Textavinnsla

Strengir: notaðir til að vinna með texta

Gildi: runa af Unicode stöfum

API

`public class String` *(Java string data type)*

	`String(String s)`	*create a string with the same value as* `s`
`int`	`length()`	*string length*
`char`	`charAt(int i)`	`i`*th character*
`String`	`substring(int i, int j)`	`i`*th through (*`j-1`*)st characters*
`boolean`	`contains(String sub)`	*does string contain* `sub` *as a substring?*
`boolean`	`startsWith(String pre)`	*does string start with* `pre`*?*
`boolean`	`endsWith(String post)`	*does string end with* `post`*?*
`int`	`indexOf(String p)`	*index of first occurrence of* `p`
`int`	`indexOf(String p, int i)`	*index of first occurrence of* `p` *after* `i`
`String`	`concat(String t)`	*this string with* `t` *appended*
`int`	`compareTo(String t)`	*string comparison*
`String`	`replaceAll(String a, String b)`	*result of changing* `a`*s to* `b`*s*
`String[]`	`split(String delim)`	*strings between occurrences of* `delim`
`boolean`	`equals(String t)`	*is this string's value the same as* `t`*'s?*

Strengjavinnsla

is the string a palindrome?

```
public static boolean isPalindrome(String s)
{
   int N = s.length();
   for (int i = 0; i < N/2; i++)
      if (s.charAt(i) != s.charAt(N-1-i))
         return false;
   return true;
}
```

extract file name and extension from a command-line argument

```
String s = args[0];
int dot = s.indexOf(".");
String base      = s.substring(0, dot);
String extension = s.substring(dot + 1, s.length());
```

print all lines in standard input that contain a string specified on the command line

```
String query = args[0];
while (!StdIn.isEmpty())
{
   String s = StdIn.readLine();
   if (s.contains(query)) StdOut.println(s);
}
```

print all the hyperlinks (to educational institutions) in the text file on standard input

```
while (!StdIn.isEmpty())
{
   String s = StdIn.readString();
   if (s.startsWith("http://") && s.endsWith(".edu"))
      StdOut.println(s);
}
```

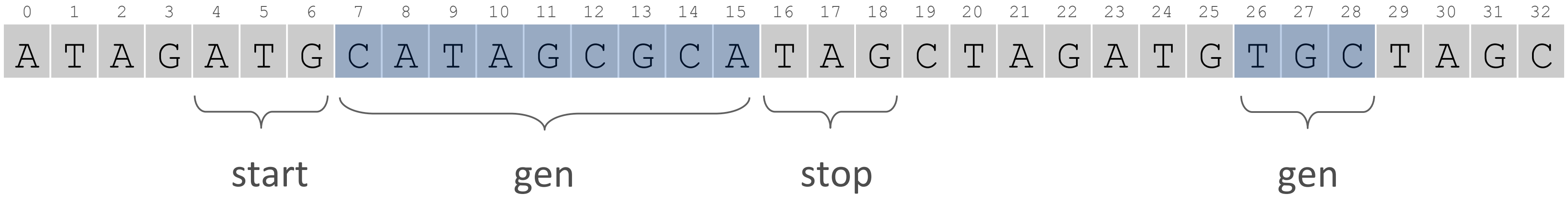
Genaleit

Erfðamengi: runa af stöfum (A,C,G,T)

Gen: kóði í erfðamenginu sem inniheldur uppskrift af prótíni

- Byrjar á ATG (start codon)
- Margfeldi af 3 kjarnsýrum (3 stafir kóða amínósýru)
- Endar á TAG, TAA eða TGA (stop codon)

Finnum öll möguleg gen í erfðamengi



Genaleit: reiknirit

Reiknirit: skönnum vinstri-hægri í erfðamenginu

- Ef við finnum start, setjum beg = i
- Ef við finnum stop og hlutstrengurinn frá beg til i er margfeldi af 3
    - prentum út genið
    - setjum beg = -1

i	codon		beg	gene	*remaining portion of input string*
	start	*stop*			
0			-1		ATAGATGCATAGCGCATAGCTAGATGTGCTAGC
1		TAG	-1		ATAGATGCATAGCGCATAGCTAGATGTGCTAGC
4	ATG		4		ATAGATGCATAGCGCATAGCTAGATGTGCTAGC
9		TAG	4		ATAGATGCATAGCGCATAGCTAGATGTGCTAGC
16		TAG	4	CATAGCGCA	ATAGATGCATAGCGCATAGCTAGATGTGCTAGC
20		TAG	-1		ATAGATGCATAGCGCATAGCTAGATGTGCTAGC
23	ATG		23		ATAGATGCATAGCGCATAGCTAGATGTGCTAGC
29		TAG	23	TGC	ATAGATGCATAGCGCATAGCTAGATGTGCTAGC

