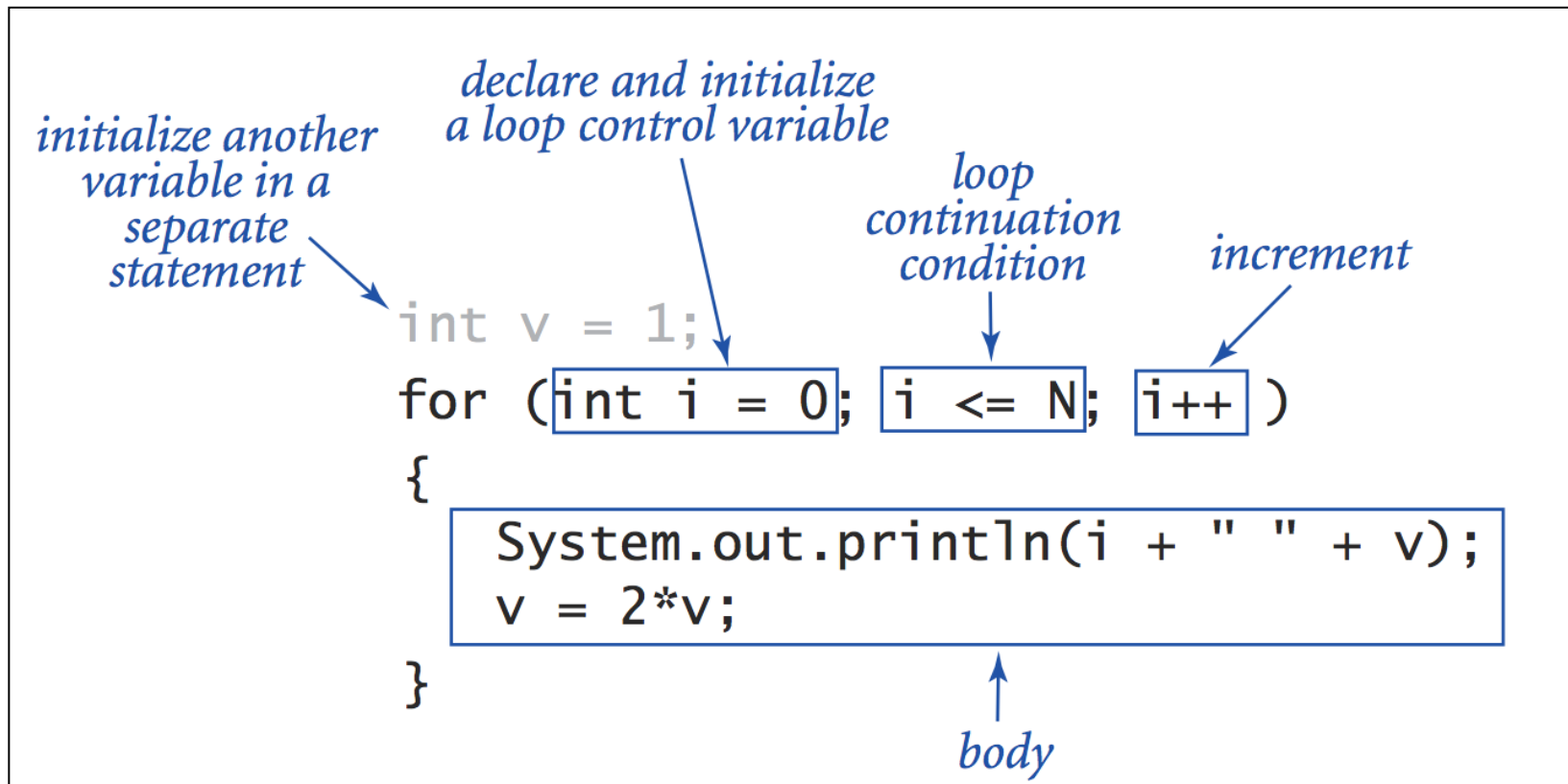


HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Dæmi um for-lykkju





Upphækkun breyta

- Java notar oft styttingu til að hækka og lækka gildi breyta:

