

Fylki

Fylki eru notuð til að halda utan um mikið magn gagna

Fylki er runa gagna af sama tagi

- 52 Spil í spilastokk
- 230 nemendur í námskeiðinu
- 1 milljón stafir í bók
- 10 milljón hljóðbylgjur í mp3 skrá
- 3 milljarðar basapara í DNA
- 73 milljarðar fyrirspurna hjá Google á ári

Margar breytur

Með 10 nemendum í námskeiði, hvernig höldum við utan um einkunnir?

```
// slæmt, auðvelt að gera villur
double a0, a1, a2, a3, a4, a5, a6, a7, a8, a9;
a0 = 0.0;
a1 = 0.0;
a2 = 0.0;
a3 = 0.0;
a4 = 0.0;
a5 = 0.0;
a6 = 0.0;
a7 = 0.0;
a8 = 0.0;
a9 = 0.0;
...
a4 = 3.0;
...
a8 = 8.0;
...
double x = a4 + a8;
```

Með 10 nemendur í námskeiði, hvernig höldum við utan um einkunnir?

```
// einfaldara, nær því sem við viljum gera
double[] a = new double[10];
...
a[4] = 3.0;
...
a[8] = 8.0;
...
double x = a[4] + a[8];
```

Hvað með 1 milljón breyta?

```
// einfalt að búa til stór fylki  
double[] a = new double[1000000];  
...  
a[123456] = 3.0;  
...  
a[987654] = 8.0;  
...  
double x = a[123456] + a[987654];
```

Fylki eru innbyggð í Java

- Til að búa til fylki: skilgreina, skapa og upphafsstill
- Til að nota gildi númer i í fylki a , notum við $a[i]$
- Við byrjum að telja í 0

```
int N = 10;           // size of array
double[] a;          // declare the array
a = new double[N];   // create the array
for (int i = 0; i < N; i++) // initialize the array
    a[i] = 0.0;      // all to 0.0
```

Fylki eru innbyggð í Java

- Til að búa til fylki: skilgreina, skapa og upphafsstillta
- Til að nota gildi númer i í fylki a , notum við $a[i]$
- Við byrjum að telja í 0

```
int N = 10;           // size of array
double[] a;          // declare the array
a = new double[N];   // create the array
for (int i = 0; i < N; i++) // initialize the array
    a[i] = 0.0;      // all to 0.0
```

Einfaldari lausn

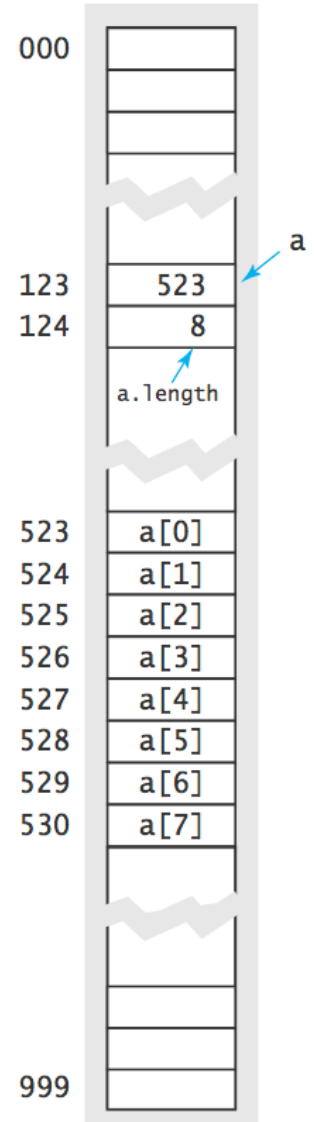
- Skilgreina og skapa fylki í sömu setningu
- Java núllstillir fylkin fyrir okkur

```
int N = 10;           // size of array
double[] a = new double[N]; // declare, create, init
```

Fylki í minni

Fylki eru geymd í minni tölvu í
“rétttri” röð

- Öll fylki hafa fasta stærð
- Fljótlegt að finna gildi í fylkjum
- Enn fljótleggra að finna gildi nálægt hvoru öðru
- Tvívíð fylki: fylki af fylkjum, geta dreifst um minnið



Innfeldi

Reiknum innfeldi tveggja fylkja $x[]$ og $y[]$ af lengd N með því að leggja saman margfeldi stakanna

```
double[] x = { 0.3, 0.6, 0.1 };
double[] y = { 0.5, 0.1, 0.4 };
int N = x.length;
double sum = 0.0;
for (int i = 0; i < N; i++) {
    sum = sum + x[i]*y[i];
}
```

<u>i</u>	<u>x[i]</u>	<u>y[i]</u>	<u>x[i]*y[i]</u>	<u>sum</u>
				0
0	.30	.50	.15	.15
1	.60	.10	.06	.21
2	.10	.40	.04	.25
				.25

<i>create an array with random values</i>	<pre>double[] a = new double[N]; for (int i = 0; i < N; i++) a[i] = Math.random();</pre>
<i>print the array values, one per line</i>	<pre>for (int i = 0; i < N; i++) System.out.println(a[i]);</pre>
<i>find the maximum of the array values</i>	<pre>double max = Double.NEGATIVE_INFINITY; for (int i = 0; i < N; i++) if (a[i] > max) max = a[i];</pre>
<i>compute the average of the array values</i>	<pre>double sum = 0.0; for (int i = 0; i < N; i++) sum += a[i]; double average = sum / N;</pre>
<i>copy to another array</i>	<pre>double[] b = new double[N]; for (int i = 0; i < N; i++) b[i] = a[i];</pre>
<i>reverse the elements within an array</i>	<pre>for (int i = 0; i < N/2; i++) { double temp = b[i]; b[i] = b[N-1-i]; b[N-i-1] = temp; }</pre>