Genaleit: útfærsla

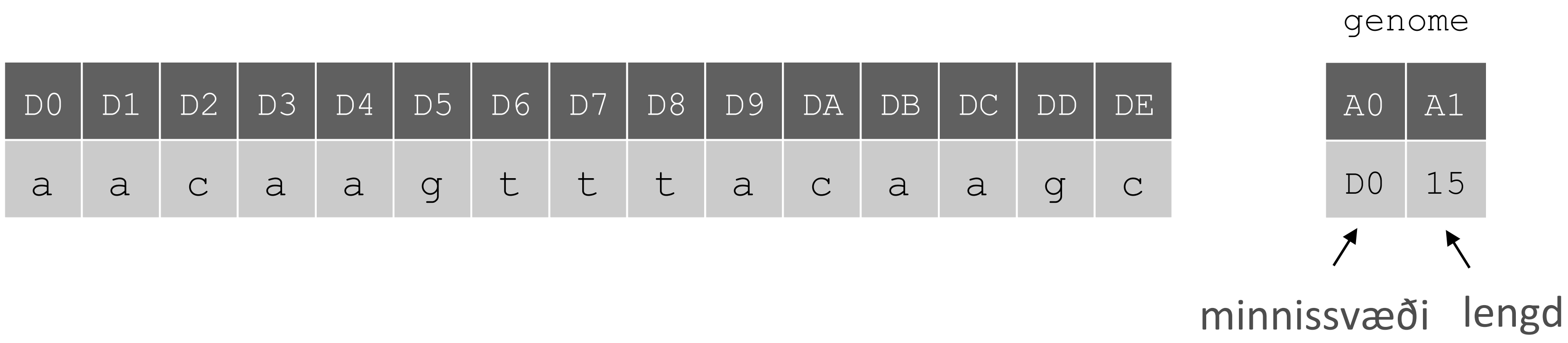
```java
public class GeneFind {
   public static void main(String[] args) {
      String start  = args[0];
      String stop   = args[1];
      String genome = StdIn.readAll();

      int beg = -1;
      for (int i = 0; i < genome.length() - 2; i++) {
         String codon = genome.substring(i, i+3);
         if (codon.equals(start)) beg = i;
         if (codon.equals(stop) && beg != -1) {
            String gene = genome.substring(beg+3, i);
            if (gene.length() % 3 == 0) {
               StdOut.println(gene);
               beg = -1;
            }
         }
      }
   }
}
```

```
% more genomeTiny.txt
ATAGATGCATAGCGCATAGCTAGATGTGCTAGC

% java GeneFind ATG TAG < genomeTiny.txt
CATAGCGCA
TGC
```

Strengir og minni

Strengir eru geymdir í minni sem fylki af stöfum

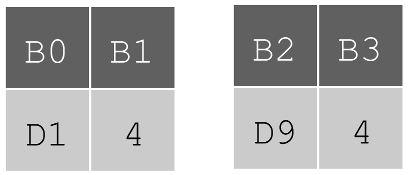
- genome = “`aacaagtttacaagc`”



D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE
a	a	c	a	a	g	t	t	t	a	c	a	a	g	c

A0	A1
D0	15

- `s = genome.substring(1,5);`
- `t = genome.substring(9,13);`



B0	B1
D1	4

B2	B3
D9	4

`s` og `t` vísa á ólíka strengi með sama gildi "`acaa`"

- `(s==t)` er false, en `(s.equals(t))` er true

ber saman tilvísun
(þ.e. minnissvæði)

ber saman stafina í strengjunum

Hlutir og klasar - samantekt

- Klasar í Java skilgreina ný gagnatög

- Hlutir í Java eru tilvik af klösum

- Breytur vísa á hluti (ólíkt frumstæðum gagnatögum, eins og með fylki)
  sjá bls. 352-358

- `a == b` er saman tilvísanir `a.equals(b)` (ef `a` hefur `equals` aðferð) ber saman gildin í hlutunum

Ný gagnatög

Til að búa til nýtt gagnatag, skilgreinum við

- Mengi gilda
- Aðgerðir skilgreindar á gildum

Java klasar: skilgreina gagnatag með

- Tilviksbreytu (instance variable) – mengi gilda
- Aðferð (methods) – aðgerðir á gildum
- Smið (constructor) – býr til og upphafsstillir hlut