Stytting

breyta++;

breyta--;

Lengri útgáfa

breyta = breyta + 1;

breyta = breyta - 1;

```
int i = 5;  
i++;
```

Nú hefur *i* gildið 6

```
double x = 4.2;  
x--;
```

Nú hefur *x* gildið 3.2



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Upphækkun breyta

- Við getum sett upphækkunina á undan breytunni eða á eftir
 - Breytir því hvaða gildi er notað ef segðin er í útreikningi

```
int i = 2;  
int k = i++;
```

Breytan `k` fær gildið 2, þ.e. gamla gildi `i`

```
int i = 2;  
int k = ++i;
```

Breytan `k` fær gildið 3, því `i` er fyrst hækkað

```
...  
i++;  
...
```

jafngilt

```
...  
++i;  
...
```



Upphækkun um almennt gildi

- Líka til stytting ef uppfæra á gildi breytu um almennt gildi

Stytting

```
breyta += gildi;  
breyta -= gildi;  
breyta *= gildi;  
breyta /= gildi;  
breyta %= gildi;
```

```
int i = 5;  
i += 2;
```

Nú hefur *i* gildið 7

Löng útgáfa

```
breyta = breyta + gildi;  
breyta = breyta - gildi;  
breyta = breyta * gildi;  
breyta = breyta / gildi;  
breyta = breyta % gildi;
```

```
double x = 4.2;  
x -= 0.5;
```

Nú hefur *x* gildið 3.7



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Dæmi um for-lykkjur

- Reikna summuna $1+2+\dots+N$:

([Java Visualizer](#))

```
int sum = 0;
for (int i = 1; i <= N; i++)
    sum += i;
System.out.println(sum);
```

- Reikna summuna $1^2+2^2+\dots+N^2$:

```
int sum = 0;
for (int i = 1; i <= N; i++)
    sum += i*i;
System.out.println(sum);
```

Breytum i í $i*i$



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Dæmi um for-lykkjur

- Reikna töflu af fallsgildum á bilinu 0 til 2π
 - Viðfangið N ákveður fjölda uppskiptinga: $2\pi \frac{i}{N}$

([Java Visualizer](#))

```
for (int i = 0; i <= N; i++)  
    System.out.println(i + ": " + 2*Math.PI*i/N);
```

Gætum notað þessi gildi
sem inntak í sínus- eða
kósínusfallið

```
% java FallsGildi 5  
0: 0.0  
1: 1.2566370614359172  
2: 2.5132741228718345  
3: 3.7699111843077517  
4: 5.026548245743669  
5: 6.283185307179586
```



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Dæmi um for-lykkjur

- Skilyrði og upphækkun geta verið almennari:

```
for (int v = 1; v <= N; v *= 2)
    System.out.println(v);
```

- Líka hægt að hafa fleiri en eina skipun:

([Java Visualizer](#))

```
int i, v;
for (i = 0, v = 1; i <= N; v *= 2, i++)
    System.out.println(i + ": " + v);
```

Verðum þá að skilgreina
breyturnar fyrir utan for-lykkuna



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Fyrirlestraræfing

4. Skrifið `for`-lykkju sem reiknar $1*2*3*...*N$, þ.e. $N!$
5. Ef upphafsgildi `a` er 5 og `b` er 2, hvert er gildi þeirra í lok eftirfarandi kóða?
6. Hvert er gildi `j` eftir að eftirfarandi skipun hefur verið framkvæmd?

```
a = b++ - 1;  
b -= ++a;
```

```
for (i = 0, j = 0; i < 10; i++)  
    j += i;
```



Samantekt

- Í þessum tíma:
 - `while`-lykkjur
 - `for`-lykkjur
- Í næsta tíma:
 - Dæmi um notkun lykkja
 - Hreiðraðar lykkjur

Kafli 1.3

Kafli 1.3



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD