

TDA 548: Grundläggande Programvaruutveckling

Föreläsning 1, vecka 8:

Att förbereda sig för tentan

Magnus Myréen

Chalmers, läsperiod 1, 2016-2017

Idag

Idag: talar om tentan, förberedelse, exempel, mm.

Inget nytt material.

Viktigt i denna föreläsning:

- ▶ hur förbereder man sig?
- ▶ hur skriver man bra svar på tentan?
- ▶ att maximera poäng summan...
- ▶ hur ser en tenta ut?

Repetition av:

- ▶ exceptions
- ▶ hur man ritar grafik

Exempel & Quiz.

Tentan närmar sig!

Tentan: granskar om ni kan programmera i Java

Fråga: hur lär man sig programmera?

... genom att läsa kod? **Nej.**

... genom att läsa en bok? **Nej.**

... genom att lyssna på föreläsningar? **Nej.**

Testar: kan ni tillämpa kursens material för att lösa programmeringsuppgifter

Svar: genom att programmera!

Tips: **Planera** din kod *innan du börjar skriva.*

Kompilera ofta.

Testa halvfärdig kod.

Dela programmeringsproblemet i mindre bitar.

... men jag kan programmera med IntelliJ, inte med penna och papper.

Vad är ert mål?

Vem vill gå om kursen nästa år?

Vem vill sitta omtenta i januari?

*Siktar ni att **komma igenom**?*

*Eller siktar ni att **få bra vitsord**?*

Majoriteten av dessa hamnar att sitta om tentan eller gå om kursen.

Dessa kommer igenom kursen.

Maximera era chanser!

7 hela dagar till tentan

Kommer ni ihåg räknaren?

Tiden går fort!

Hur mycket Java lärde ni er de *senaste 7 dagarna*?

Hur använda tiden bäst?

Vad är din svaghet?

Exempel svar: rekursion

läs om rekursion,

det handlar ju om metodanrop (hur fungerar de?) ... och typer
gör uppgifter,

hitta nya uppgifter på nätet

programmera, programmera, programmera!



Mycket viktigt.

... och **ännu viktigare** är det att man stänger telefonen, Facebook, etc.

- "Multitasking riskerar att eliminera vår kapacitet till djupinläarning och att förstå sammanhang."
- "Eric Kandel menar att det endast är när vi förmår vara djupt koncentrerade på ny information som vi kan koppla ihop den på ett meningsfullt sätt med kunskap vi redan har. Sådana kopplingar är nödvändiga för att få sammanhang och lagra kunskaper i det obegränsade långtidsminnet.

... och **ännu viktigare** är det att man **stänger telefonen, Facebook, etc.**

“[...] It is also what makes multitasking a poor long-term strategy for learning.”

<http://www.thenewatlantis.com/publications/the-myth-of-multitasking>

http://brainrules.blogspot.se/2008/03/brain-cannot-multitask_16.html

<http://www.forbes.com/sites/douglasmerrill/2012/08/17/why-multitasking-doesnt-work/>

<http://www.psychologytoday.com/blog/brain-wise/201209/the-true-cost-multi-tasking>

<http://psychology.about.com/od/cognitivepsychology/a/costs-of-multitasking.htm>

<http://www.npr.org/templates/story/story.php?storyId=95256794>

... och **ännu viktigare** är det att man **stänger** telefonen, Facebook, etc.

Ett par dagar innan tentan...

Reservera en dag (och en gammal tenta) för en **övningstenta**.

Sitt i **fyra timmar** och **skriva svar** på frågorna **på papper**.

(Precis som på tentan...)

(Absolut inga telefoner, ingen laptop, inget prat med vänner, mm.)

Ta en paus!

Läs sedan igenom dina svar. **Hur gick det?** Hur många poäng tror du?

Vad behöver du kunna bättre?

Dagen innan tentan

Gå igenom dina anteckningar.

Kolla vad tycker du är viktigt.

Stärk dina svagaste områden.

Läs sedan igenom dina svar. **Hur gick det?** Hur många poäng tror du?
Vad behöver du kunna bättre?

Packa innan du lägger dig.

Sov en lång natt. Sömn är viktigt!

Hur ser er tentan ut?