Hleðslu gagnatag

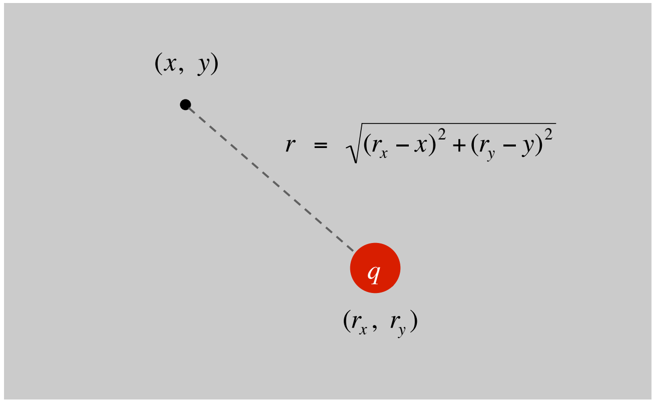
Búum til gagnatag fyrir rafhleðslu í punkti.

Mengi gilda: Þrjár rauntölur (x,y staðsetning og rafhleðsla)

Aðgerðir:

- Búa til nýja hleðslu á (r_x,r_y) með rafhleðslu q
- Finna spennu í punkt (a,b) vegna hleðslunnar
- Breyta yfir í streng

$$V = k\frac{q}{r}$$



r = fjarlægð milli (x, y) og (r_x, r_y)

k = Kúlombfasti = $8.99 \times 10^{9}\ \mathrm{N} \cdot \mathrm{m}^2 / \mathrm{C}^2$

Varúð

Ekki hafa áhyggjur af eðlisfræðinni

Það þarf ekki að skilja formúlurnar

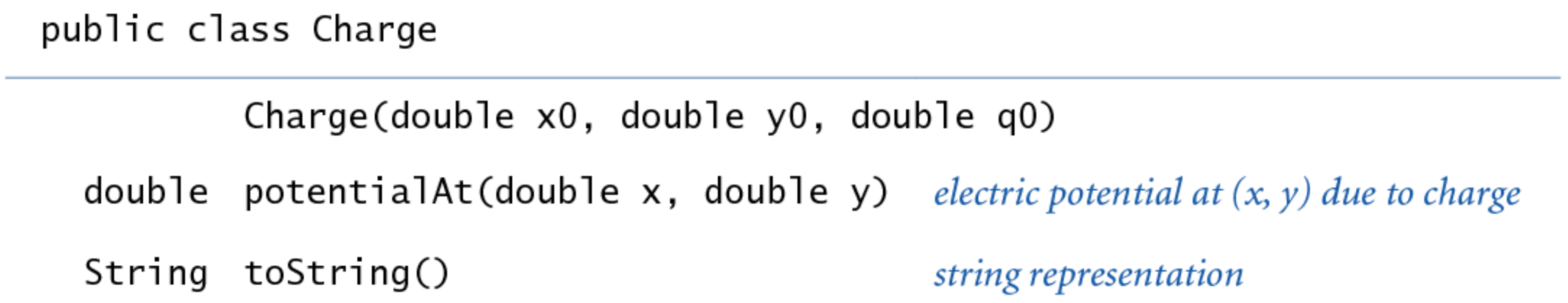
Bara breyta þeim í kóða

Hleðslu gagnatag

Búum til gagnatag fyrir rafhleðslu í punkti.

Mengi gilda: Þrjár rauntölur (x,y staðsetning og rafhleðsla)

API



```
public class Charge
```

	`Charge(double x0, double y0, double q0)`	
`double`	`potentialAt(double x, double y)`	*electric potential at (x, y) due to charge*
`String`	`toString()`	*string representation*

