# Leksjon 1

## Leksjon 1 - Lærebok

### Hei verden

/\*

\* Programmet skriver en hilsen med stjernelinjer

\*/

import static java.lang.System.\*;

public class StjerneHilsen {

public static void main(String[] args) {

out.println("\*\*\*\*\*\*\*\*\*\*\*");

out.println("Hei verden!");

out.println("\*\*\*\*\*\*\*\*\*\*\*");

}

}

### Hilsen med innlest navn

/\*

\* Programmet skriver ut en hilsen til et

\* innlest navn, både fornavn og etternavn.

\*/

import static javax.swing.JOptionPane.\*;

public class DialogTo {

public static void main(String[] args) {

String fornavn = showInputDialog("Skriv fornavnet ditt:");

String etternavn = showInputDialog("Skriv etternavnet ditt:");

showMessageDialog(null, "Hei " + fornavn + " " + etternavn + "!");

}

}

### Easygraphics – Tegner to sirkler

/\*

\* Programmet tegner ut to sirkler

\* forskjellige steder på skjermen.

\*/

public class Grafikk2 extends EasyGraphics {

public static void main(String[] args) {

launch(args);

}

public void run() {

makeWindow("To ulike sirkler");

drawCircle(150, 70, 60);

drawCircle(500, 500, 40);

}

}

### Easygraphics – Tegner et åttetall, to sirkler

/\*

\* Programmet tegner et åttetall vha. to sirkler

\*/

public class Aattetall extends EasyGraphics {

public static void main(String[] args) {

launch(args);

}

public void run() {

makeWindow ("Åttetall", 500, 400);

drawCircle (200, 120, 60);

drawCircle (200, 250, 70);

}

}

# Leksjon 2

## Leksjon 2 – Lærebok

### Lese inn to heltall, skrive ut sum

/\*

\* Programmet leser inn to heltall og

\* skriver ut summen av dem.

\*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

public class SumAvHeltall {

public static void main(String[] args) {

// Leser inn de to heltallene fra brukeren

String tall1Txt = showInputDialog("Gi første heltall:");

int tall1 = parseInt(tall1Txt);

String tall2Txt = showInputDialog("Gi andre heltall:");

int tall2 = parseInt(tall2Txt);

// Beregner summen av de to heltallene

int sum = tall1 + tall2;

showMessageDialog(null, "Sum: " + sum);

}

}

### Beregne arealet av et rektangel

/\*

\* Program for å beregne arealet av et rektangel.

\*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

public class RektangelAreal {

public static void main(String[] args) {

// Innlesing

String tallTxt = showInputDialog("Gi lengde:");

int lengde = parseInt(tallTxt);

tallTxt = showInputDialog("Gi bredde:");

int bredde = parseInt(tallTxt);

// Beregning

int areal = lengde \* bredde;

// Utskrift

showMessageDialog(null, "Areal = " + areal);

}

}

### Leser inn radius av sirkel, skriver ut arealet, avrundet heltall

/\*

\* Programmet leser inn radius i en sirkel og skriver

\* ut arealet av den avrundet til nærmeste heltall.

\*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Double.\*;

import static java.lang.Math.\*;

public class SirkelAreal {

public static void main(String[] args) {

// Leser inn radius

String radiusTxt = showInputDialog("Gi radius: r = ");

double radius = parseDouble(radiusTxt);

// Beregner arealet av sirkelen vha Math.PI

double areal = PI \* radius \* radius;

// Avrunder til nærmeste heltall vha Math.round

int helAreal = (int) round(areal);

showMessageDialog(null, "Radius: r = " + radius + "\n"

+ "Areal: " + helAreal);

}

}

### USD til NOK

/\*

\* Programmet leser inn et beløp i amerikanske dollar,

\* og skriver ut tilsvarende beløp i norske kroner.

\*/

import static java.lang.Double.\*;

import static javax.swing.JOptionPane.\*;

public class ValutaOmregning {

public static void main(String[] args) {

double kurs = 5.88; // dagens kurs, dollar mot krone

String tallTxt = showInputDialog("Gi dollarbeløp:");

double dollar = parseDouble(tallTxt);

double kroner = dollar \* kurs;

String ut = "Vekslebeløp: " + dollar + " USD" + "\n" +

"Dagens kurs: " + kurs + " NKR/USD" + "\n" +

"Utbetaling: " + kroner + " NKR";

showMessageDialog(null, ut);

}

}

### Leser inn heltall, skriver ut TVERRSUMMEN

/\*

\* Programmet leser inn et heltall mellom 0 og 1000,

\* og skriver ut summen av sifrene i tallet (tverrsummen).

\*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

public class Tverrsum {

public static void main(String[] args) {

String tallTxt = showInputDialog("Gi heltall (1-999):");

int tall = parseInt(tallTxt);

// Tar kopi av innlest heltall til bruk i utskriften.

// Feil rettet etter innspill fra Roy Christian Wego, 2015.

int heltall = tall;

// Tenk deg at tall = "s1 s2 s3" (tre siffer i rekkefølge)

int s3 = tall % 10; // finner bakerste siffer

tall = tall / 10; // tar bort bakerste siffer fra tall

int s2 = tall % 10; // finner bakerste siffer i redusert tall

// som er det midterste i det opprinnelige

int s1 = tall / 10; // finner første siffer i redusert tall

// som også er første siffer i det opprinnelige

int tverrsum = s1 + s2 + s3;

// Viser både sifre og tverrsum i utskriften

String ut = "Tallet " + heltall + " har tverrsummen: " +

s1 + "+" + s2 + "+" + s3 + "=" + tverrsum;

showMessageDialog(null, ut);

}

}

### Leser inn 4 heltall, skriver dem ut, finner det minste tallet.

/\*

\* Programmet leser inn fire heltall og skriver dem ut igjen

\* sammen med det minste av dem funnet vha. Math.min-metoden.

\*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

public class MinstAvFire {

public static void main(String[] args) {

// Setter sammen metodekall for dialogvindu og tallkverning

int tall1 = parseInt( showInputDialog("Gi heltall nr. 1:") );

int tall2 = parseInt( showInputDialog("Gi heltall nr. 2:") );

int tall3 = parseInt( showInputDialog("Gi heltall nr. 3:") );

int tall4 = parseInt( showInputDialog("Gi heltall nr. 4:") );

// I Java 8 så finnes metoden min både i Integer og i Math, og

// det oppstår dermed en tvetydighet som ikke var der i Java 7.

//

// Vi må dermed skrive på klassenavnet foran: Math.min

// Slike metodekall blir omtalt på side 61 og forklart på side

// 158-159 i læreboken. Se også kommentar nederst i denne filen.

int minst1 = Math.min(tall1, tall2);

int minst2 = Math.min(tall3, tall4);

int allerMinste = Math.min(minst1, minst2);

// Alternativt oppsett uten bruk av minst1 og minst2, kun:

// int allerMinste =

// Math.min(Math.min(tall1, tall2), Math.min(tall3, tall4));

String utTxt = "Innleste tall: " + tall1 + ", " + tall2 + ", "

+ tall3 + ", " + tall4 + "\n"

+ "Det minste tallet er: " + allerMinste;

showMessageDialog(null, utTxt);

}

}

// Statisk import gjør at vi f.eks. kan skrive parseInt i stedet for

// Integer.parseInt. Det kan være behagelig i et nybegynnerkurs.

// Desverre kan statisk import også føre til såkalte navnekollisjoner,

// ved at samme metodenavn er deklarert i flere klasser, som metoden

// min i denne oppgaven. Ved utvikling av større program vil man være

// forsiktig med bruk av statisk import, som kommentert øverst på

// side 195 i læreboken.