Gamla tentor hittar ni på kurshemsidan:

<http://www.cse.chalmers.se/edu/course/TDA548/#resources>

<http://www.cse.chalmers.se/~myreen/tda548/#tentan>

Er tenta:

Max 60 poäng: 40 från Joachims uppgifter och 20 från mina uppgifter

24 poäng för 3

38 poäng för 4

48 poäng för 5

På tentan

Tid: endast 4 timmar.

Läs snabbt igenom alla frågor.

Bestäm ordningen, vilken svara först osv.

Svara på alla frågor, och försök skriva någonting på alla delar.

Maximera dina chanser!

Viktigt: undvik att fastna.

Bestäm hur länge du tänker använda på denna fråga.

När tiden är slut för frågan, börja med nästa.

Strategi: hur många minuter per poäng?

På tentan (forts)

Men om jag inte vet, inte kan...

- ▶ försök bryta ner problemet i mindre steg
- ▶ beskriv hur du tänker lösa problemet
- ▶ skriv din lösning så gott du kan

Rättaren **kan inte ge poäng för tomt papper!**

... men om det finns någonting där **kanske** du får **ett eller två poäng**.

Fråga om något är oklart. **Skriv vad du antar.**

Underhållning

Tentarättning?

<https://www.youtube.com/watch?v=t8XMeocLflc>

... livet efter tentan

Resultat via Ladok (skickas till er per epost)

Tentagranskning

... omtenta? (jag hoppas inte)

... men vad ska vi kunna?!

Från Joachims del:

variabler, loopar, typer, referensvärden, arrays, att skriva klass, och mer

Från min del:

att lösa problem med objektorietering och abstraktion (interface, arv)

tillstånd i objektorienterade program: mutable/immutable, problematik

rekursion av olika slag: metodanrop (t.ex. fib) och typer (t.ex. Many3D)

Ni ska kunna tillämpa kunskaperna i samband med problemlösning

Repetition av:

- ▶ exceptions
- ▶ hur man ritar grafik

Repetition av **exceptions**

Syntax

Kastar en exception:

```
throw exception
```

uttrycket måste ha typen **Exception**

Man fångar exceptions med:

```
try {
```

```
    kod som kanske kastar exception
```

```
} catch (MyExceptionClass1 name) {
```

```
    kod som reagerar till exception  
    som kom i namn variabeln.
```

```
} catch (MyExceptionClass2 name) {
```

```
    ...
```

```
} finally {
```

```
    kod som alltid körs
```

```
}
```

Fångar exceptions av typen/klassen
MyExceptionClass1

Om exception inte fångades så kollar
vi om det är en **MyExceptionClass2**

Behöver inte finnas.

Kod som alltid utförs oavsett om
ett fel inträffade eller ej. Vanligen
för att "städa upp" tex stänga filer.

Varför exceptions?

Separerar felhantering från “vanlig” kod.

Man programmerar först som om felen inte inträffade, sen tar man hand om felen.

Enkel felpropagering

... tackvare propagering till närmsta felhanterare. Ibland långsamt.

Ger gruppering av feltyper

Fel är ju objekt som tillhör klasser, **arv!**

Terminologi för exceptions

`IllegalArgumentException`,
`RuntimeException`, mm.

Exceptions kan vara:

fördefinierade (finns i Javas API), eller

egendefinierade (du får skriva själv)

Exceptions kan vara:

checked (måste hanteras eller nämnas i signaturen), eller

unchecked (kan ignoreras)

Exempel strax....

Att kasta exceptions

```
public class Circle {  
    private int radius = 0;  
  
    public void setRadius(int r) {  
        if (0 <= r) {  
            radius = r;  
        } else {  
            throw new IllegalArgumentException("Cannot have negative radius");  
        }  
    }  
}
```

Skapar ett nytt exception objekt av typen **IllegalArgumentException**.

Kastar exception objektet.

Att kasta exceptions

Man kan skriva i signaturen att en exception kanske kastas.

Obs. måste nämnas i signaturen ifall det är en **checked exception**.

```
public class Circle {  
    private int radius = 0;  
  
    public void setRadius(int r) throws IllegalArgumentException {  
        if (0 <= r) {  
            radius = r;  
        } else {  
            throw new IllegalArgumentException("Cannot have negative radius.");  
        }  
    }  
}
```

Skapar ett nytt exception objekt av typen **IllegalArgumentException**.

Kastar exception objektet.