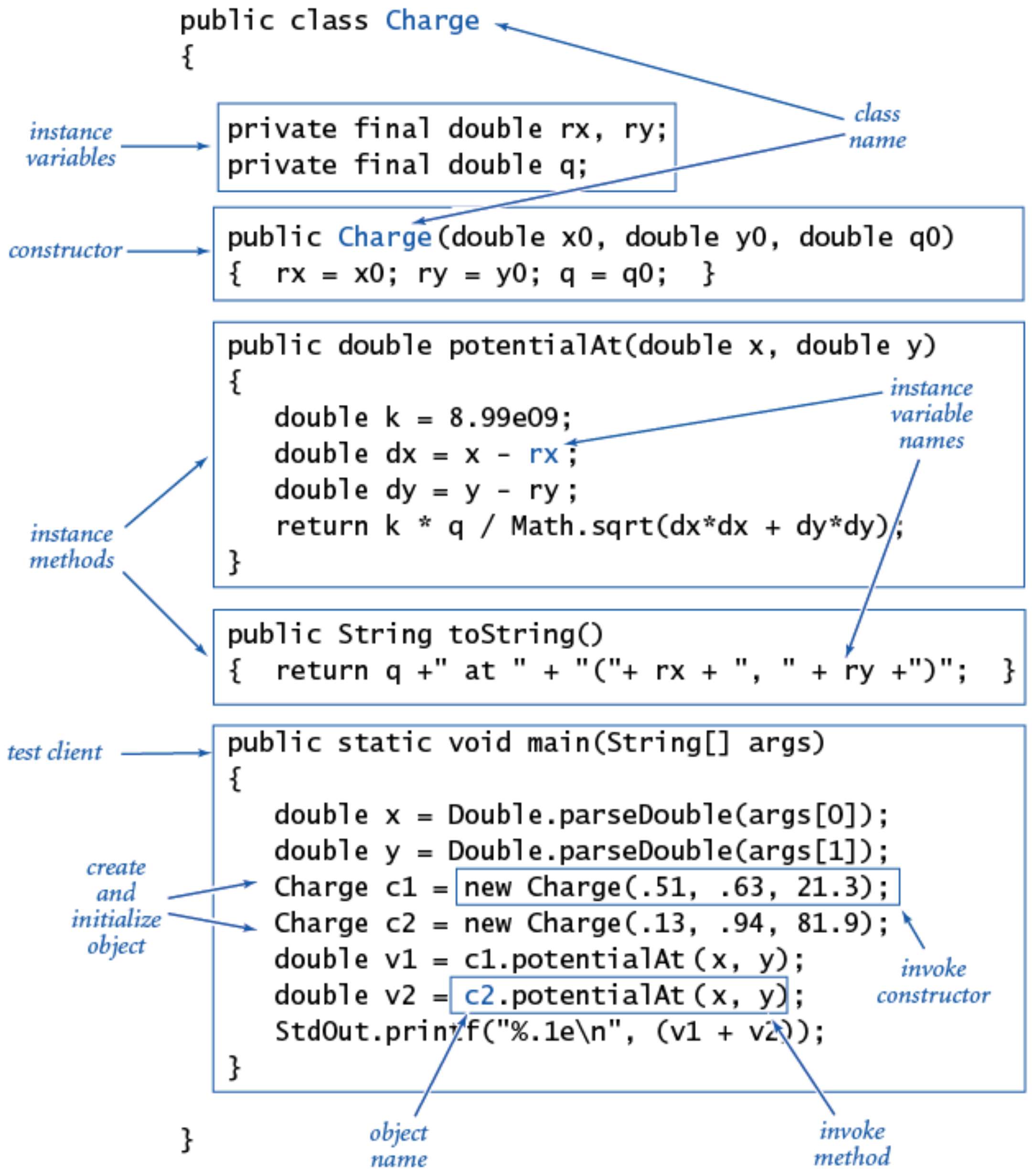
Prufuforrit: notar klasann og prufar aðferðir

```
public static void main(String[] args) {
   double x = Double.parseDouble(args[0]);
   double y = Double.parseDouble(args[1]);
   Charge c1 = new Charge(.51, .63, 21.3);
   Charge c2 = new Charge(.13, .94, 81.9);
   double v1 = c1.potentialAt(x, y);
   double v2 = c2.potentialAt(x, y);
   StdOut.println(c1);
   StdOut.println(c2);
   StdOut.println(v1 + v2);
}
```

Kallar á `toString()` aðferðina

```
% java Charge .50 .50
21.3 at (0.51, 0.63)
81.9 at (0.13, 0.94)
2.74936907085912e12
```

