Innhold

[1 Leksjon 1 6](#_Toc501041554)

[1.1 Leksjon 1 - Lærebok 6](#_Toc501041555)

[1.1.1 Hilsen med innlest navn 6](#_Toc501041556)

[1.1.2 Easygraphics – Tegner to sirkler 6](#_Toc501041557)

[1.1.3 Easygraphics – Tegner et åttetall, to sirkler 6](#_Toc501041558)

[2 Leksjon 2 7](#_Toc501041559)

[2.1 Leksjon 2 – Lærebok 7](#_Toc501041560)

[2.1.1 Lese inn to heltall, skrive ut sum 7](#_Toc501041561)

[2.1.2 Beregne arealet av et rektangel 7](#_Toc501041562)

[2.1.3 Leser inn radius av sirkel, skriver ut arealet, avrundet heltall 7](#_Toc501041563)

[2.1.4 USD til NOK 8](#_Toc501041564)

[2.1.5 Leser inn heltall, skriver ut TVERRSUMMEN 8](#_Toc501041565)

[2.1.6 Leser inn 4 heltall, skriver dem ut, finner det minste tallet. 9](#_Toc501041566)

[2.1.7 Beregner mulige rekker på en tippekupong 10](#_Toc501041567)

[2.1.8 Bytter verdien i tre variabler 10](#_Toc501041568)

[2.1.9 Leser inn tre heltall for dag, mnd og år. Skriver ut på datoformat. 11](#_Toc501041569)

[2.1.10 Pris som skal betales, og hvordan i 10 og 1kr uten bruk av modul(restoperatoren %) 11](#_Toc501041570)

[2.1.11 Easygraphics – Tegner et hode med hatt, øyne og strekmunn 12](#_Toc501041571)

[2.1.12 Easygraphics – tegner et spillbrett for «bondesjakk» 12](#_Toc501041572)

[2.1.13 Beregner den beste måten å veksle et beløp i 10 og 1kr. 13](#_Toc501041573)

[2.1.14 Bytter verdien i tre variabler uten bruk av hjelpevariabel 14](#_Toc501041574)

[2.1.15 Leser inn Unicode kodenr, konverterer til tegn(char) og tilbake. 14](#_Toc501041575)

[2.1.16 Easygraphics – Brukerdefinert størrelse på fransk flagg 15](#_Toc501041576)

[2.1.17 Easygraphics – Tegner en smiley 16](#_Toc501041577)

[2.1.18 Easygraphics – Tegner/Presenterer temperaturmåling for en rekke dager. 16](#_Toc501041578)

[2.2 Leksjon 2 – Roy 18](#_Toc501041579)

[2.2.1 Innlesing, lagring i variabler og utskrift i dialogvindu 18](#_Toc501041580)

[2.2.2 Leser inn heltall, beregner summen(pluss og gange) skriver ut. 18](#_Toc501041581)

[2.2.3 Leser inn lengde og bredde for et rektangel, beregner og skriver ut omkrets 19](#_Toc501041582)

[2.2.4 Leser lengde i tommer, omregner til meter og skriver ut 20](#_Toc501041583)

[2.2.5 Leser inn antall myntenheter, skriver ut i tillegg til kroneverdi(sum) 20](#_Toc501041584)

[2.2.6 Leser inn beløp, beregner effektiv utbetaling i mynter 21](#_Toc501041585)

[2.2.7 Easygraphics – Fransk flagg 22](#_Toc501041586)

[3 Leksjon 3 24](#_Toc501041587)

[3.1 Leksjon 3 – Lærebok 24](#_Toc501041588)

[3.1.1 Leser to tall, deler disse om mulig, hvis ikke feilmelding 24](#_Toc501041589)

[3.1.2 Leser inn tre tall, skriver ut i stigende rekkefølge 24](#_Toc501041590)

[3.1.3 Leser inn 5-sifret personnummer, skriver ut hvilket kjønn (midtsiffer) 25](#_Toc501041591)

[3.1.4 Leser inn månedsnummer, skriver ut navnet på måneden (Bruker case) 25](#_Toc501041592)

[3.1.5 Skriver ut kvadrattallene i området 0-100 26](#_Toc501041593)

[3.1.6 Beregner kinopris basert på alder (Bruker if og else) 26](#_Toc501041594)

[3.1.7 Skriver ut den lille gangetabellen 27](#_Toc501041595)

[3.1.8 Leser inn antall linjer, og tegner en rettvinklet trekant-figur (I konsoll, med ‘+’ tegn) 27](#_Toc501041596)