Att kasta egna exceptions

Ärver *inte* `RuntimeException`, är alltså en *checked* exception.

```
class CircleEx extends Exception {  
    public CircleEx(String str) { super(str); }  
}
```

Ärver `RuntimeException`, är alltså en *unchecked* exception.

```
class CircleRunEx extends RuntimeException {  
    public CircleRunEx(String str) { super(str); }  
}
```

```
public class Circle2 {  
  
    private int radius;  
  
    public void setRadius(int r) {  
        if (0 <= r) {  
            radius = r;  
        } else {  
            throw new CircleRunEx("Cannot have negative radius.");  
        }  
    }  
}
```

Här kastar vi vår egen exception.

Exempel: felutskriften och avsluta

I vissa fall är det bäst att **avsluta** programmet.

```
public static int getInt() {  
    ...// ev annan kod  
    int i = 0;  
    try {  
        String str = myIn.nextLine();  
        i = Integer.parseInt(str);  
    } catch (NumberFormatException e) {  
        System.out.println("Read failed!");  
        e.printStackTrace();  
        System.exit(1);  
    }  
    return i;  
}
```

Skriver ut alla metodanrop som gjorts för att komma hit.

Stänger programmet.

Exempel: ta hand om felet, vidtag åtgärd

Vi kan ta hand om felet och fortsätta eller försöka igen.

```
public static int getInt() {  
    int i=0;  
    // ev. annan kod här  
    boolean notReady = true;  
    while (notReady) {  
        try {  
            String str = myIn.nextLine();  
            i = Integer.parseInt(str);  
            notReady = false;  
        } catch (NumberFormatException e) {  
            System.out.println("Read failed!");  
            System.out.print(" Ge talet på nytt: ");  
        }  
    }  
    return i;  
}
```

try-catch inuti en while loop.

Vad ni behöver veta om **grafik**

Vad ni behöver veta om **grafik**

Slag 1:

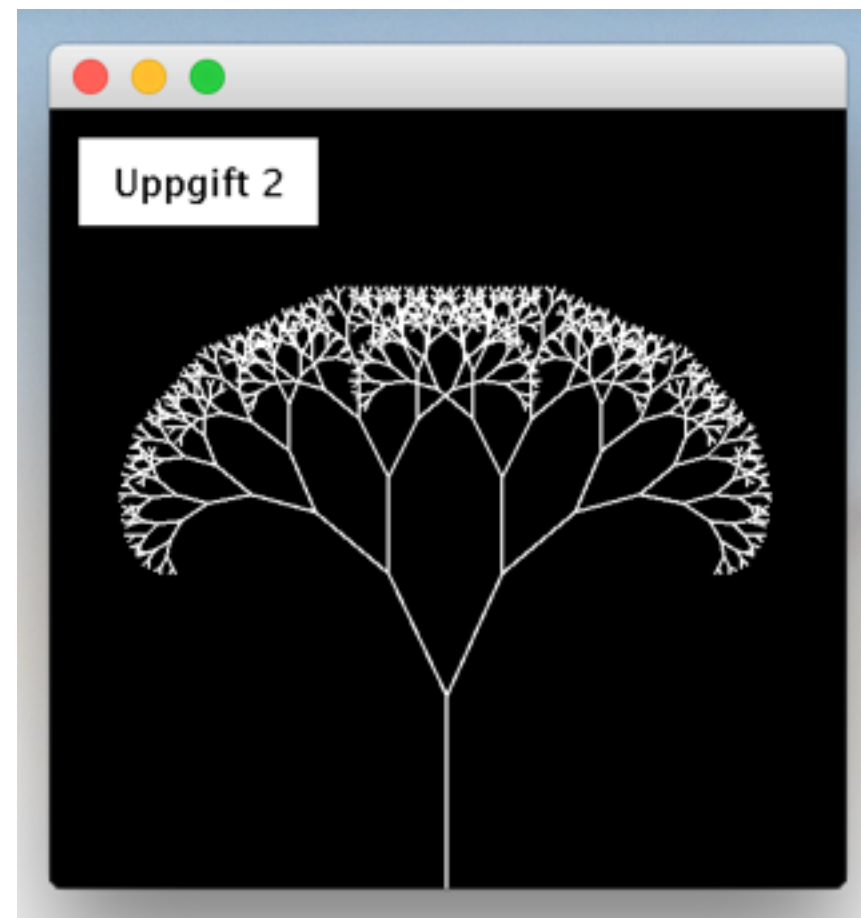
Att placera ut knappar mm för att skapa grafiska användargränssnitt, t.ex. med *swing*.



