



Tölvunarfræði 1

Fyrirlestur 2: Keyrsluumhverfi Java

Hjálmtyr Hafsteinsson
Haust 2015





Í síðasta fyrirlestri

- Kynning á námskeiðinu
 - Fyrirkomulag, námsmat
 - Námssefni
 - Hvernig best að læra
- Java
 - Uppsetning
 - Fyrsta forritið

Kafli 1.1



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Í þessum fyrirlestri

- Keyrsluumhverfi Java
- Fleiri einföld Java forrit
- Skipulag tölva
- Breytur í Java

Kaflar 1.1 - 1.2



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Hvernig keyrast forrit?

- Örgjörvar (*CPU*) hafa innbygggt vélarmál (*machine language*)
 - Mjög frumstæðar skipanir
 - afrita minnishólf, leggja saman innihald hólfa, hoppa á annan stað, ...
 - Nokkrar gerðir vélarmála:
 - Intel 64, IA-32, IBM Power, ARM, ...
- Keyrsluskrá (t.d. **word.exe**) inniheldur runu af vélarmálsskipunum

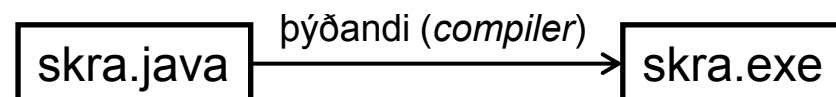




Hvernig keyrast forrit?

- Tvær leiðir til að keyra forrit í æðra forritunarmáli:
- **Þýða** (*compile*) forritið yfir í vélarmálsforrit

- Þurfum þýðanda fyrir hvert stýrikerfi (og vélarmál)



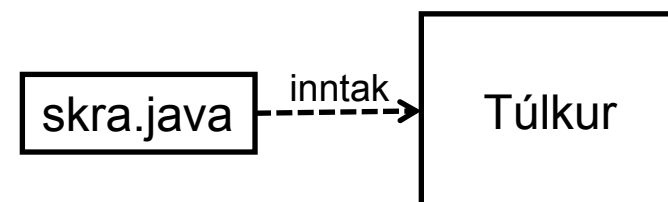
- Hver þýðing tekur tíma

- Keyrslan verður mjög hraðvirk

- **Túlka** (*interpret*) forritið í hvert sinn

- Túlkuning hefst strax (engin töf)

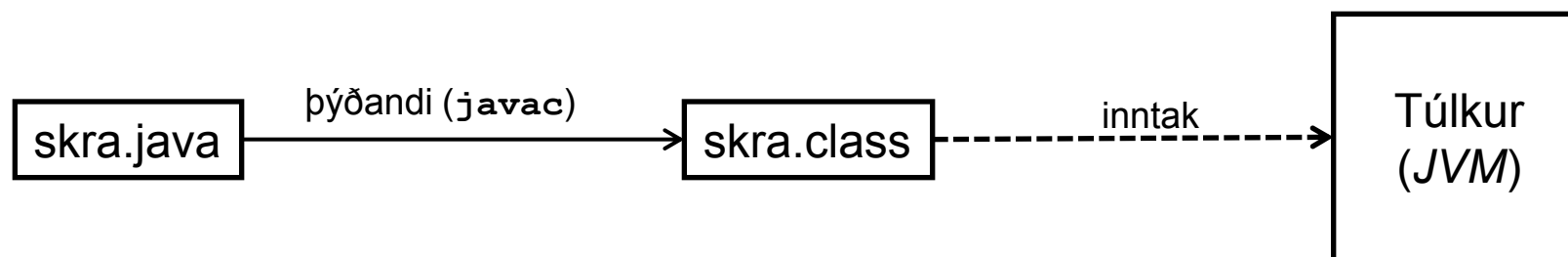
- Keyrsla verður hægsvirkari





Java notar millimál

- Hægt er að nota millileið:
 - Þýða forritið yfir í millimál og túlka það



- Þá þarf aðeins einn þýðanda fyrir öll stýrikerfi
- Millimálið er líkt flestum vélarmálum og auðvelt að túlka það
- Auðvelt að flytja Java á önnur stýrikerfi/tölvur, þarf aðeins frekar einfaldan túlk