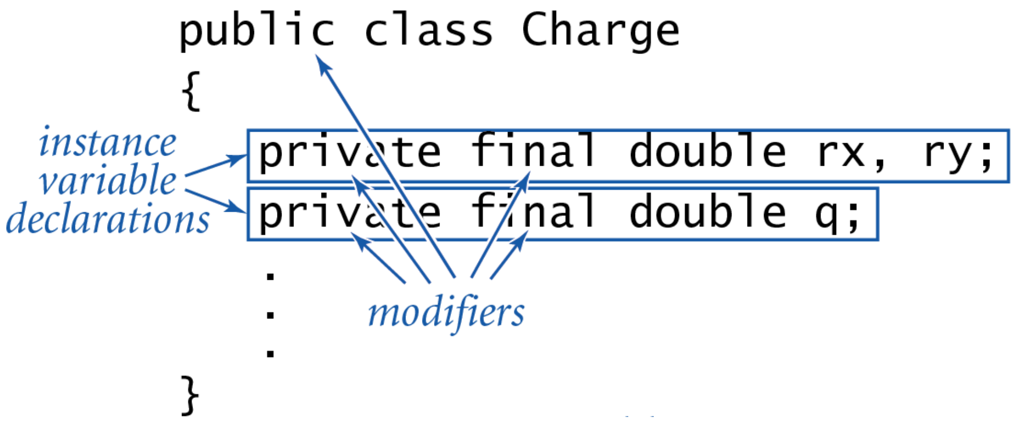
Yfirlit



```
public class Charge
{
   private final double rx, ry;
   private final double q;

   public Charge(double x0, double y0, double q0)
   {  rx = x0; ry = y0; q = q0;  }

   public double potentialAt(double x, double y)
   {
      double k = 8.99e09;
      double dx = x - rx;
      double dy = y - ry;
      return k * q / Math.sqrt(dx*dx + dy*dy);
   }

   public String toString()
   {  return q +" at " + "("+ rx + ", " + ry +")";  }

   public static void main(String[] args)
   {
      double x = Double.parseDouble(args[0]);
      double y = Double.parseDouble(args[1]);
      Charge c1 = new Charge(.51, .63, 21.3);
      Charge c2 = new Charge(.13, .94, 81.9);
      double v1 = c1.potentialAt(x, y);
      double v2 = c2.potentialAt(x, y);
      StdOut.printf("%.1e\n", (v1 + v2));
   }

}
```

Tilviksbreytur: skilgreina mengi gilda

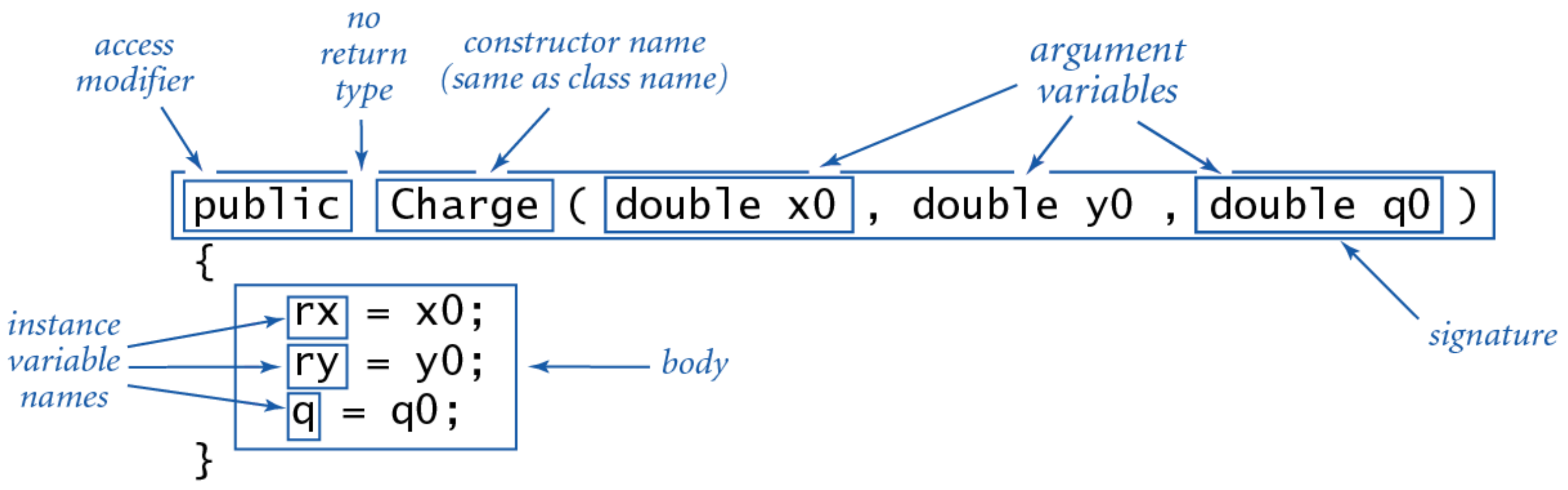
- Skilgreinum í klasa fyrir utan aðferðir
- Notum alltaf `private` aðgangsheimild
- Notum `final` breytinn fyrir tilviksbreytur sem breytast aldrei



```
public class Charge
{
   private final double rx, ry;
   private final double q;
   .
   .
   .
}
```