[3.1.9 Leser inn antall linjer og tegner en rettvinklet L-figur (I konsoll, med ‘+’ tegn) 28](#_Toc501041597)

[3.1.10 Easygraphics – 10 tall, nedtelling (Fjerner også forrige tall) 28](#_Toc501041598)

[3.1.11 Easygraphics – «Slide-show» Fast overskrift, rullerende tekst. 29](#_Toc501041599)

[3.1.12 Leser inn antall tall, skriver ut antall Fibonacci-tall 31](#_Toc501041600)

[3.1.13 Easygraphics – Tegner en «murvegg» 31](#_Toc501041601)

[3.1.14 Easygraphics – Tegner blinkskiver 32](#_Toc501041602)

[3.2 Leksjon 3 – Roy 33](#_Toc501041603)

[3.2.1 Leser heltall, avgjør og skriver ut hvis mer enn tre siffer 33](#_Toc501041604)

[3.2.2 Kalkulator, leser to heltall og en regneoperasjon, skriver ut. 33](#_Toc501041605)

[3.2.3 Leser inn dagnummer, skriver ut dagnavnet (bruker switch --> case) 34](#_Toc501041606)

[3.2.4 Leser et heltall, plukker ut siste siffer, oppgir i tekstform. 35](#_Toc501041607)

[3.2.5 Skriver ut alle positive heltall mellom innlest nedre og øvre grense, sorterer odde/partall 36](#_Toc501041608)

[3.2.6 Leser inn ett og ett heltall, finner sum, minste/største tall, gjennomsnitt og skriver ut avrundet til en desimal 37](#_Toc501041609)

[3.2.7 Leser inn ant terningkast, simulerer kast og teller ant kast til første 6er 38](#_Toc501041610)

[3.2.8 Easygraphics – Fransk flagg v.2 39](#_Toc501041611)

[3.2.9 Easygraphics – Tegner en murvegg, animerer med pause. 40](#_Toc501041612)

[4 Leksjon 4 - Metoder 43](#_Toc501041613)

[4.1 Leksjon 4 – Lærebok 43](#_Toc501041614)

[4.1.1 Bruker metode til å lage rektangel med 2 parametre 43](#_Toc501041615)

[4.1.2 Bruker metode til å lage trekant med en parameter 43](#_Toc501041616)

[4.1.3 Bruker metode til å skrive ut et pil-mønster 44](#_Toc501041617)

[4.1.4 Bruker metoder for omregning, en for lengder i cm og en i tommer. 44](#_Toc501041618)

[4.1.5 Bruker metode fakultet, beregner noe ved bruk av en løkke 45](#_Toc501041619)

[4.1.6 Leser inn et heltall, sjekker at dette ligger i et angitt invervall 45](#_Toc501041620)

[4.1.7 Leser inn et tegn, bruker metoder for å avgjøre om dette er en bokstav, stor eller liten, vokal eller konsonant. 46](#_Toc501041621)

[4.1.8 Beregner summen av en sekvens med påfølgende heltall 47](#_Toc501041622)

[4.1.9 Easygraphics – Tegner et antall lastebiler i en kolonne, lastebilene blir større lengre frem i kolonnen 47](#_Toc501041623)

[4.1.10 Easygraphics – Tegner ut en husrekke med takutstikk 48](#_Toc501041624)

[4.1.11 Easygraphics – Tegner en famile av fyrstikkfigurer, antall leses inn, to foreldre resten (små) barn. 49](#_Toc501041625)

[4.1.12 Samling av metoder for geometriske beregninger 50](#_Toc501041626)

[4.1.13 Tester metodene i programmet over 51](#_Toc501041627)

[4.1.14 Bruker metode for å avgjøre om et heltall er et primtall eller ikke. 52](#_Toc501041628)

[4.1.15 Easygraphics – Tegner husrekke med takutstikk 52](#_Toc501041629)

[4.2 Leksjon 4 – Roy 53](#_Toc501041630)

[4.2.1 Tegner trappetrinn figur i konsoll ved bruk av metode med ulike tegn 53](#_Toc501041631)

[4.2.2 Beregner gjennomsnittlig tverrsum for et antall tosifrede tall. Trekk av tall via metoder 54](#_Toc501041632)

[4.2.3 Leser ønsket antall ord, skriver ut setning med trekte bokstaver 55](#_Toc501041633)

[4.2.4 Samling av metoder 57](#_Toc501041634)

[4.2.5 Tester alle metodene i programmet over 58](#_Toc501041635)