### Beregner mulige rekker på en tippekupong

/\*

\* Programmet beregner antall mulige rekker på en tippekupong.

\* Antall kamper på kupongen blir lest inn fra bruker

\* (ikke alltid 12 kamper).

\*/

import static java.lang.Integer.\*;

import static java.lang.Math.\*;

import static javax.swing.JOptionPane.\*;

public class Tippekupong {

public static void main(String[] args) {

String antTxt = showInputDialog("Antall kamper:");

int antKamper = parseInt(antTxt);

int antRekker = (int) pow(3, antKamper);

String ut = "Antall rekker med " + antKamper +

" kamper: " + antRekker;

showMessageDialog(null, ut);

}

}

### Bytter verdien i tre variabler

/\*

\* Programmet bytter om verdien til tre variabler x, y og z

\* på en "sirkulær" måte, dvs. at y får startverdien til x,

\* z får startverdien til y, og x får startverdien til z.

\*/

import static javax.swing.JOptionPane.\*;

public class Bytt {

public static void main(String[] args) {

// Gir variablene "tilfeldige" startverdier

// (kan evt. endres til å lese inn fra bruker)

int x = 25;

int y = 7;

int z = 12;

String ut = "Før: x = " + x + ", y = " + y + ", z = " + z + "\n";

int tmp = z; // x=25, y= 7, z=12, tmp=12

z = y; // x=25, y= 7, z= 7, tmp=12

y = x; // x=25, y=25, z= 7, tmp=12

x = tmp; // x=12, y=25, z= 7, tmp=12

ut += "Etter: x = " + x + ", y = " + y + ", z = " + z;

showMessageDialog(null, ut);

}

}

### Leser inn tre heltall for dag, mnd og år. Skriver ut på datoformat.

/\*

\* Programmet leser inn tre heltall for hhv. dag, mnd og år.

\* Skriver tallene ut på vanlig datoformat.

\*/

import static javax.swing.JOptionPane.\*;

import static java.lang.System.\*;

import static java.lang.Integer.\*;

import static java.lang.Math.\*;

public class SkrivDato {

public static void main(String[] args) {

// Setter sammen dialogvindu og tallkverning

int dag = parseInt( showInputDialog("Gi dagnr.:") );

int mnd = parseInt( showInputDialog("Gi mndnr.:") );

int år = parseInt( showInputDialog("Gi år:") );

String utTxt = "Dato: " + dag + "." + mnd + "." + år;

showMessageDialog(null, utTxt);

}

}

### Pris som skal betales, og hvordan i 10 og 1kr uten bruk av modul(restoperatoren %)

/\*

\* Programmet setter en pris som skal betales og et

\* beløp som er betalt inn. Det beregnes hvor mye som skal

\* betales tilbake og hvordan i 10-kr og 1-kr, uten bruk

\* av restoperatoren (%).

\*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

public class AlternativRest {

public static void main(String[] args) {

int pris = 124;

int beløp = 200; // Disse to bør leses inn!

// Beregner tilbakebetalingen

int veksel = beløp - pris;

// Utbetaling av veksel

int ant10kr = veksel / 10;

int ant1kr = veksel - 10\*ant10kr;

showMessageDialog(null, "Pris å betale: kr " + pris + "\n"

+ "Innbetalt: kr " + beløp + "\n"

+ "Tilbakebetalt 10-kr: " + ant10kr + " stk" + "\n"

+ "Tilbakebetalt 1-kr: " + ant1kr + " stk");

}

}

### Easygraphics – Tegner det franske flagget.

/\*

\* Programmet tegner det franske flagget.

\*/

public class FranskFlagg extends EasyGraphics {

public static void main(String[] args) {

launch(args);

}

public void run() {

makeWindow("Det franske flagget");

setColor(0, 0, 255);

fillRectangle(50, 50, 100, 200);

setColor(255, 0, 0);

fillRectangle(250, 50, 100, 200);

setColor(0, 0, 0);

drawRectangle(50, 50, 300, 200);

}

}

### Easygraphics – Tegner et hode med hatt, øyne og strekmunn

/\*

\* Programmet tegner et hode med hatt, øyne og strekmunn.

\*/

public class MannMedHatt extends EasyGraphics {

public static void main(String[] args) {

launch(args);

}

public void run() {

makeWindow("Mann med hatt");

// Definerer midtpunktet (x,y) og radius r til hodet.

// Figuren kan skaleres og flyttes ved å endre startverdiene.

int x = 100;

int y = 160;

int r = 80;

// tegner hodet ved å fylle en gul sirkel

setColor(255, 255, 0);

fillCircle(x, y, r);

// tegner øyne og munn i svart

setColor(0, 0, 0);

fillCircle(x-r/5, y-r/4, r/15);

fillCircle(x+r/5, y-r/4, r/15);

drawLine(x-r/4, y+r/2, x+r/4, y+r/2);

// tegner hatten ved å fylle to svarte rektangler

fillRectangle(x-r, y-3\*r/4, 2\*r, r/3);

fillRectangle(x-5\*r/8, y-3\*r/2, 5\*r/4, r);

}

}

### Easygraphics – tegner et spillbrett for «bondesjakk»

/\*

\* Programmet tegner et spillbrett for "bondesjakk".

\*/

public class Spillbrett extends EasyGraphics {

public static void main(String[] args) {

launch(args);

}

public void run() {

makeWindow("Bondesjakk");

int venstre = 100; // Øverste venstre hjørne

int topp = 100; // er (x=100, y=100)

int rute = 200; // Sidelengden til en rute

int rute2 = rute\*2;

int rute3 = rute\*3;

// Tegner vertikale linjer

drawLine(venstre, topp, venstre, topp+rute3);

drawLine(venstre+rute, topp, venstre+rute, topp+rute3);

drawLine(venstre+rute2, topp, venstre+rute2, topp+rute3);

drawLine(venstre+rute3, topp, venstre+rute3, topp+rute3);

// Tegner horisontale linjer

drawLine(venstre, topp, venstre+rute3, topp);

drawLine(venstre, topp+rute, venstre+rute3, topp+rute);

drawLine(venstre, topp+rute2, venstre+rute3, topp+rute2);

drawLine(venstre, topp+rute3, venstre+rute3, topp+rute3);

// Dette kan alternativt gjøres ved å tegne 3\*3 kvadrater.

}

}

### Beregner den beste måten å veksle et beløp i 10 og 1kr.

/\*

\* Programmet beregner den "beste" måten å veksle et

\* beløp i 10-kroner og 1-kroner.

\*/

import static javax.swing.JOptionPane.\*;

public class Veksel {

public static void main(String[] args) {

// Gir sum (prisen å betale) og betalt beløp

// "tilfeldige" verdier (kan evt. lese inn fra bruker)

int sum = 124;

int beløp = 200;

int veksel = beløp - sum ; // veksel er 76

int ant10kr = veksel / 10; // ant10kr er 7

int ant1kr = veksel - 10\*ant10kr; // ant1kr er 6

String ut = veksel + " kr kan veksles i " +

ant10kr + " 10-kroner og " +

ant1kr + " 1-kroner";

showMessageDialog(null, ut);

}

}

### Bytter verdien i tre variabler uten bruk av hjelpevariabel

/\*

\* Programmet bytter om verdien til tre variabler x, y og z

\* på en "sirkulær" måte, dvs. at y får startverdien til x,

\* z får startverdien til y, og x får startverdien til z.

\*

\* Dette gjøres uten bruk av hjelpevariabler.