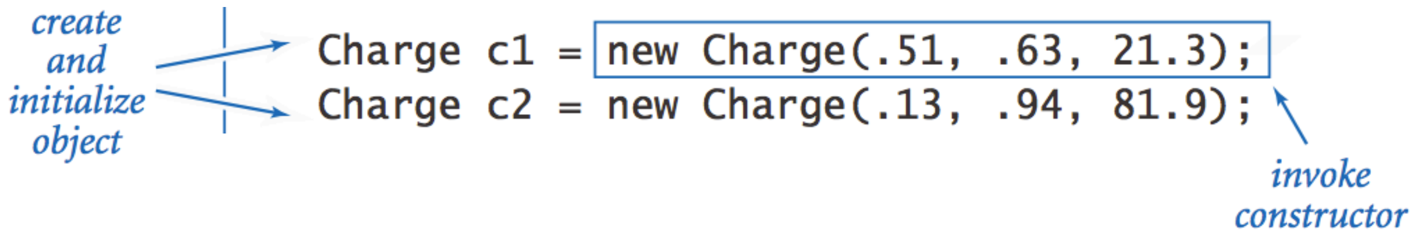
Smiður

Smiður: skilgreinir hvað gerist þegar við búum til hlut



```java
public Charge ( double x0 , double y0 , double q0 )
{
   rx = x0;
   ry = y0;
   q = q0;
}
```

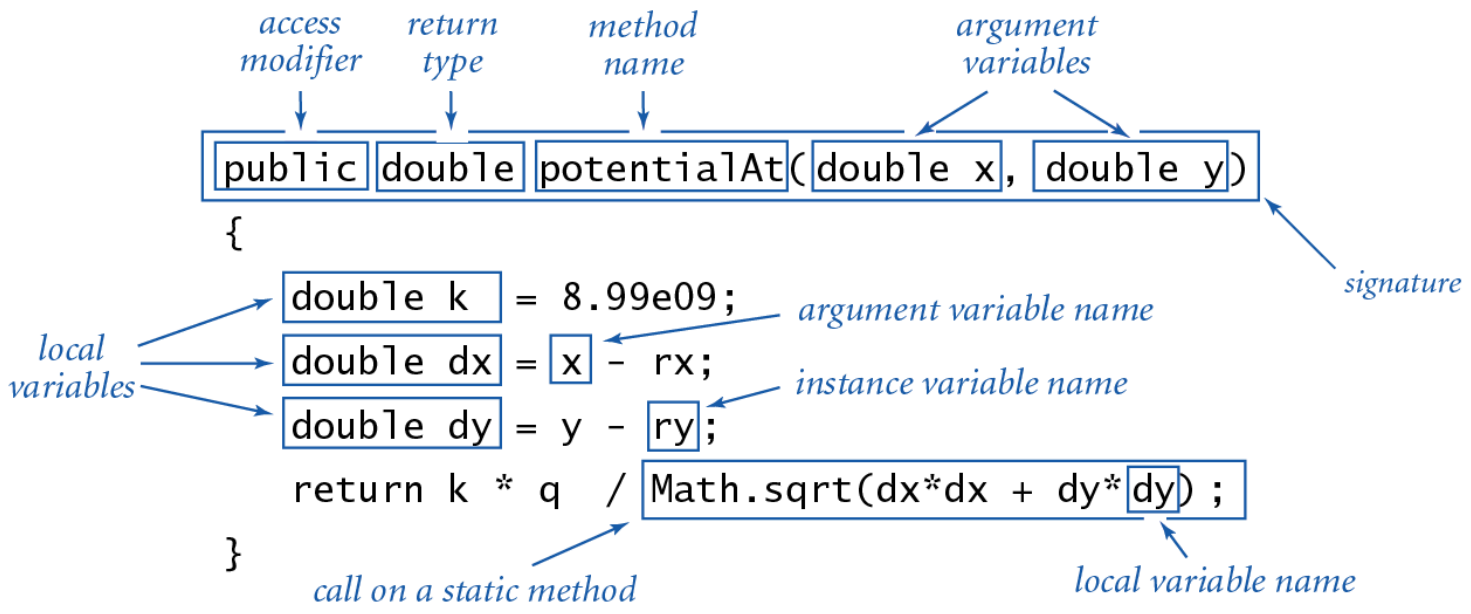
Kallað á smið: notum `new` til að búa til nýjan hlut



```java
Charge c1 = new Charge(.51, .63, 21.3);
Charge c2 = new Charge(.13, .94, 81.9);
```

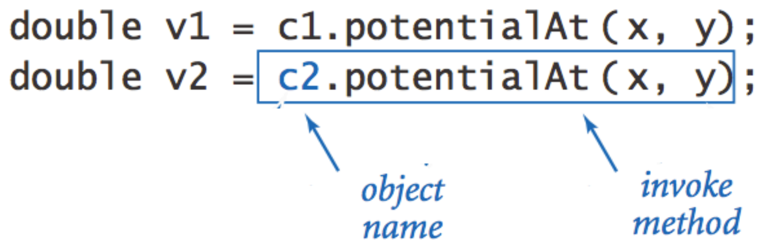
Aðferð

Aðferð: skilgreinir aðgerðir á hlut (tilviksbreytum)