Ni **behöver inte** kunna detta.

Slag 2:

Att rita linjer, rektanglar, ovaler mm. genom att använda *Graphics* eller *Graphics2D*.



Ni **bör kunna rita med Graphics**.

Exempel: att rita i Java

```
import java.awt.*;  
import javax.swing.*;
```

```
class DemoPanel extends JPanel {
```

```
    public void paintComponent(Graphics g) {  
        super.paintComponent(g);  
        g.drawRect(10, 10, 10, 10);  
        g.drawRect(30, 10, 20, 10);  
        int x = this.getWidth();  
        int y = this.getHeight();  
        g.drawRect(x/5, y/5, x-2*x/5, y/3);  
    }
```

```
}
```

```
public class Demo1 extends JFrame {
```

```
    public Demo1() {  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setSize(200,150);  
        setLocation(50,50);  
        DemoPanel panel = new DemoPanel();  
        add(panel);  
        setVisible(true);  
    }
```

```
    public static void main(String[] args) {  
        Demo1 f = new Demo1();  
    }
```

```
}
```

Ärver JPanel som vi ska rita i.

Överskuggar paintComponent metoden.

I Java ritar man med Graphics objekt.

Vi ritar tre rektanglar.

Den sista anpassar sig till storleken av JPanelen, dvs använder x och y.

Ni behöver inte kunna skriva den svarta koden.

Exempel: att rita i Java

```
import java.awt.*;
import javax.swing.*;

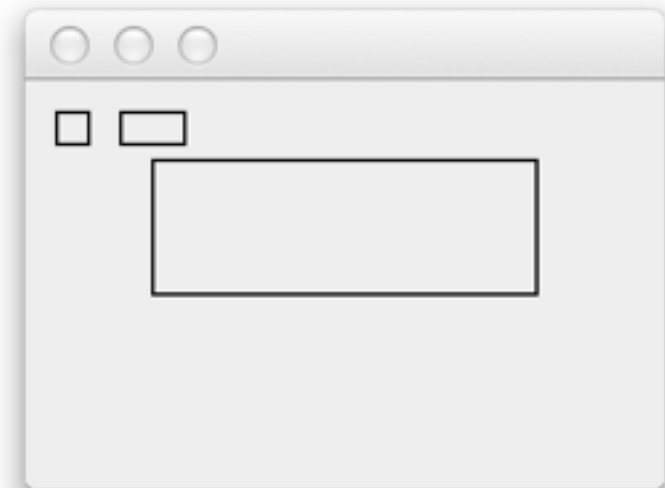
class DemoPanel extends JPanel {

    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.drawRect(10, 10, 10, 10);
        g.drawRect(30, 10, 20, 10);
        int x = this.getWidth();
        int y = this.getHeight();
        g.drawRect(x/5, y/5, x-2*x/5, y/3);
    }
}

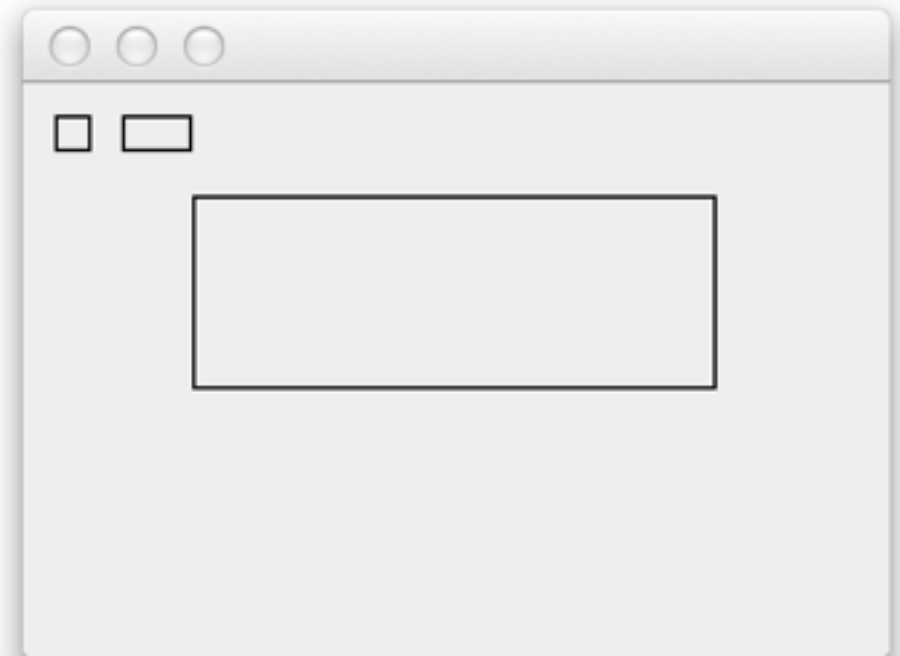
public class Demo1 extends JFrame {

    public Demo1() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(200,150);
        setLocation(50,50);
        DemoPanel panel = new DemoPanel();
        add(panel);
        setVisible(true);
    }

    public static void main(String[] args) {
        Demo1 f = new Demo1();
    }
}
```