\*/

import static javax.swing.JOptionPane.\*;

public class ByttUtenHjelpevariabler {

public static void main(String[] args) {

// Gir variablene "tilfeldige" startverdier

// (kan evt. endres til å lese inn fra bruker)

int x = 25;

int y = 7;

int z = 12;

String ut = "Før: x = " + x + ", y = " + y + ", z = " + z + "\n";

// x=25, y=7, z=12

z = z + y; // x=25, y= 7, z=19

y = z - y; // x=25, y=12, z=19

z = z - y; // x=25, y=12, z= 7

// Nå har y og z byttet verdier

x = x + y; // x=37, y=12, z= 7

y = x - y; // x=37, y=25, z= 7

x = x - y; // x=12, y=25, z= 7

ut += "Etter: x = " + x + ", y = " + y + ", z = " + z;

showMessageDialog(null, ut);

}

}

### Leser inn Unicode kodenr, konverterer til tegn(char) og tilbake.

/\*

\* Programmet er en utvidelse av Unicode-eksemplet

\* på side 59, der et innlest kodenummer (int) blir

\* konvertert til tegn (char). Nå blir dette tegnet

\* konvertert tilbake til kodenummeret.

\*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

public class UnicodeMotsatt {

public static void main(String[] args) {

String melding = "Kodenummer: ";

String kodeTxt = showInputDialog(melding);

int kodeNr = parseInt(kodeTxt);

char tegn = (char) kodeNr;

melding = melding + kodeNr + " => " + tegn;

// Utvidelse: Automatisk typetvang 'ned' til

// den mindre datatypen char

int tegnNr = tegn;

melding = melding + "\n"

+ "Enkelttegn " + tegn + " => " + tegnNr;

showMessageDialog(null, melding);

}

}

### Easygraphics – Brukerdefinert størrelse på fransk flagg

/\*

\* Programmet tegner det franske flagget. Bruker

\* kan skalere flagget ved å skrive inn en prosentsats.

\* 100 (prosent) gir "vanlig" størrelse, 50 halv størrelse,

\* 200 dobbel størrelse.

\*/

import static java.lang.Integer.\*;

public class FranskFlaggMedSkalering extends EasyGraphics {

public static void main(String[] args) {

launch(args);

}

public void run() {

makeWindow("Det franske flagget");

String tallTxt = getText("Skalering i prosent:");

int str = parseInt(tallTxt);

int vanligBredde = 300, vanligHøyde = 200;

int bredde = vanligBredde\*str/100;

int høyde = vanligHøyde\*str/100;

setColor(0, 0, 255);

fillRectangle(50, 50, bredde/3, høyde);

setColor(255, 0, 0);

fillRectangle(50+2\*bredde/3, 50, bredde/3, høyde);

setColor(0, 0, 0);

drawRectangle(50, 50, bredde, høyde);

}

}

### Easygraphics – Tegner en smiley

/\*

\* Programmet tegner en "smiley".

\*/

public class Smiley extends EasyGraphics {

public static void main(String[] args) {

launch(args);

}

public void run() {

makeWindow("Smiley");

// Definerer midtpunktet (x,y) og radius r til hodet.

// Figuren kan skaleres og flyttes ved å endre startverdiene.

int x = 100;

int y = 160;

int r = 80;

// tegner hodet ved å fylle en gul sirkel

setColor(255, 255, 0);

fillCircle(x, y, r);

// tegner øyne og munn i svart

setColor(0, 0, 0);

fillEllipse(x-r/5, y-r/4, r/20, r/10);

fillEllipse(x+r/5, y-r/4, r/20, r/10);

drawArc(x, y+r/4, 2\*r/5, 2\*r/5, 220, 100);

}

}

### Easygraphics – Tegner/Presenterer temperaturmåling for en rekke dager.

/\*

\* Programmet presenterer temperaturmålinger for dagene i en uke på

\* et gitt (tenkt) sted i et søylediagram.

\*

\* Det vil si, løsningsforslaget viser kun søyler for mandag-onsdag.

\* Fortsett gjerne på egen hånd fram til søndag - så vil du oppdage

\* at du må skrive nesten samme kode flere ganger. Etter at du har lært

\* om gjentak (løkker) i kapittel 3 og tabeller i kapittel 5, kan du vende

\* tilbake til denne oppgaven og løse den på en langt mer elegant måte!

\*

\*/

public class SoyleDiagram extends EasyGraphics {

public static void main(String[] args) {

launch(args);

}

public void run() {

// Nederste venstre hjørne til venstre søyle

int x = 50;

int y = 700;

// Bredden til en søyle

int bredde = 50;

// Luft mellom søylene

int luft = bredde/2;

// Omregningsfaktor fra målt temperatur til piksler

int faktor = 50;

makeWindow("Søylediagram");

setFont("Arial", 24);

// Vis søyle for mandag

String ukedag = "man";

int temp = 5; // Målt temperatur

int høyde = temp\*faktor; // Regner om til piksler

setColor(160, 160, 160); // Grå fyllfarge

fillRectangle(x, y-høyde, bredde, høyde);

setColor(0, 0, 0); // Sort til omriss og tekst

drawRectangle(x, y-høyde, bredde, høyde);

drawString(ukedag, x, y+luft);

drawString(temp+"", x+bredde/4, y-høyde-luft);

// Vis søyle for tirsdag

ukedag = "tir";

temp = 5;

høyde = temp\*faktor; // Må regne ut på nytt for denne dagen

x = x + bredde + luft; // Flytt til høyre

setColor(160, 160, 160);

fillRectangle(x, y-høyde, bredde, høyde);

setColor(0, 0, 0);

drawRectangle(x, y-høyde, bredde, høyde);

drawString(ukedag, x, y+luft);

drawString(temp+"", x+bredde/4, y-høyde-luft);

// Vis søyle for onsdag

ukedag = "ons";

temp = 7;

høyde = temp\*faktor;

x = x + bredde + luft;

setColor(160, 160, 160);

fillRectangle(x, y-høyde, bredde, høyde);

setColor(0, 0, 0);

drawRectangle(x, y-høyde, bredde, høyde);

drawString(ukedag, x, y+luft);

drawString(temp+"", x+bredde/4, y-høyde-luft);

// Og så videre for torsdag til søndag.

// Men husk: Ved hjelp av løkker (kap. 3) og tabeller (kap. 5) kan

// denne oppgaven løses mer elegant og med adskillig færre kodelinjer!