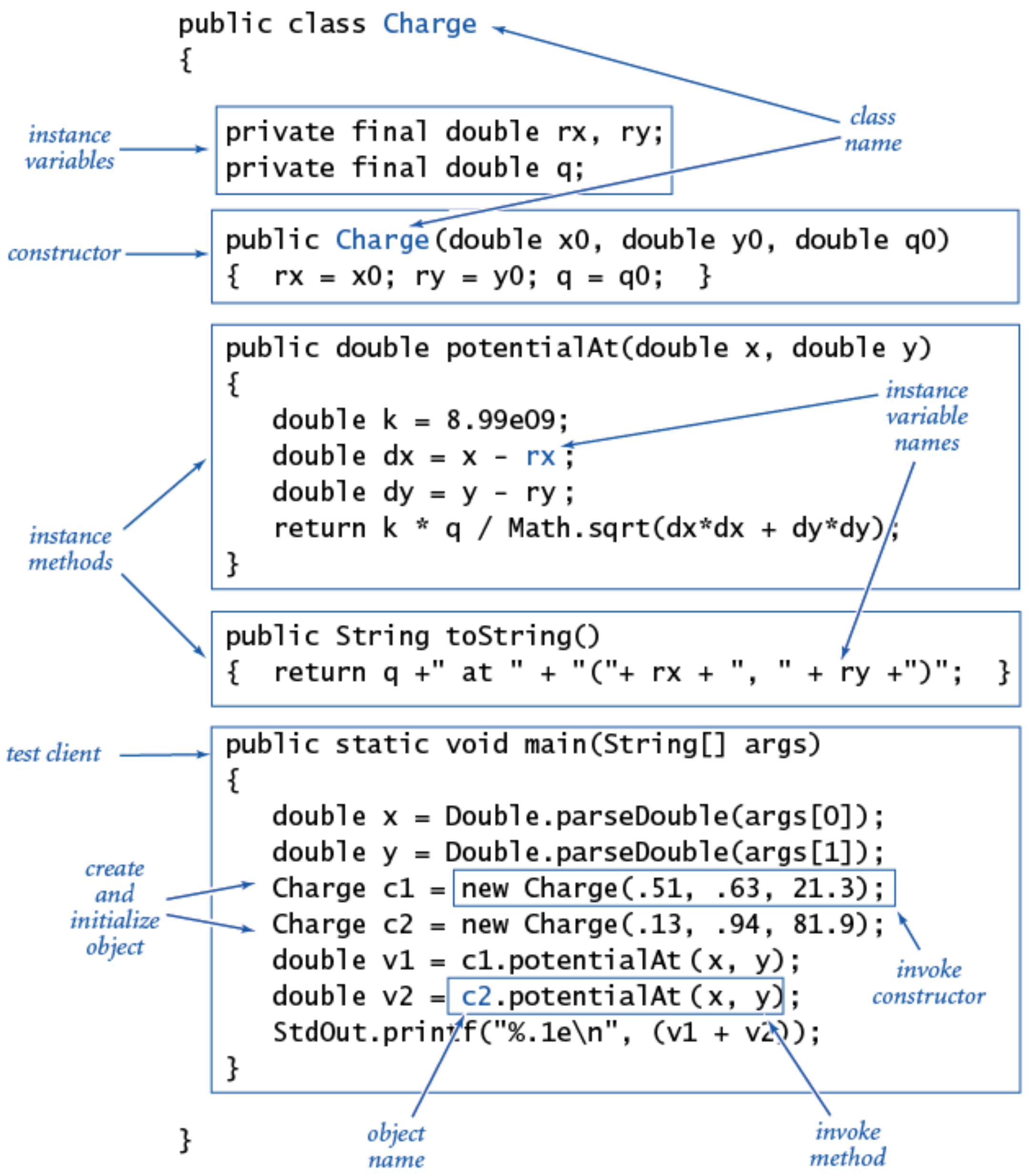
```
public double potentialAt(double x, double y)
{
    double k  = 8.99e09;
    double dx = x - rx;
    double dy = y - ry;
    return k * q / Math.sqrt(dx*dx + dy*dy) ;
}
```

Notkun á aðferð:



```
double v1 = c1.potentialAt(x, y);
double v2 = c2.potentialAt(x, y);
```