Gildisveiting

Getum sett gildin í fylki þegar við þýðum forrit

```
String[] rank = {
    "2", "3", "4", "5", "6", "7", "8", "9",
    "10", "Jack", "Queen", "King", "Ace"
};

String[] suit = {
    "Clubs", "Diamonds", "Hearts", "Spades"
};

int i = (int) (Math.random() * 13); // between 0 and 12
int j = (int) (Math.random() * 4); // between 0 and 3

System.out.println(rank[i] + " of " + suit[j]);
```

Gildisveiting

Setjum oftast gildi í fylki við keyrslu.

```
String[] deck = new String[52];  
for (int i = 0; i < 13; i++)  
    for (int j = 0; j < 4; j++)  
        deck[4*i + j] = rank[i] + " of " + suit[j];  
  
for (int i = 0; i < 52; i++)  
    System.out.println(deck[i]);
```

Hvað prentast?

A

```
two of clubs  
two of diamonds  
two of hearts  
two of spades  
three of clubs  
...
```

B

```
two of clubs  
three of clubs  
four of clubs  
five of clubs  
six of clubs  
...
```

Viljum stokka spilín í slembiröð

Reiknirit:

- Í i -tu ítrun veljum spil frá $\text{deck}[i]$ til $\text{deck}[N-1]$ af handahófi (öll jafn líkleg)
- Skiptum því út fyrir $\text{deck}[i]$

```
int N = deck.length;
for (int i = 0; i < N; i++) {
    int r = i + (int) (Math.random() * (N-i));
    String t = deck[r];
    deck[r] = deck[i];
    deck[i] = t;
}
```

Stokkun

```
public class Deck {  
    public static void main(String[] args) {  
        String[] suit = { "Clubs", "Diamonds", "Hearts", "Spades" };  
        String[] rank = { "2", "3", "4", "5", "6", "7", "8", "9",  
                          "10", "Jack", "Queen", "King", "Ace" };  
  
        int SUITS = suit.length;  
        int RANKS = rank.length;  
        int N = SUITS * RANKS;  
  
        String[] deck = new String[N];  
        for (int i = 0; i < RANKS; i++)  
            for (int j = 0; j < SUITS; j++)  
                deck[SUITS*i + j] = rank[i] + " of " + suit[j];  
  
        for (int i = 0; i < N; i++) {  
            int r = i + (int) (Math.random() * (N-i));  
            String t = deck[r];  
            deck[r] = deck[i];  
            deck[i] = t;  
        }  
  
        for (int i = 0; i < N; i++)  
            System.out.println(deck[i]);  
    }  
}
```

build the deck

shuffle

print shuffled deck

Safnaravandamálið

Við erum með N gerðir af málsháttum sem settir eru í páskaegg af handahófi. Hve mörg páskaegg þurfum við að borða til að sjá alla málshættina

Hermun: Veljum i af handahófi frá 0 til $N-1$, stoppum þegar við höfum séð þau öll

Hvernig höldum við utan um hvaða málshætti við höfum séð?

Safnaravandamálið

Við erum með N gerðir af málsháttum sem settir eru í páskaegg af handahófi. Hve mörg páskaegg þurfum við að borða til að sjá alla málshættina

Hermun: Veljum i af handahófi frá 0 til $N-1$, stoppum þegar við höfum séð þau öll

Hvernig höldum við utan um hvaða málshætti við höfum séð?

Notum fylki af boolean, `found[i]` er true þegar við höfum séð málshátt i

```
public class CouponCollector {
    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);
        int cardcnt = 0;    // number of cards collected
        int valcnt = 0;    // number of distinct cards

        // do simulation
        boolean[] found = new boolean[N];
        while (valcnt < N) {
            int val = (int) (Math.random() * N);
            cardcnt++;
            if (!found[val]) {
                valcnt++;
                found[val] = true;
            }
        }

        // all N distinct cards found
        System.out.println(cardcnt);
    }
}
```


Villuleit

Við villuleit getur verið gott að prenta út ástand forrits

val	found						valcnt	cardcnt
	0	1	2	3	4	5		
	F	F	F	F	F	F	0	0
2	F	F	T	F	F	F	1	1
0	T	F	T	F	F	F	2	2
4	T	F	T	F	T	F	3	3
0	T	F	T	F	T	F	3	4
1	T	T	T	F	T	F	4	5
2	T	T	T	F	T	F	4	6
5	T	T	T	F	T	T	5	7
0	T	T	T	F	T	T	5	8
1	T	T	T	F	T	T	5	9
3	T	T	T	T	T	T	6	10

Með fylkjum getur ástandið orðið flókið

Samantekt

Fylki er runa af gildum af sama tagi

- Fylki eru innbyggð í Java
- `int[] v; // skilgreinir fylki v`
- Sköpum fylki v með
 - `int[] v = { 1,2,3}; // fast fylki`
 - `v = new int[N]; // minni tekið frá`
 - N má vera fasti eða reiknuð segð
 - Lengd fylkis er `v.length`
- Upphafsstillum fylki
 - `int, double` fylki eru með 0 eða 0.0
 - `boolean` með `false`
 - `String` með `null`, ekki `""` !

Fylki og for-lykkjur fara vel saman