}

}

## Leksjon 2 – Roy

### Innlesing, lagring i variabler og utskrift i dialogvindu

// Leksjon 2: Jobb 1

// Utviklet av RMI - Aug 2017

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Programmet tester innlesing fra brukeren,

// lagring i variabler og utskrift i et

// dialogvindu.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import static java.lang.Integer.\*;

import static javax.swing.JOptionPane.\*;

public class Variabeltest {

public static void main(String[] args) {

// Deklarerer String-variabler

String fornavn, etternavn;

// Initierer en String-variabel

String melding = "Lykke til med Java!";

// Deklarerer int-variabel

int årstall;

// Innlesing av både for- og etternavn

fornavn = showInputDialog("Gi inn fornavnet ditt:");

etternavn = showInputDialog("Gi inn etternavnet ditt:");

// OBS!! Innlesing av heltallsverdier i 2 steg

String tallTekst = showInputDialog("Gi inn årstall:");

årstall = parseInt(tallTekst);

// Utskrift av info

showMessageDialog(null,"Fornavn:" + fornavn + "\n"

+ "Etternavn:" + etternavn + "\n"

+ "Årstall: " + årstall + "\n"

+ melding);

}

}

### Leser inn heltall, beregner summen(pluss og gange) skriver ut.

// Leksjon 2: Jobb 2

// Utviklet av RMI - Aug 2017

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Programmet leser inn et heltall, beregner

// summen av tallet med seg selv, og samme

// produkt. Skriver ut resultatene.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

public class Kalkulator {

public static void main(String[] args) {

// Innlesing

String tallTekst = showInputDialog("Gi et heltall: ");

// Konverterer fra String til heltall:

int tall = parseInt(tallTekst);

// Beregning

int svarAdd = tall + tall; // Sum, addisjon +

int svarMult = tall \* tall; // Produkt

// Utskrift

SshowMessageDialog(null, tall + " + " + tall + " = " + svarAdd + "\n"

+ tall + " \* " + tall + " = " + svarMult);

}

}

### Leser inn lengde og bredde for et rektangel, beregner og skriver ut omkrets

/\* Leksjon 2: Jobb 3

\* Utviklet av RMI - August 2017

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Programmet leser inn lengde og bredde for

\* et rektangel, med en grafisk veiledning,

\* beregner og skriver ut omkretsen.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

public class Omkretsberegning {

public static void main(String[] args) {

// Innlesing...

String ledetekst = "Beregning av omkrets:" + "\n"

+ "\*\*\*\*\*\*\*\*\*\*\*" + "\n"

+ "| | Høyde" + "\n"

+ "\*\*\*\*\*\*\*\*\*\*\*" + "\n"

+ " Bredde "+ "\n" + "\n";

String innTxt = showInputDialog(ledetekst + "Gi bredde: ");

int lengde = parseInt(innTxt);

// NB! Gjenbruker allerede deklarert variabel 'innTxt'

innTxt = showInputDialog(ledetekst + "Gi høyde: ");

int bredde = parseInt(innTxt);

// Beregning...

int omkrets = lengde\*2 + bredde\*2;

// Skriver ut resultat...

showMessageDialog(null, "Rektangel med " + "\n"

+ "Bredde: " + lengde + "\n"

+ "Høyde: " + bredde + "\n"

+ "Omkrets: " + omkrets);

}

}

### Leser lengde i tommer, omregner til meter og skriver ut

/\* Leksjon 2: Jobb 4

\* Utviklet av RMI - August 2017

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Programmet leser en lengde gitt i tommer,

\* omregner lengden til meter og skriver ut

\* en kvittering med inndata og resultat.

\* Obs! Programmet bruker en konstant.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

import static java.lang.Integer.\*;

import static javax.swing.JOptionPane.\*;

public class Lengdeomregning {

public static void main(String[] args) {

// Deklarerer en konstant

final double CM\_PR\_TOMME = 2.54;

// Innlesing av lengde i tommer

String tallTekst = showInputDialog("Gi lengde i antall tommer:");

int tommeTall = parseInt(tallTekst);

// Omregning: tommer -> meter

double cmTall = CM\_PR\_TOMME \* tommeTall;

double meterTall = cmTall / 100; // Deletegnet: shift 7

// Utskrift

String utTekst = "Innlest antall tommer: " + tommeTall + "\n"

+ "Tilsvarende lengde i meter: " + meterTall;

showMessageDialog(null, utTekst);

}

}

### Leser inn antall myntenheter, skriver ut i tillegg til kroneverdi(sum)

/\* Leksjon 2: Jobb nr 5

\* RMI - August 2017

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Programmet leser inn et beløp via et antall

\* 20-kr, antall 10-kr, antall 5-kr og antall

\* 1-kr. Beregner og skriver ut kroneverdien

\* til dette beløpet sammen med inndata.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

import static java.lang.System.\*;

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

public class Mynttelling {

public static void main(String[] args) {

// Innlesing av antall 20-kr

String ant20krTekst = showInputDialog("Gi ant. 20-kr:");

// Konvertering (tallkverning) fra tekst til heltall

int ant20kr = parseInt(ant20krTekst);

// Leser inn antall av de andre myntene på samme måte

String ant10krTekst = showInputDialog("Gi ant. 10-kr:");

int ant10kr = parseInt(ant10krTekst);

String ant5krTekst = showInputDialog("Gi ant. 5-kr:");

int ant5kr = parseInt(ant5krTekst);

String ant1krTekst = showInputDialog("Gi ant. 1-kr:");

int ant1kr = parseInt(ant1krTekst);

// Beregning av samlet kroneverdi i myntene

int verdi = ant20kr\*20 + ant10kr\*10 + ant5kr\*5 + ant1kr;

// Lager en oversiktlig utskrift (i konsollet)

String utTxt = "Myntveksling" + "\n"

+ "------------------" + "\n"

+ "Antall 20-kr: " + ant20kr + "\n"

+ "Antall 10-kr: " + ant10kr + "\n"

+ "Antall 5-kr: " + ant5kr + "\n"

+ "Antall 1-kr: " + ant1kr + "\n"

+ "------------------" + "\n"

+ "Samlet verdi: " + verdi + "\n"

+ "==================" + "\n";

out.println(utTxt);

// Endre selv: Skriv resultatet i et dialogvindu

}

}

### Leser inn beløp, beregner effektiv utbetaling i mynter

/\* Leksjon 2: Jobb nr 7

\* RMI - September 2017

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Programmet leser inn et beløp og beregner

\* hvordan dette best bør utbetales ved effektiv

\* bruk av myntene: 20-kr, 10-kr, 5-kr og 1-kr.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

import static java.lang.System.\*;

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

public class Mynttelling {

public static void main(String[] args) {

// Innlesing av beløp

String beløpTekst = showInputDialog("Gi beløp i hele kr:");

// Konvertering (tallkverning) fra tekst til heltall

int beløp = parseInt(beløpTekst);

// Beregning av antall ulike myntenheter

int ant20kr, ant10kr, ant5kr, ant1kr;

ant20kr = beløp / 20;

// Restkroner til utebetaling på øvrige mynter

int restKr = beløp % 20;

ant10kr = restKr / 10;

restKr = restKr % 10;

ant5kr = restKr / 5;

ant1kr = restKr % 5;

// Lager en oversiktlig utskrift (i konsollet)

String utTxt = "Beløp: " + beløp + " kr" + "\n"

+ "Utbetales ved" + "\n"

+ "Antall 20-kr: " + ant20kr + "\n"

+ "Antall 10-kr: " + ant10kr + "\n"

+ "Antall 5-kr: " + ant5kr + "\n"

+ "Antall 1-kr: " + ant1kr;

out.println(utTxt);

// Endre selv: Skriv resultatet i et dialogvindu

}

}

### Easygraphics – Fransk flagg

/\* Leksjon 2: Eksempel på grafikk

\* RMI - September 2017

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Programmet tegner det franske flagget, tre

\* stolper i fargene blått, hvitt og rødt.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*

\* NB! Se oppgave 17a) for utvide programmet

\* til å skalere flagget til et større, eller

\* mindre flagg.

\*/

public class FranskFlagg extends EasyGraphics {

public static void main(String[] args) {

launch(args);

}

public void run() {

makeWindow("Fransk flagg", 500, 500);

// Tegner tre stolper, like høye, like brede

setColor(0,0,255);

fillRectangle(50, 50, 100, 200);

setColor(255,255,255);

fillRectangle(150, 50, 100, 200);

setColor(255,0,0);

fillRectangle(250, 50, 100, 200);

setColor(0,0,0);

drawRectangle(50, 50, 300, 200);

}

}

# Leksjon 3

## Leksjon 3 – Lærebok

### Leser to tall, deler disse om mulig, hvis ikke feilmelding

/\*

\* Leser to tall a og b, og skriver ut resultatet

\* av a/b om dette er mulig, ellers feilmelding.