[4.2.6 Easygraphics – Terningkast, trekker tall, skriver tilfeldig plassert på skjermen som en terning 59](#_Toc501041636)

[5 Leksjon 5 62](#_Toc501041637)

[5.1 Leksjon 5 – Lærebok 62](#_Toc501041638)

[5.1.1 Leser inn ett siffer og teller antall mindre enn dette i et tallsett lagret i en tabell 62](#_Toc501041639)

[5.1.2 Sorterer tallsett i avtagende rekkefølge 62](#_Toc501041640)

[5.1.3 Teller antall tall i et tallsett som ikke flytter seg etter stigende sortering 63](#_Toc501041641)

[5.1.4 Registrerer temperatur for hver dag i en uke, skriver ut oversikt, gjennomsnitt, maks og min. 63](#_Toc501041642)

[5.1.5 Tar to sorterte tabeller og fletter dem til en sortert tabell 64](#_Toc501041643)

[5.1.6 Leser inn et antall tallkarakterer, beregner gjennomsnitt, skriver ut som bokstavkarakter 65](#_Toc501041644)

[5.1.7 Tester metode som skriver ut heltallsvariabel til konsollet 66](#_Toc501041645)

[5.1.8 Tester metode som fyller en tabell med et gitt heltall 66](#_Toc501041646)

[5.1.9 Tester metode som reverserer heltallstabell 67](#_Toc501041647)

[5.1.10 Tester en metode som sjekker om en tegnsekvens er palindrom 67](#_Toc501041648)

[5.1.11 Tar to tabeller, slår dem sammen til en tabell og sorterer 69](#_Toc501041649)

[5.1.12 Tar inn tegnsekvenser, skriver ut en og en sekvens med info om det er et palindrom 69](#_Toc501041650)

[5.1.13 Leser inn fornavn og etternavn, skriver ut med små bokstaver, men stor forbokstav 70](#_Toc501041651)

[5.1.14 Simulerer tilfeldig trekning av kast med en terning, teller antall av hvert resultat 70](#_Toc501041652)

[5.1.15 Simulerer kast med to terninger og teller forekomster av hver sum. 71](#_Toc501041653)

[5.1.16 Trekker lotto-rekker. Teller rette i forhold til en vinnerrekke 72](#_Toc501041654)

[5.1.17 Bruker en 2d tabell til å representere matrise, Blir speilet om hoveddiagonalen og skrevet ut 73](#_Toc501041655)

[5.1.18 Oppretter varelager med tabell pr vare-opplusning, leser inn varenummer, all info om varen blir funnet i tabellen og skrevet ut 74](#_Toc501041656)

[5.1.19 Easygraphics – Simulerer kast med to terninger, teller antall av hver sum og lager søyler 75](#_Toc501041657)

[5.1.20 Easygraphics – Bruker 2d tabell, presenterer matrise, metode speiler og en annen skriver ut med ved svarte og hvite ruter 77](#_Toc501041658)

[5.2 Leksjon 5 - Roy 78](#_Toc501041659)

[5.2.1 Tallsett, beregner gjennomsnitt, finner tall mindre enn snitt, sjekker om snitt er i tallsett 78](#_Toc501041660)

[5.2.2 Easygraphics – Leser inn ant kast. Teller ant av hver sum, telles i tabell, presenteres i diagram med blokker. 80](#_Toc501041661)

[6 Leksjon 6 82](#_Toc501041662)

[6.1 Leksjon 6 – Lærebok 82](#_Toc501041663)

[6.1.1 Ber om navnet på en tekstfil, teller totalt antall tegn. 82](#_Toc501041664)

[6.1.2 Leser heltall fra fil. Beregner summen per linje 82](#_Toc501041665)

[6.1.3 Leser en tekstfil, teller forekomst av korte, halvlange og lange ord. Skrives ut som tall og prosent. 83](#_Toc501041666)

[6.1.4 Fjerner alle blanke linjer fra en fil. 84](#_Toc501041667)

[6.1.5 Sammenligner to tekstfiler, finne og skriver ut linjer som er forskjellige 85](#_Toc501041668)

[6.1.6 Leser en tekstfil med alfabetisk sorterte navn, skriver listen tilbake med motsatt sortering 86](#_Toc501041669)

[6.1.7 Leser fornavnsliste fra fil, nytt navn leses inn og settes i liste, skrives tilbake alfabetisk sortert 87](#_Toc501041670)