Bætakóði Java

- Java skilgreinir Java sýndarvél (*JVM, Java Virtual Machine*)
 - Hefur bætakóða (*bytecode*) sem vélarmál
 - Skráin inniheldur runu af kóðum og er ólæsileg fyrir okkur (flest!)

Java bætakóðaskrár byrja alltaf á HEX tölunum CAFE BABE

Innihald skrárinnar
HelloWorld.class

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f		
00000000	ca	fe	ba	be	00	00	00	33	00	1d	01	00	0a	48	65	6c	3	Hel
00000001	6c	6f	57	6f	72	6c	64	07	00	01	01	00	10	6a	61	76	loWorld	jav
00000002	61	2f	6c	61	6e	67	2f	4f	62	6a	65	63	74	07	00	03	a/lang/Object	
00000003	01	00	0f	48	65	6c	6c	6f	57	6f	72	6c	64	2e	6a	61	HelloWorld.java	
00000004	76	61	01	00	06	3c	69	6e	69	74	3e	01	00	03	28	29	va <init>	()
00000005	56	0c	00	06	00	07	0a	00	04	00	08	01	00	04	6d	61	V	ma
00000006	69	6e	01	00	16	28	5b	4c	6a	61	76	61	2f	6c	61	6e	in	[[Ljava/lang
00000007	67	2f	53	74	72	69	6e	67	3b	29	56	01	00	10	6a	61	g/String;)V	ja
00000008	76	61	2f	6c	61	6e	67	2f	53	79	73	74	65	6d	07	00	va/lang/System	
00000009	0c	01	00	03	6f	75	74	01	00	15	4c	6a	61	76	61	2f	out	Ljava/
0000000a	69	6f	2f	50	72	69	6e	74	53	74	72	65	61	6d	3b	0c	io/PrintStream;	



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Önnur notkun á bætakóða

- Hægt að þýða önnur forritunarmál yfir í Java bætakóða
 - Getum keyrt þau þar sem JVM túlkur er til
 - Til dæmis: Clojure, Scala, Jython, ...
- Það eru til JVM túlkar sem þýða bætakóðann smátt og smátt yfir í vélarmálskóða
 - Kallast JIT (*Just-In-Time*) þýðendur
 - Byrja á því að túlka, en þýða um leið yfir í vélarmál og nota vélarmálsútgáfuna næst





Java beinagrind

- Í þessu námskeiði byrjum við alltaf með ákveðna forritsbeinagrind
 - Lærum svo merkingu einstakra hluta hennar síðar

Beinagrind.java

```
public class Beinagrind {  
  
    public static void main(String[] args) {  
  
        // Hér koma einhverjar skipanir  
  
    }  
}
```





Java dæmi

- Skipunin `print` fer ekki í nýja línu, en `println` færir í nýja línu á eftir strengnum

```
public class HalloHeimur {  
    public static void main(String[] args) {  
        System.out.println();  
        System.out.print("Halló heimur");  
        System.out.println();  
    }  
}
```

Annað stafamengi í skipanaglugga,
séríslenskir stafir virka ekki



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD

```
C:\Tfr1\Forrit> javac HalloHeimur.java
```

```
C:\Tfr1\Forrit> java HalloHeimur
```

```
Hall¾ heimur
```



Inntak í skipanalínu

- Hægt að senda gögn inn í forritið í skipanalínunni
 - Notum þá viðfangið (*argument*) sem er í `main` aðferðinni

```
public class UseArgument {  
    public static void main(String[] args) {  
        System.out.print("Hi, ");  
        System.out.print(args[0]);  
        System.out.println(". How are you?");  
    }  
}
```

```
C:\Tfr1\Forrit> javac UseArgument.java  
C:\Tfr1\Forrit> java UseArgument Siggi  
Hi, Siggi. How are you?  
C:\Tfr1\Forrit> java UseArgument 123  
Hi, 123. How are you?
```

Muna eftir
viðfangi!