\*/

import static java.lang.Integer.\*;

import static javax.swing.JOptionPane.\*;

public class Divisjon {

public static void main(String[] args) {

String tallTxt = showInputDialog("Gi tall nr 1: a = ");

int a = parseInt(tallTxt);

tallTxt = showInputDialog("Gi tall nr 2: b = ");

int b = parseInt(tallTxt);

String ut = "Resultat: " + a + "/" + b;

if ( b != 0 )

ut += " = " + a/b;

else

ut += " er umulig!";

showMessageDialog(null, ut);

}

}

### Leser inn tre tall, skriver ut i stigende rekkefølge

/\*

\* Leser tre tall og skriver dem ut sortert i stigende rekkefølge.

\*/

import static java.lang.Integer.\*;

import static javax.swing.JOptionPane.\*;

public class SorterTreTall {

public static void main(String[] args) {

String tallTxt = showInputDialog("Gi tall nr 1: a = ");

int a = parseInt(tallTxt);

tallTxt = showInputDialog("Gi tall nr 2: b = ");

int b = parseInt(tallTxt);

tallTxt = showInputDialog("Gi tall nr 3: c = ");

int c = parseInt(tallTxt);

String ut = "Sorterte tall: ";

if ( a < b )

if ( b < c )

ut += a + " " + b + " " + c;

else // da vet vi at c <= b

if ( a < c )

ut += a + " " + c + " " + b;

else

ut += c + " " + a + " " + b;

else // da vet vi at b <= a

if ( a < c )

ut += b + " " + a + " " + c;

else // da vet vi at c <= a

if ( b < c )

ut += b + " " + c + " " + a;

else

ut += c + " " + b + " " + a;

showMessageDialog(null, ut);

}

}

### Leser inn 5-sifret personnummer, skriver ut hvilket kjønn (midtsiffer)

/\*

\* Leser 5-sifret personnummer og skriver

\* ut hvilket kjønn dette hører til.

\*/

import static java.lang.Integer.\*;

import static javax.swing.JOptionPane.\*;

public class PersonNr {

public static void main(String[] args) {

String tallTxt = showInputDialog("Gi personnummer (5 siffer):");

int nr = parseInt(tallTxt);

String ut = "Personnummer: " + nr + " er fra en ";

int midtsiffer = (nr / 100) % 10;

if ( midtsiffer % 2 == 0 ) // et partall?

ut += "kvinne";

else

ut += "mann";

showMessageDialog(null, ut);

}

}

### Leser inn månedsnummer, skriver ut navnet på måneden (Bruker case)

/ \* Leser et månedsnummer og skriver ut

\* navnet på måneden dette tilhører. \*/

import static java.lang.Integer.\*;

import static javax.swing.JOptionPane.\*;

public class Maanedsnavn {

public static void main(String[] args) {

String tallTxt = showInputDialog("Gi månedsnummer (1-12):");

int mndNr = parseInt(tallTxt);

String ut = "Månedsnummer " + mndNr + " er ";

switch (mndNr) {

case 1 : ut += "januar"; break;

case 2 : ut += "februar"; break;

case 3 : ut += "mars"; break;

case 4 : ut += "april"; break;

case 5 : ut += "mai"; break;

case 6 : ut += "juni"; break;

case 7 : ut += "juli"; break;

case 8 : ut += "august"; break;

case 9 : ut += "september"; break;

case 10: ut += "oktober"; break;

case 11: ut += "november"; break;

case 12: ut += "desember"; break;

default: ut += "er ugyldig!";

}

showMessageDialog(null, ut);

}

}

### Skriver ut kvadrattallene i området 0-100

/ \* Programmet skriver ut kvadrattallene i området 0-100.

\* Du bør selv utvide programmet slik at det leser inn

\* øvre grense for området, dvs. ikke bare 100! \*/

import static javax.swing.JOptionPane.\*;

public class KvadratTall {

public static void main(String[] args) {

String ut = "Kvadrattall: ";

for (int tall=1; tall<=10; tall++) {

ut += tall\*tall + " ";

}

showMessageDialog(null, ut);

}

}

### Skriver ut ASCII-tabellen

/\*

\* Programmet skriver ut ASCII-tabellen (for alle tallkodene 1–128).

\*/

import static java.lang.Integer.\*;

import static java.lang.System.\*;

public class Unicode {

public static void main ( String [] args ) {

for (int kodeNr=1; kodeNr<=128; kodeNr++) {

char tegn = (char) kodeNr;

out.println(kodeNr + " = " + tegn);

}

}

}

### Beregner kinopris basert på alder (Bruker if og else)

import static javax.swing.JOptionPane.\*;

import static java.lang.Double.\*;

import static java.lang.Integer.\*;

import static java.lang.String.\*;

public class KinoprisVerdisjekk {

public static void main(String[] args) {

final double MIN\_PRIS = 10;

final double MAX\_PRIS = 150;

String prisTxt = showInputDialog("Fullpris:");

double pris = parseDouble(prisTxt);

if ( pris < MIN\_PRIS || pris > MAX\_PRIS )

showMessageDialog(null, "Ulovlig pris!");

else { // Prisen er altså ok

String alderTxt = showInputDialog("Alder:");

int alder = parseInt(alderTxt);

if ( alder < 1 || alder > 110 )

showMessageDialog(null, "Alderen må være feil!");

else { // Alderen er altså ok

if ( alder <= 5 ) // Gratis

pris = 0.0;

else if ( alder < 12 || alder >= 60 ) // Halv pris

pris = pris/2.0;

showMessageDialog(null, "Pris: " + pris);

} // slutt alder

} // slutt pris

} // slutt main

}

### Skriver ut den lille gangetabellen

/\*

\* Programmet skriver ut den lille gangetabellen

\* (uten kolonne- og rad-overskrifter).

\*/

import static java.lang.Integer.\*;

import static java.lang.System.\*;

import static javax.swing.JOptionPane.\*;

public class Gangetabell {

public static void main ( String [] args ) {

for(int x=1; x <=10; x++) {

for (int y=1; y <=10; y++) {

out.print(x\*y + " ");

}

out.println();

}

}

}

### Leser inn antall linjer, og tegner en rettvinklet trekant-figur (I konsoll, med ‘+’ tegn)

/ \* Leser inn antall linjer og tegner

\* en rettvinklet trekant-figur. \*/

import static java.lang.Integer.\*;

import static java.lang.System.\*;

import static javax.swing.JOptionPane.\*;

public class TegnTrekant {

public static void main(String[] args) {

int antall;

// Leser inn en lovlig verdi i området 1 - 10

do {

String tallTxt = showInputDialog("Gi antall linjer (1 - 10):");

antall = parseInt(tallTxt);

} while ( antall < 1 || antall > 10 );

// Nøstet for-setning for å tegne linjene lengre og lengre

for (int linjeNr = 1; linjeNr <= antall; linjeNr++) {

for (int tegnNr = 1; tegnNr <= linjeNr; tegnNr++)

out.print("+ ");

out.println();

}

}

}

### Leser inn antall linjer og tegner en rettvinklet L-figur (I konsoll, med ‘+’ tegn)

/\*

\* Leser inn antall linjer og tegner ut

\* en rettvinklet figur, en slags L.