Om ändrar på fönstrets storlek ändras också den sista rektangeln.



Idag

Idag: talar om tentan, förberedelse, exempel, mm.

Inget nytt material.

Viktigt i denna föreläsning:

- ▶ hur förbereder man sig?
- ▶ hur skriver man bra svar på tentan?
- ▶ att maximera poäng summan...
- ▶ hur ser en tenta ut?

Repetition av:

- ▶ exceptions
- ▶ hur man ritar grafik

Exempel, Quiz.

Exempel

En tentauppgift med lösning: Skriv en metod

```
void columnSort(int[][] x)
```

som givet en matris x sorterar varje **kolumn**.

Skriv också en metod `print(int[][] x)`

som skriver ut en matris. Skriv sedan kod som skapar första matrisen nedan, anropar `columnSort` och sedan `print` som då skriver den andra matrisen nedan.

Exempel: om x är första 5x7 matrisen nedan så skall metoden `columnSort` arrangera matrisen som andra matrisen nedan.

```
int[][] a = {  
    { 12,  6,  7, 17, 18, 19,  8 },  
    {  0, 14,  8, 15,  5,  3,  2 },  
    {  7,  2,  1,  6,  9, 18, 21 },  
    {  1,  5,  9,  3,  7, 11,  2 },  
    { 89, 12,  6,  1,  0, 19, 27 }  
};
```

12	6	7	17	18	19	8
0	14	8	15	5	3	2
7	2	1	6	9	18	21
1	5	9	3	7	11	2
89	12	6	1	0	19	27

skall bli:

0	2	1	1	0	3	2
1	5	6	3	5	11	2
7	6	7	6	7	18	8
12	12	8	15	9	19	21
89	14	9	17	18	19	27

Tips: i en loop, byt plats på element som inte är i rätt ordning.

En tentauppgift med lösning: Skriv en metod

```
void columnSort(int[][] x)
```

som givet en matris x sorterar varje **kolumn**.

Skriv också en metod `print(int[][] x)`

som skriver ut en matris. Skriv sedan kod som skapar första matrisen nedan, anropar `columnSort` och sedan `print` som då skriver den andra matrisen nedan.

Exempel: om x är första 5x7 matrisen nedan så skall metoden `columnSort` arrangera matrisen som andra matrisen nedan.

Låter detta svårt?

Ifall det gör det lönar det sig att börja med att försöka lösa ett lättare problem.

12	6	7	17	18	19	8
0	14	8	15	5	3	2
7	2	1	6	9	18	21
1	5	9	3	7	11	2
89	12	6	1	0	19	27

skall bli:

0	2	1	1	0	3	2
1	5	6	3	5	11	2
7	6	7	6	7	18	8
12	12	8	15	9	19	21
89	14	9	17	18	19	27

Tips: i en loop, byt plats på element som inte är i rätt ordning.

Sortering

Hur sorterar man en array av ints?

```
int[] a = { 5, 1, 2, 6, 3, 9 };
```

Bubblesort:

<https://www.youtube.com/watch?v=lyZQPjUT5B4>

Sortering

Hur sorterar man ett fält av ints?

Denna kod byter två element i fältet.

```
int tmp0 = a[i];  
int tmp1 = a[i+1];  
a[i] = tmp1;  
a[i+1] = tmp0;
```

Sortering

Hur sorterar man ett fält av ints?