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Inntak í DrJava

- Ekki hægt að smella á **Run** í DrJava þegar inntak er í skipanalínu

```
File Edit Tools Project Debugger Language Level Help
New Open Save Close Cut Copy Paste Undo Redo Find Compile Reset Run

UseArgument.java
15
16 public class UseArgument {
17
18     public static void main(String[] args) {
19         System.out.print("Hi, ");
20         System.out.print(args[0]);
21         System.out.println(". How are you?");
22     }
23

Interactions Console Compiler Output
> run UseArgument
Hi, java.lang.ArrayIndexOutOfBoundsException: 0
    at UseArgument.main(UseArgument.java:20)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)
    at java.lang.reflect.Method.invoke(Unknown Source)
    at edu.rice.cs.drjava.model.compiler.JavacCompiler.runCommand(JavacCompiler.java:272)
> java UseArgument 123
Hi, 123. How are you?
Running main Method of Current Document 1:0
```



Villur í Java forritum

- Þið munuð gera villur í Java forritun!!
 - Mikilvægt að kunna að bregðast við

Skrifuðum w í stað v

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

Þýðandinn reynir að átta sig á því hver villan er

```
C:\Tfr1\Forrit> javac HelloWorld.java  
HelloWorld.java:2: error: cannot find symbol  
    public static void main(String[] args) {  
                   ^  
symbol:   class void  
location: class HelloWorld  
1 error
```



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Fyrirlestraræfing

1. Java forrit væru hraðvirkari ef þau væru þýdd beint yfir í vélarmál tölvunnar. Hvers vegna er það ekki gert?
2. Hver er munurinn á athugasemdum skilgreindum með `//` og með `/* ... */`?
3. Hvað gerist ef við sleppum orðinu `class` í fyrstu línu `HelloWorld.java`?