\*/

import static java.lang.Integer.\*;

import static java.lang.System.\*;

import static javax.swing.JOptionPane.\*;

public class TegnVinkel {

public static void main(String[] args) {

int antall;

// Leser inn en lovlig verdi i området 1 - 10

do {

String tallTxt = showInputDialog("Gi antall linjer (1 - 10):");

antall = parseInt(tallTxt);

if ( antall < 1 || antall > 10 )

showMessageDialog(null, "Ulovlig antall linjer!");

} while ( antall < 1 || antall > 10 );

// Første for-setning tegner vertikal linje nedover, men

// stopper hakket før horisontal linje starter.

for (int linjeNr = 1; linjeNr < antall; linjeNr++)

out.println("+");

// Andre for-setning tegner horisontal linje.

for (int tegnNr = 1; tegnNr <= antall; tegnNr++)

out.print("+ ");

out.println();

}

}

### Easygraphics – 10 tall, nedtelling (Fjerner også forrige tall)

/\*

\* Teller ned 10 tall, ved å vise en 'animasjon'

\* av ett og ett tall 9 --> 0

\*/

import static java.lang.Integer.\*;

import static java.lang.Math.\*;

public class Nedtelling extends EasyGraphics {

public static void main(String[] args) {

launch(args);

}

public void run() {

makeWindow("Nedtelling", 500, 500);

setFont("Arial", 240);

for (int nr = 9; nr >= 0; nr--) {

setColor(255,0,0);

// Stroe røde tall midt på tegneflaten

drawString(""+nr, 200, 300);

pause(1000); // Vent i ett sekund

setColor(255,255,255);

// Hvisker ut forrige tall ved å legge

// en hvit firkant oppå tegneflaten.

fillRectangle(0,0,500,500);

}

}

}

### Easygraphics – «Slide-show» Fast overskrift, rullerende tekst.

/\*

\* Programmet simulerer et "slide show" med ett enkelt lysbilde,

\* der overskriften blir vist med en gang, mens tre tekster i en

\* punktliste "flyr inn" etter hvert.

\*

\* Oppgaveteksten sier at de tre tekstene i punktlisten skal lagres

\* i tre String-variabler (linje 25-27). Koden for å "fly inn" hver

\* av de tre tekstene blir nesten helt lik, og du tenker kanskje at

\* det burde være mulig å ta i bruk løkker her? Da tenker du riktig,

\* men først må du lære om tabeller (kapittel 5).

\*

\*/

import static java.lang.Integer.\*;

import static java.lang.Math.\*;

public class Slideshow extends EasyGraphics {

public static void main(String[] args) {

launch(args);

}

public void run() {

makeWindow("Slideshow", 500, 500);

// Innholdet på lysarket

String overskrift = "Løkker i Java";

String punkt1 = "while-løkke";

String punkt2 = "do-løkke";

String punkt3 = "for-løkke";

// Sett stor font og vis overskrift

setFont("Arial", 48);

setColor(0, 0, 0);

int x=20;

int y=50;

drawString(overskrift, x, y);

// Skru ned font og endre farge

setFont("Arial", 28);

setColor(0, 0, 255);

// Utgangspunkt for punktlisten

x = x + 50;

y = 100;

// Langsom fart

setSpeed(1);

// Vis og "fly inn" første punkt

int id = drawString(punkt1, 0, 0);

moveString(id, x, y);

// 1 sekund pause

pause(1000);

// Posisjon til neste punkt

y = y+30;

// Vis og "fly inn" andre punkt

id = drawString(punkt2, 0, 0);

moveString(id, x, y);

// Og tilsvarende for punkt 3

pause(1000);

y = y+30;

id = drawString(punkt3, 0, 0);

moveString(id, x, y);

// Koden for å "fly inn" tekstene i punktlisten kan kodes

// mer elegant ved å bruke tabeller. Vend gjerne tilbake til

// denne oppgaven etter at du har gått gjennom kapittel 5.

}

}

### Leser inn antall tall, skriver ut antall Fibonacci-tall

/\*

\* Programmet skriver ut et antall Fibonacci-tall.

\* Bruker skriver inn antall tall.

\* Pr. def. er neste Fibonacci-tall: f = a + b

\* der a og b er de to foregående Fibonacci-tallene.

\*/

import static java.lang.Integer.\*;

import static javax.swing.JOptionPane.\*;

public class Fibonacci {

public static void main(String[] args) {

String tallTxt = showInputDialog("Antall Fibonaccitall: ");

int n = parseInt(tallTxt);

int a=1, b=1; // De to første Fibonacci-tallene.

String ut = "Fibonaccitall: " + a + " " + b + " ";

// Skriver alltid ut de to første tallene, men her bør du

// legge inn en if-setning som sjekker om n >= 3 som er

// det første tallet som virkelig må beregnes.

for (int tall=3; tall<=n; tall++) {

int f = a + b;

ut += f + " ";

b = a;

a = f;

}

showMessageDialog(null, ut);

}

}

### Easygraphics – Tegner en «murvegg»

/\*

\* Tegner en "murvegg" i et grafisk vindu.

\*/

public class Murvegg extends EasyGraphics {

public static void main(String[] args) {

launch(args);

}

public void run() {

makeWindow("Murvegg");

int xStart = 100, yStart = 350;

int xStr = 80, yStr = 50;

int bredde = 7, høyde = 5;

int x=xStart, y=yStart;

for (int rad=1; rad<=høyde; rad++) {

if ( rad%2 == 1 ) {

// Oddetallsrader starter med en halv-sten

drawRectangle(x, y, xStr/2, yStr);

x += xStr/2;

pause(180);

}

for (int stein=1; stein<=bredde; stein++) {

drawRectangle(x, y, xStr, yStr);

x += xStr;

pause(180);

}

if ( rad%2 == 0 ) {

// Partallsrader slutter med en halv-sten

drawRectangle(x, y, xStr/2, yStr);

pause(150);

}

x = xStart;

y -= yStr;

}

}

}

### Easygraphics – Tegner blinkskiver

import static java.lang.Integer.\*;

public class BlinkSkiver extends EasyGraphics {

public static void main(String[] args) {

launch(args);

}

public void run() {

makeWindow("BlinkSkiver");

// Leser antall ringer til lovlig verdi (2-10) er sikret

int antRinger = parseInt( getText("Antall ringer (2-10): ") );

if ( 2 <= antRinger && antRinger <= 10 ) {

int startRadius = 25;

int x = 400;

int y = 400;

boolean erHvit = true;

for (int i=antRinger; i>=1; i--) {

int radius = startRadius\*i;

if ( erHvit )

setColor(0,0,0);

// skifter til svart

else

setColor(255,255,255);

// skifter til hvit igjen

erHvit = !erHvit;

// Passer på aktuell tegnefarge

fillCircle (x, y, radius);

pause(300);

}

}

else

if ( antRinger <= 2 )

drawString("Ulovlig verdi: Antall ringer er for lavt!", 100, 100);

else

drawString("Ulovlig verdi: Antall ringer er for høyt!", 100, 100);

}

}

## Leksjon 3 – Roy

### Leser heltall, avgjør og skriver ut hvis mer enn tre siffer

/\* Leksjon 3: Jobb 1

\* Utviklet av RMI - September 2017

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Programmet leser et heltall, avgjør og

\* skriver ut om det har mer enn tre siffer. \*/

import static java.lang.Integer.\*;

import static javax.swing.JOptionPane.\*;

public class Sifferantall {

public static void main(String[] args) {

// Innlesing av heltall

String tallTekst = showInputDialog("Gi et heltall:");

int tall = parseInt(tallTekst);

// Avgjør om antall siffer overstiger 3

// Obs! Tar høyde for at tall kan være negativt

String utTekst = "Tallet " + tall + " har ";

if ( Math.abs(tall) >= 1000 )

utTekst += "flere enn 3 siffer";

else

utTekst += "ikke flere enn 3 siffer";

// Utskrift til dialogvindu

showMessageDialog(null,utTekst);

}

}

### Kalkulator, leser to heltall og en regneoperasjon, skriver ut.

/\* Leksjon 3: Jobb 2, pkt b - Kalkulator

\* Utviklet av RMI - Sep 2017

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Leser to heltall og en regneoperasjon. Skriver kun ut det

\* tilhørende svaret, enten sum, differans, produkt eller

\* brøk (hvis mulig).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Nyere Java-versjoner tillater String som datatype på

\* kontrollvariabelen i switch-setningen. Dette kan gjøre

\* sortering på regneoperasjon litt enklere.