Denna kod byter två element i fältet, ifall de inte är i ordning.

```
if (a[i] > a[i+1]) {  
    int tmp0 = a[i];  
    int tmp1 = a[i+1];  
    a[i] = tmp1;  
    a[i+1] = tmp0;  
}
```

Sortering

Hur sorterar man ett fält av ints?

Denna kod **bubblar upp det största elementet** till slutet av fältet.

```
for (int i=0; i<a.length-1; i++) {  
    if (a[i] > a[i+1]) {  
        int tmp0 = a[i];  
        int tmp1 = a[i+1];  
        a[i] = tmp1;  
        a[i+1] = tmp0;  
    }  
}
```

Sortering

Hur sorterar man ett fält av ints?

Denna kod **sorterar fältet** genom att köra `a.length-1` “bubblingar”.

```
for (int j=0; j<a.length-1; j++) {  
    for (int i=0; i<a.length-1; i++) {  
        if (a[i] > a[i+1]) {  
            int tmp0 = a[i];  
            int tmp1 = a[i+1];  
            a[i] = tmp1;  
            a[i+1] = tmp0;  
        }  
    }  
}
```

Sortering

Hur sorterar man ett fält av ints?

Denna kod **sorterar fältet** genom att köra `a.length-1` “bubblingar”.

```
for (int j=0; j<a.length-1; j++) {  
    for (int i=0; i<a.length-1; i++) {  
        if (a[i] > a[i+1]) {  
            int tmp0 = a[i];  
            int tmp1 = a[i+1];  
            a[i] = tmp1;  
            a[i+1] = tmp0;  
        }  
    }  
}
```

Hur vet vi om koden är korrekt?

Sortering

Hur sorterar man ett fält av ints?

Vi kan skriva ut lite debug data för att testa programmet.

```
System.out.print("a = { ");  
for (int i=0; i<a.length; i++) {  
    System.out.print(a[i] + " ");  
}  
System.out.println("}");
```

Exempel utskrift:

```
a = { 5 1 2 6 3 9 }
```

Sortering av ett fält

Hela koden:

```
public class Test {  
  
    public static void print(int[] a) {  
        System.out.print("a = { ");  
        for (int i=0; i<a.length; i++) {  
            System.out.print(a[i] + " ");  
        }  
        System.out.println("}");  
    }  
  
    public static void main(String[] args) {  
  
        int[] a = { 5,1,2,6,3,9 };  
  
        print(a);  
  
        for (int j=0; j<a.length-1; j++) {  
            for (int i=0; i<a.length-1; i++) {  
                if (a[i] > a[i+1]) {  
                    int tmp0 = a[i];  
                    int tmp1 = a[i+1];  
                    a[i] = tmp1;  
                    a[i+1] = tmp0;  
                }  
            }  
        }  
  
        print(a);  
    }  
}
```

En tentauppgift med lösning: Skriv en metod

`void columnSort(int[][] x)`

som givet en matris `x` sorterar varje **kolumn**.

Skriv också en metod `print(int[][] x)`

som skriver ut en matris. Skriv sedan kod som skapar första matrisen nedan, anropar `columnSort` och sedan `print` som då skriver den andra matrisen nedan.

Exempel: om `x` är första 5x7 matrisen nedan så skall metoden `columnSort` arrangera matrisen som andra matrisen nedan.

12	6	7	17	18	19	8
0	14	8	15	5	3	2
7	2	1	6	9	18	21
1	5	9	3	7	11	2
89	12	6	1	0	19	27

skall bli:

0	2	1	1	0	3	2
1	5	6	3	5	11	2
7	6	7	6	7	18	8
12	12	8	15	9	19	21
89	14	9	17	18	19	27

Sortering av kolumner i matris

Hur representerar man matriser i Java?

```
int[][] a = {  
    { 12,  6,  7, 17, 18, 19,  8 },  
    {  0, 14,  8, 15,  5,  3,  2 },  
    {  7,  2,  1,  6,  9, 18, 21 },  
    {  1,  5,  9,  3,  7, 11,  2 },  
    { 89, 12,  6,  1,  0, 19, 27 }  
};
```