Skipulag tölva

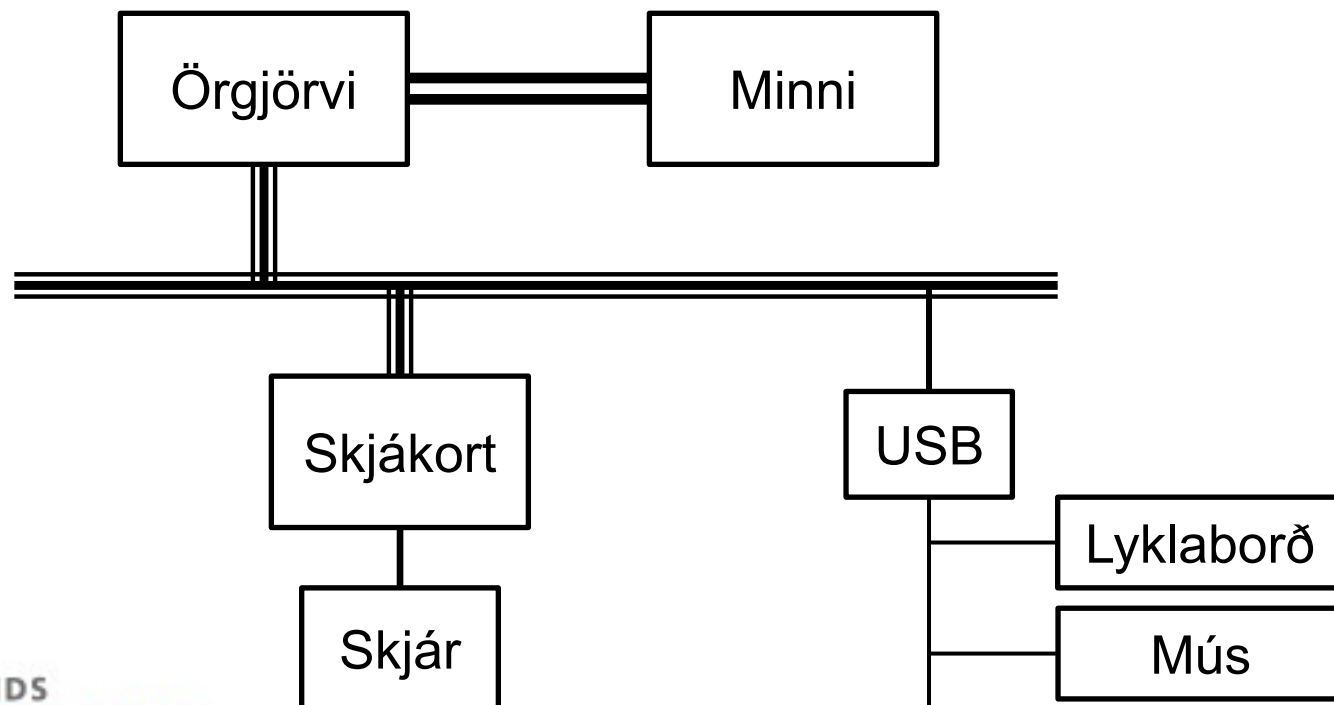
- Tölvur samsettar úr mörgum einingum:
 - Örgjörvi (*CPU*)
 - Framkvæmir útreikninga, stýrir framkvæmd
 - Minni (*memory*)
 - Innra minni geymir gögn og forrit meðan kveikt á tölvu
 - Ytra minni (t.d. harðir diskar) geymir í lengri tíma
 - Inntaks/úttaks-tæki
 - Skjár birtir úttak, hátalarar "birta" hljóð
 - Lyklaborð, mús, hljóðnemi