\*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

public class Kalkulator {

public static void main(String[] args) {

// Innlesing

String innTekst = showInputDialog("Gi første tall: ");

int t1 = parseInt(innTekst);

// NB! Bruker korte (dårlige) variabelnavn av plasshensyn.

innTekst = showInputDialog("Gi andre tall: ");

int t2 = parseInt(innTekst);

// Ber om at brukeren oppgir symbol for ønsket regneoperasjon.

String op = showInputDialog("Velg regneoperasjon: (+ - \* /)");

// Beregner svar hvis lovlig operator, ellers feilmelding

int svar;

double brøk; // Divisjon kan gi desimalsvar

String utTekst="";

// op har datatypen String, verdier skrives mellom " "

switch (op) {

// NB! String kan være 'datatype' for tilslagsverdier i case

case "+": utTekst = t1+"+"+t2+"="+(t1+t2); break;

case "-": utTekst = t1+"-"+t2+"="+(t1-t2); break;

case "\*": utTekst = t1+"\*"+t2+"="+(t1\*t2); break;

case "/": utTekst = t1+"/"+t2+"=";

if (t2 != 0)

utTekst += ((double)t1/t2);

else // kan ikke dele på 0

utTekst += "Udefinert"; break;

default: utTekst = "Ulovlig regneoperasjon!";

}

// Utskrift

showMessageDialog(null, utTekst);

}

}

### Leser inn dagnummer, skriver ut dagnavnet (bruker switch --> case)

/\* Leksjon 3: Jobb 3

\* Utviklet av RMI - Sep 2017

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Leser inn et dagnummer (1-7), og skriver

\* ut det tilhørende dagnavnet.

\* NB! Testprogram for switch-setning

\*/

import static java.lang.Integer.\*;

import static javax.swing.JOptionPane.\*;

public class Dagnavn {

public static void main(String[] args) {

// Innlesing av dagnummer

String tallTekst = showInputDialog("Gi et dagnummer (1-7):");

int dagNr = parseInt(tallTekst);

// Beregningsdel, kobler dagnr og dagnavn

String dagnavn = ""; // Må initiere pga. valgsetningen

switch ( dagNr ) {

case 1: dagnavn = "mandag"; break;

case 2: dagnavn = "tirsdag"; break;

case 3: dagnavn = "onsdag"; break;

case 4: dagnavn = "torsdag"; break;

case 5: dagnavn = "fredag"; break;

case 6: dagnavn = "lørdag"; break;

case 7: dagnavn = "søndag"; break;

default:dagnavn = "ugyldig dag!";

}

// Utskrift til dialogvindu

String utTekst = "Dag nr " + dagNr + " er en " + dagnavn;

showMessageDialog(null,utTekst);

}

}

### Leser et heltall, plukker ut siste siffer, oppgir i tekstform.

/\* Leksjon 3: Jobb 4

\* Utviklet av RMI - Sep 2017

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Programmet leser et heltall, plukker ut siste

\* siffer og oppgir dette på tekstlig form.

\*/

import static java.lang.Integer.\*;

import static javax.swing.JOptionPane.\*;

public class Siffertekst {

public static void main(String[] args) {

// Innlesing av heltall

String tallTekst = showInputDialog("Gi et heltall:");

int tall = parseInt(tallTekst);

utTekst = "Siste i siffer i " + tall + " er ";

// Beregningsdel: Finner siste siffer og tekster dette

// I tilfelle tallet er negativt - uten import her.

tall = Math.abs(tall);

// Plukker ut bakerste siffer:

int siffer = tall % 10;

// Tekster sifferet via switch:

String utTekst += siffer + " -> ";

switch ( siffer ) {

case 9 : utTekst += "NI"; break;

case 8 : utTekst += "ÅTTE"; break;

case 7 : utTekst += "SJU"; break;

case 6 : utTekst += "SEKS"; break;

case 5 : utTekst += "FEM"; break;

case 4 : utTekst += "FIRE"; break;

case 3 : utTekst += "TRE"; break;

case 2 : utTekst += "TO"; break;

case 1 : utTekst += "EN"; break;

case 0 : utTekst += "NULL";

// Trenger ikke default her

}

// Utskrift til dialogvindu

showMessageDialog(null, utTekst);

}

}

### Skriver ut alle positive heltall mellom innlest nedre og øvre grense, sorterer odde/partall

/\* Leksjon 3: Jobb 5 (Tallserier av odde- og partall)

\* Utviklet av RMI - Sep 2017

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Programmet skriver ut alle positive heltall mellom innlest nedre og

\* øvre grense, ved å skrive alle oddetall på en linje og alle partall

\* på neste linje. Programmet kjøres om igjen helt til det blir gitt

\* beskjed om å avslutte.

\*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

public class OddeOgPartall {

public static void main(String[] args) {

String innTekst = showInputDialog("Gi nedre grense ( >= 1): ");

int nedreGrense = parseInt(innTekst);

innTekst = showInputDialog("Gi øvre grense ( >= " + nedreGrense + "):");

int øvreGrense = parseInt(innTekst);

String utTekst = "";

if ( nedreGrense < 1 || øvreGrense < nedreGrense )

utTekst = "Ulovlig grenseverdier!";

else {

// Lager to tekststrenger, en for hver linje i utskriften

String utOdde = "Oddetall: "; // Linje nr 1

String utPar = "Partall: "; // Linje nr 2

for (int tall=nedreGrense; tall<=øvreGrense; tall++)

if ( tall%2==1 )

utOdde += tall + " ";

else

utPar += tall + " ";

utTekst = utOdde + "\n" + utPar;

}

showMessageDialog(null, utTekst);

}

}

/\* Leksjon 3: Jobb 6 (Gjennomsnitt)

\* Utviklet av RMI - Sep 2017

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Leser inn ett og ett posistivt heltall (ved do-løkke), beregner summen,

\* finner minste og største tall, beregner gjennomsnittet og skriver ut de

\* aktuelle resultatene (avrundet til en desimal).

\*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

public class Gjennomsnitt {

public static void main(String[] args) {

// Deklarerer variabler som blir brukt i programmet

int antall = 0; // Ingen tall lest inn ennå

int sum = 0; // Ingen tall summert ennå

int min = MAX\_VALUE; // Alle tall er mindre enn dette

int max = 0; // Alle tall er større enn dette

int tall; // Innlest tall fra brukeren

do {

String tallTekst = showInputDialog("Gi et positivt heltall (stopp=-1):");

tall = parseInt(tallTekst);

if ( tall != -1 ) // Ikke stoppsignal enda vel . . .

if ( tall > 0 ) { // Feilskrift? Kun positive heltall skal behandles

antall++; // Øker antall innleste tall ett hakk

sum += tall; // Øker summen med det innleste tallet

if ( tall < min )

min = tall; // Er dette minste tall (så langt)?

if ( tall > max )

max = tall; // Er dette største tall (så langt)?