Sortering av kolumner i matris

Vi börjar med att skriva en print metod:

for går genom alla
kolumner

for går genom alla
element i raden

skriver ut ett element

byter till nästa rad

```
public static void print(int[][] x) {  
    for (int i=0; i<x.length; i++) {  
        System.out.print("Rad " + i + ": ");  
        for (int j=0; j<x[i].length; j++) {  
            System.out.print(x[i][j] + " ");  
        }  
        System.out.println("");  
    }  
}
```

Exempel utskrift:

```
Rad 0: 12 6 7 17 18 19 8  
Rad 1: 0 14 8 15 5 3 2  
Rad 2: 7 2 1 6 9 18 21  
Rad 3: 1 5 9 3 7 11 2  
Rad 4: 89 12 6 1 0 19 27
```

Sortering av kolumner i matris

Så här kan vi ändra koden för sortering av ett fält till kod som sorterar den första kolumnen.

```
for (int j=0; j<a.length-1; j++) {  
    for (int i=0; i<a.length-1; i++) {  
        if (a[i][0] > a[i+1][0]) {  
            int tmp0 = a[i][0];  
            int tmp1 = a[i+1][0];  
            a[i][0] = tmp1;  
            a[i+1][0] = tmp0;  
        }  
    }  
}
```

Sortering av kolumner i matris

Med en for-loop kan vi få alla kolumner sorterade.

```
for (int k=0; k<a[0].length; k++) {  
    for (int j=0; j<a.length-1; j++) {  
        for (int i=0; i<a.length-1; i++) {  
            if (a[i][k] > a[i+1][k]) {  
                int tmp0 = a[i][k];  
                int tmp1 = a[i+1][k];  
                a[i][k] = tmp1;  
                a[i+1][k] = tmp0;  
            }  
        }  
    }  
}
```

Sortering av kolumner i matris

... och så här sätter vi koden in i en metod.

```
public static void columnSort(int[][] a) {  
    for (int k=0; k<a[0].length; k++) {  
        for (int j=0; j<a.length-1; j++) {  
            for (int i=0; i<a.length-1; i++) {  
                if (a[i][k] > a[i+1][k]) {  
                    int tmp0 = a[i][k];  
                    int tmp1 = a[i+1][k];  
                    a[i][k] = tmp1;  
                    a[i+1][k] = tmp0;  
                }  
            }  
        }  
    }  
}
```

Svar på tentafrågan

Så här kör vi programmet:

```
$ javac Test2.java
```

```
$ java Test2
```

```
Rad 0: 12 6 7 17 18 19 8
```

```
Rad 1: 0 14 8 15 5 3 2
```

```
Rad 2: 7 2 1 6 9 18 21
```

```
Rad 3: 1 5 9 3 7 11 2
```

```
Rad 4: 89 12 6 1 0 19 27
```

```
Rad 0: 0 2 1 1 0 3 2
```

```
Rad 1: 1 5 6 3 5 11 2
```

```
Rad 2: 7 6 7 6 7 18 8
```

```
Rad 3: 12 12 8 15 9 19 21
```

```
Rad 4: 89 14 9 17 18 19 27
```

Hela koden:

```
public class Test2 {

    public static void print(int[][] x) {
        System.out.println();
        for (int i=0; i<x.length; i++) {
            System.out.print("Rad " + i + ": ");
            for (int j=0; j<x[i].length; j++) {
                System.out.print(x[i][j] + " ");
            }
            System.out.println();
        }
    }

    public static void columnSort(int[][] a) {
        for (int k=0; k<a[0].length; k++) {
            for (int j=0; j<a.length-1; j++) {
                for (int i=0; i<a.length-1; i++) {
                    if (a[i][k] > a[i+1][k]) {
                        int tmp0 = a[i][k];
                        int tmp1 = a[i+1][k];
                        a[i][k] = tmp1;
                        a[i+1][k] = tmp0;
                    }
                }
            }
        }
    }

    public static void main(String[] args) {

        int[][] a = {
            { 12, 6, 7, 17, 18, 19, 8 },
            { 0, 14, 8, 15, 5, 3, 2 },
            { 7, 2, 1, 6, 9, 18, 21 },
            { 1, 5, 9, 3, 7, 11, 2 },
            { 89, 12, 6, 1, 0, 19, 27 }
        };
        columnSort(a);
        print(a);
    }
}
```

Kahoot!

Quiz

www.kahoot.it