Brautir (*buses*) í tölvum

- Einingarnar eru tengdar með brautum
 - Hver tölva hefur nokkrar gerðir brauta



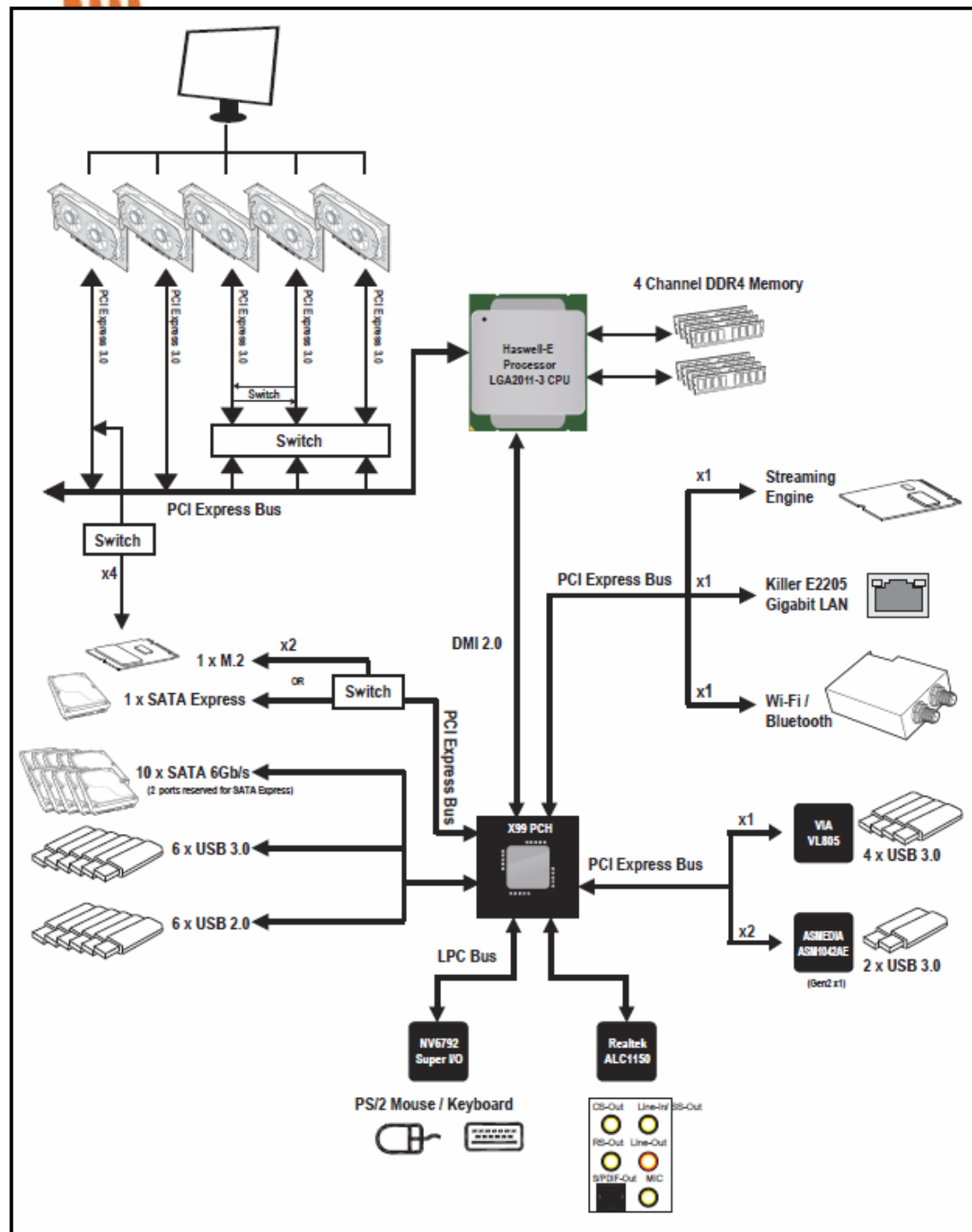
Dæmigert móðurborð

Aðalbrautin er PCI Express

SATA braut fyrir diska

Margar USB brautir

Minnisbraut milli CPU og minnis



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Breytur í Java

- Breyta (*variable*) er minnishólf sem geymir gildi
 - Til nokkrar gerðir af breytum:
 - heiltölu-, kommutölu-, bókstafa-, rökbreyta
- Í Java þarf fyrst að skilgreina (*declare*) breytur
 - Segjum þá **tag** og **nafn** breytunnar:

```
int a, b;  
double x;
```

- Síðan hægt að gefa þeim gildi og nota:

```
a = 5;  
b = a + 2;
```





Breytunöfn í Java

- Breytunafn þarf að byrja á bókstaf, en síðan mega koma bókstafir og tölustafir
- Oft ákveðnar venjur um breytunöfn (eins og í bókinni)
- Ákveðin orð eru þó frátekin (*reserved*) í Java:

<code>abstract</code>	<code>default</code>	<code>goto</code>	<code>package</code>	<code>this</code>
<code>assert</code>	<code>do</code>	<code>if</code>	<code>private</code>	<code>throw</code>
<code>boolean</code>	<code>double</code>	<code>implement</code>	<code>protected</code>	<code>throws</code>
<code>break</code>	<code>else</code>	<code>import</code>	<code>public</code>	<code>transient</code>
<code>byte</code>	<code>enum</code>	<code>instanceof</code>	<code>return</code>	<code>true</code>
<code>case</code>	<code>extends</code>	<code>int</code>	<code>short</code>	<code>try</code>
<code>catch</code>	<code>false</code>	<code>interface</code>	<code>static</code>	<code>void</code>
<code>char</code>	<code>final</code>	<code>long</code>	<code>strictfp</code>	<code>volatile</code>
<code>class</code>	<code>finally</code>	<code>native</code>	<code>super</code>	<code>while</code>
<code>const</code>	<code>float</code>	<code>new</code>	<code>switch</code>	
<code>continue</code>	<code>for</code>	<code>null</code>	<code>synchronized</code>	