}

else

showMessageDialog(null, "Ulovlig inndata!");

} while ( tall != -1 );

String utTekst;

if ( antall == 0 )

utTekst = "Ingen tall er lest inn!";

else {

double snitt = (double)sum / antall;

snitt = (int)(snitt\*10 + 0.5) / 10.0; // Avrunder til én desimal

// Se side 57 i læreboka

utTekst = "Resultat" + "\n" + " Antall tall: " + antall + "\n"

+ "Gjennomsnitt: " + snitt + "\n"

+ " Minste tall: " + min + "\n"

+ "Største tall: " + max;

}

showMessageDialog(null, utTekst);

}

}

/\* Leksjon 3, Jobb 7: Terningkast

\* Utviklet av RMI - Sep 2017

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Simulerer kast med en terning, og teller antall kast til første 6-er.

\* Utvider til innlesing av et antall kastserier, og så beregnes et

\* gjennomsnittlig antall kast til første sekser. Dvs. hvilket antall

\* kast til første 6-er er det mest sannsynlig at det blir ?

\*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

public class Terningkast {

public static void main( String[] args ) {

int antKastserier = parseInt( showInputDialog("Ant. serier: ") );

int sumKast = 0;

for (int i=1; i<=antKastserier; i++) {

int antKast = 0;

int øyne;

do {

øyne = (int)(6\*Math.random() + 1);

antKast++;

} while ( øyne != 6);

sumKast += antKast;

}

// Beregner snittet – husk typeskift i divisjonen

double snitt = (double)sumKast/antKastserier;

// Avrunder snittet til en desimal (se side 57 i læreboka)

snitt = (int)(10\*snitt + 0.5)/10.0; // Kan bruke Math.round

String utTekst = "Snitt ant. kast til første 6-er: " + snitt + "\n"

+ "Antall kastserier: " + antKastserier;

showMessageDialog(null, utTekst);

}

}

///////////////////////////////////////////////////////////

Eksempel på grafikk-program (EasyGraphic)

///////////////////////////////////////////////////////////

/\* Leksjon 3: Fransk flagg

\* RMI - September 2017

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Programmet tegner det franske flagget, tre

\* stolper i fargene blått, hvitt og rødt.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*

\* Oppgave 17a) Skalere flagget til et større,

\* eller mindre flagg (innlest %-faktor).

\*/

import static java.lang.Integer.\*;

public class FranskFlaggV2 extends EasyGraphics {

public static void main(String[] args) {

launch(args);

}

public void run() {

// Må lage vinduet først

makeWindow("Fransk flagg", 800, 500);

// Startstørrelse på flagget (100%)

final int BREDDE = 300;

final int HØYDE = 200;

// Leser inn ønsket %-skalering fra brukeren

String tallText = getText( "Skalering i % (gi verdi 50-200):" );

int faktor = parseInt(tallText);

// VALGSETNING !

if ( 50 <= faktor && faktor <= 200 ) {

// Hvis lovlig skalering --> Tegne flagg

// Skalert bredde

int flaggbredde = BREDDE\*faktor/100;

// Skalert høyde

int flagghøyde = HØYDE\*faktor/100;

// Tegner tre stolper, like høye, like brede

int stolpe = flaggbredde/3;

// Blå stolpe først

setColor(0,0,255);

fillRectangle(50, 50, stolpe, flagghøyde);

// Hvit stolpe skjøvet en stolpe mot høyre

setColor(255,255,255);

fillRectangle(50+stolpe, 50, stolpe, flagghøyde);

// Rød stolpe skjøvet to stolper mot høyre

setColor(255,0,0);

fillRectangle(50+2\*stolpe, 50, stolpe, flagghøyde);

// Svart ramme rundt flagget

setColor(0,0,0);

drawRectangle(50, 50, flaggbredde, flagghøyde);

}

else { // Ulovlig skalering --> Feilmelding til brukeren

setFont("Arial",64);

drawString("Ulovlig skalering!", 100,100);

}

}

}

/\* Leksjon 3, oppgave 13

\* RMI - September 2017

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Tegner en "murvegg" i et grafisk vindu. Lager animasjon

\* ved å legge inn 'pause' mellom hver stein som tegnes ut.

\* Du kan endre på vindustørrelsen, og muren vil utnytte

\* den plassen som stilles til rådighet - etter beste evne.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* NB! Kortversjonen av denne løsningen finner du ute på

\* lærebokas nettsider.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

import static javax.swing.JOptionPane.\*;

public class Murvegg extends EasyGraphics {

public static void main(String[] args) {

launch(args);

}

public void run() {

// Setter vindustørrelse (minst 300x300)

int vinduX = 600, vinduY = 500;

int steinHøyde = 50; // Høyden på en stein

int antStein = 8; // Lengden på steinraden

// Setter avstand mellom muren og vinduskanten

int margX = 50, margY = 50;

// Beregner maksimal høyde (max rader) på muren

int maxHøyde = (vinduY - 2\*margY) / steinHøyde;

// Oppretter selve vinduet

makeWindow("Murvegg", vinduX,vinduY);

// Setter blågrå himmel i bakgrunnen

setBackground(0,150,200);

int murhøyde; // Brukes senere og kan ikke deklareres

// lokalt inne i do-løkka.

String melding = "Gi murhøyde (1 - "+maxHøyde+"):";

boolean ulovlig, gittFeilmelding = false;

do {

String murhøydeTxt = getText(melding);

murhøyde = Integer.parseInt( murhøydeTxt );

ulovlig = ( murhøyde < 1 || murhøyde > maxHøyde );

if ( ulovlig ) {

gittFeilmelding = true;

setFont("Arial", 50);

drawString("Gi en lovlig murhøyde!", margX, 2\*margY);

}

} while (ulovlig);

// Dekker over evt. feilmelding gitt i vinduet

if ( gittFeilmelding ) {

setColor(0,150,200);

fillRectangle(0,0,vinduX, vinduY);

}

// Beregner steinlengdene til å passe med ledig bredde

// Husk at hver eneste rad har en halv (0.5) stein

int steinLengde = (int)( (vinduX-2\*margX)/(antStein-0.5) );

// Setter starthjørne (x,y) på første stein nede til høyre

int x=margX, y=vinduY-margY-steinHøyde;

for (int radNr=1; radNr<=murhøyde; radNr++) {

// Hver rad har et antall steiner

for (int steinNr=1; steinNr<=antStein; steinNr++) {

int lengde = steinLengde;

// Steinene legges i forband, dvs. en rad starter

// med halvstein, den neste avsluttes med halvstein.

if ( steinNr == 1 && radNr % 2 == 1

|| steinNr == antStein && radNr % 2 == 0 )

lengde = steinLengde/2;

// To ulik farger på steinene

if ( (steinNr+radNr) % 2 == 1 )

setColor(240,235,235); // Mørk rødgrå

else

setColor(240,225,225); // Lys rødgrå

fillRectangle(x, y, lengde, steinHøyde);

// Trekker svart kontur rundt hver stein

setColor(0,0,0);

drawRectangle(x, y, lengde, steinHøyde);

x += lengde;

pause(100);

}

// Neste rad begynner i venstre kant igjen

x = margX;

// Neste rad skal en steinhøyde oppover

y -= steinHøyde;

}

}

// Send meg en epost om du har testet og gjerne

// eksperimentert med dette programmet.

}