Geymt í Charge.java



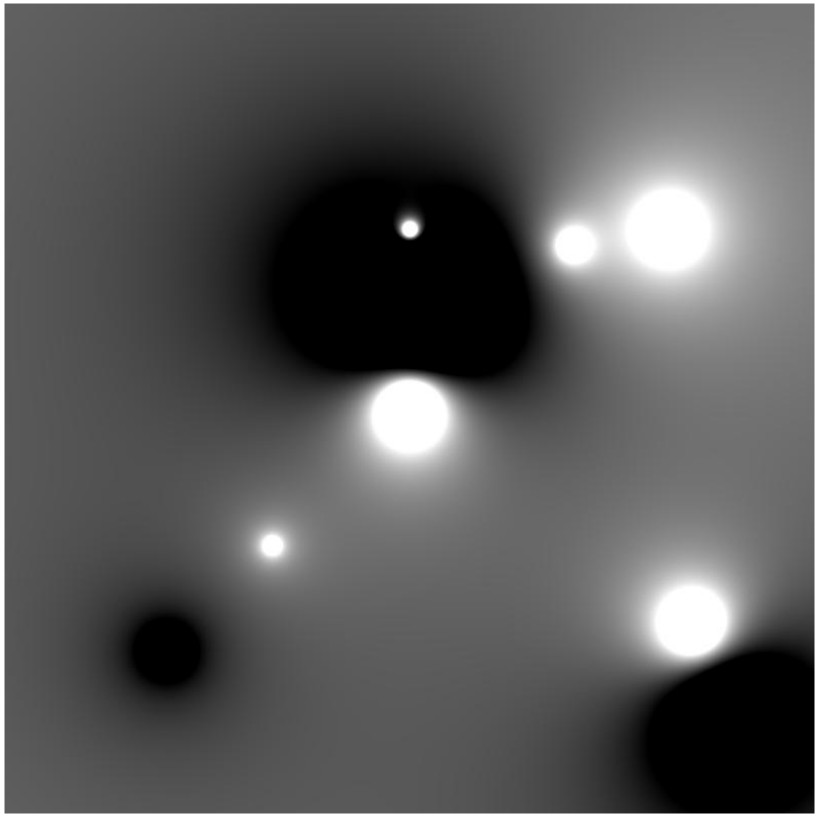
```
public class Charge
{
   private final double rx, ry;
   private final double q;

   public Charge(double x0, double y0, double q0)
   {  rx = x0; ry = y0; q = q0;  }

   public double potentialAt(double x, double y)
   {
      double k = 8.99e09;
      double dx = x - rx;
      double dy = y - ry;
      return k * q / Math.sqrt(dx*dx + dy*dy);
   }

   public String toString()
   {  return q +" at " + "("+ rx + ", " + ry +")";  }

   public static void main(String[] args)
   {
      double x = Double.parseDouble(args[0]);
      double y = Double.parseDouble(args[1]);
      Charge c1 = new Charge(.51, .63, 21.3);
      Charge c2 = new Charge(.13, .94, 81.9);
      double v1 = c1.potentialAt(x, y);
      double v2 = c2.potentialAt(x, y);
      StdOut.printf("%.1e\n", (v1 + v2));
   }

}
```

Birting spennu

Birting spennu: lesum N hleðslur (x,y,q) af staðal inntaki og reiknum samanlagða spennu í einingaferninginum (0,1) x (0,1)

```
% more charges.txt
9
.51 .63 -100
.50 .50   40
.50 .72   10
.33 .33    5
.20 .20  -10
.70 .70   10
.82 .72   20
.85 .23   30
.90 .12  -50
```

```
% java Potential < charges.txt
```



Birting spennu

Höldum utan um allar hleðslurnar í fylki af hlutum.

```
% more charges.txt
9
.51 .63 -100
.50 .50   40
.50 .72   10
.33 .33    5
.20 .20  -10
.70 .70   10
.82 .72   20
.85 .23   30
.90 .12  -50
```

```
// read in the data
int N = StdIn.readInt();
Charge[] a = new Charge[N];
for (int i = 0; i < N; i++) {
   double x0 = StdIn.readDouble();
   double y0 = StdIn.readDouble();
   double q0 = StdIn.readDouble();
   a[i] = new Charge(x0, y0, q0);
}
```

Birting spennu

```
// plot the data
int SIZE = 512;
Picture pic = new Picture(SIZE, SIZE);
for (int i = 0; i < SIZE; i++) {
   for (int j = 0; j < SIZE; j++) {
      double V = 0.0;
      for (int k = 0; k < N; k++) {
         double x = 1.0 * i / SIZE;
         double y = 1.0 * j / SIZE;
         V += a[k].potentialAt(x, y);
      }
      Color color = getColor(V);
      pic.set(i, SIZE-1-j, color);
   }
}
pic.show();
```

reiknum út lit sem fall af spennu