Helstu gagnatög

- Java hefur átta grunngagnatög (*primitive data types*)
 - `boolean`, `byte`, `char`, `double`, `float`, `int`, `long`, `short`
 - Fjórar gerðir heiltölutaga: `byte`, `int`, `long`, `short`
 - Tvö kommutölutög: `double`, `float`
 - Bókstafatag: `char`
 - Röktag: `boolean`
- Sjáum síðar fleiri gagntög
 - Við munum skilgreina okkar eigin gagnatög (þ.e. klasa)

Notum `int`, `double` og `boolean` langmest





Tagskipting

- Java er rammlega tagskipt (*strongly typed*)
 - Öll gildi hafa tög
 - Allar breytur hafa tög
 - Breyta getur ekki skipt um tag
- Tagskipting hjálpar við að skrifa rétt forrit
 - Viðheldur aga í forritun
 - Villur uppgötvast fyrr
 - Árið 1996 sprakk Ariane 5 vegna slæmra breytinga milli gagnataga





Gagnatög

- Gagnatag er mengi af gildum og aðgerðir á þau

tag	mengi gilda	einstök gildi	aðgerðir
char	bókstafir	'A' '@'	samanburður
String	runa af stöfum	"Hello World" "126 is fun"	samskeyting
int	heiltölur	17 12345	samlagning, margföldun, ...
double	kommutölur	3.1415 6.022e23	samlagning, margföldun, ...
boolean	sanngildi	true false	and, or, not

String er
klasi, sem
er inn-
byggður í
Java





Hvað gerist í breytum?

- Breytur eru minnishólf sem hafa tiltekinn líftíma

	a	b	t
int a, b;	óskilgr	óskilgr	
a = 1234;	1234	óskilgr	
b = 99;	1234	99	
int t = a;	1234	99	1234
a = b;	99	99	1234
b = t;	99	1234	1234

Víxla á
gildum a
og b

t er ekki
til hér



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Java Visualizer

- Veftól sem gefur ýmsar upplýsingar um Java forrit
 - Límið (*paste*) forrit inn í gluggann og smellið á Visualize-hnapp
 - Hægt keyra forritið skref fyrir skref, líka aftur á bak!
 - Sýnir innihald breyta

Gott fyrir smærri forrit og gott til að skilja hegðun forrita

Java Visualizer
(beta: [report a bug](#))

Frames Objects

main:6	
a	5
b	7

```
1 public class HalloHeimur {  
2     public static void main(String[] args) {  
3         int a, b;  
4         a = 5;  
5         b = a + 2;  
6         System.out.println(b);  
7     }  
8 }
```

[Edit code](#)

Step 4 of 5

line that has just executed next line to execute

Program output:



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD

slóð: http://cscircles.cemc.uwaterloo.ca/java_visualize/



Fyrirlestraræfing

4. Hvers vegna eru ekki allar einingar tölvunnar bara tengdar við eina mjög hraða braut?
5. Væri hægt að ákveða tag breyta út frá tagi gildanna sem eru settar í þær (og losna þá við að skilgreina breytur)?
6. Hvaða gildi hefur `b` eftir skipanirnar að neðan

```
int a;  
int b = a;
```





Samantekt

- Í þessum tíma:
 - Keyrsluumhverfi fyrir Java **Kaflar 1.1 - 1.2**
 - Skipulag tölva
 - Breytur í Java
- Í næsta tíma:
 - Helstu gagnatög í Java **Kafla 1.2**
 - Segðir
 - Samanburður