[6.1.8 Leser tekstfil, teller forekomst av alle ord med lengder under 12. Ord lengre enn 12 telles samlet 88](#_Toc501041671)

[6.1.9 Leser høydedata fra fil, jenter/gutter, median. 89](#_Toc501041672)

[6.1.10 Leser drivstoffsalg (4 ulike) fra fil for en enkelt dag. Skriver tilbake samlet omsetning. 90](#_Toc501041673)

[6.1.11 Leser personopplysninger fra fil, skriver ut navneliste sortert etter etternavn 91](#_Toc501041674)

[6.1.12 Metoder for Person 92](#_Toc501041675)

[6.1.13 Leser personopplysninger fra fil, skriver ut liste sortert på etternavn av dem som har innlest alder. 93](#_Toc501041676)

[6.1.14 Statistiske metoder for innlesing av teksstrenger, heltall og kommatall via dialogvindu, og visning av meldinger via dialogvindu (ikke nødvendigvis pensum) 94](#_Toc501041677)

[6.1.15 Registrering av skøytetider, lagret på fast tekstfil, registrert i to omganger. 2 dager Hvis tom dag 1. 95](#_Toc501041678)

[6.1.16 Metoder for neste program 98](#_Toc501041679)

[6.1.17 Easygraphics - Leser data om rektangler fra en fil, opprettes i tabell, tegnes på en grafisk flate 99](#_Toc501041680)

[6.2 Leksjon 6 – Roy 100](#_Toc501041681)

[6.2.1 Tester permanent lagring på fil ved å skrive navneliste til en sekvensiell tekstfil 100](#_Toc501041682)

[6.2.2 Leser fil med navn, åpner ny fil og skriver navnene i omvendt rekkefølge, leser inn og skriver nytt navn nederst på den opprinnelige filen 101](#_Toc501041683)

[6.2.3 Leser inn fil med fornavn,etternavn, skriver ut til fil med etternavn,fornavn 103](#_Toc501041684)

[6.2.4 Leser desimaltall fra fil, avrunder til heltall som skrives på ny fil. 104](#_Toc501041685)

[7 Eksamensoppgaver 106](#_Toc501041686)

[7.1 Hjelpemetode, teller antall siffer i et heltall, hvis positivt 106](#_Toc501041687)

[7.2 Leser informasjon fra en telstfil med produktgruppenummer og sekundantall for tlf-samtale. Lagres i to tabeller, en teller antall tlf-samtaler for hvert gr.nr og en summerer tlf-tid for hvert gr.nr. Skriver til ny fil gjennomsnittlig tlf-tid i sekund for hvert gr.nr. 106](#_Toc501041688)

[8 Min kode 108](#_Toc501041689)

[8.1 Obligatorisk oppgave 108](#_Toc501041690)

[8.1.1 Bankinfo (Main) 108](#_Toc501041691)

[8.1.2 Metoder for Bankinfo 111](#_Toc501041692)

# Leksjon 1

## Leksjon 1 - Lærebok

### 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 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);

}

}

### 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);

}

}

### Leser inn ett og ett heltall, finner sum, minste/største tall, gjennomsnitt og skriver ut avrundet til en desimal

/\* Leksjon 3: Jobb 6 (Gjennomsnitt) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 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);

}

}

### Leser inn ant terningkast, simulerer kast og teller ant kast til første 6er

/\* 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);

}

}

### Easygraphics – Fransk flagg v.2

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

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);

}

}

}

### Easygraphics – Tegner en murvegg, animerer med pause.

/\* 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;

}

}

}

# Leksjon 4 - Metoder

## Leksjon 4 – Lærebok

### Bruker metode til å lage rektangel med 2 parametre

/ \* Programmet bruker metoden skrivStjerner til å

\* lage en ny rektangel-metode med to parametre:

\* bredde og høyde. \*/

import static java.lang.System.\*;

import static java.lang.Integer.\*;

public class TegnRektangel {

public static void main(String[] args) {

int bredde = 8, høyde = 3;

// Lag innlesing fra brukeren selv!

rektangel(bredde, høyde);

}

public static void rektangel(int linjelengde, int antallLinjer) {

for (int i=1; i<=antallLinjer; i++)

skrivStjerner(linjelengde);

}

public static void skrivStjerner(int antall) {

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

out.print('\*');

out.println();

}

}

### Bruker metode til å lage trekant med en parameter

/\* Programmet bruker metoden skrivStjerner til å

\* lage en ny trekant-metode med en parameter

\* bredde som er også er lik høyde. \*/

import static java.lang.System.\*;

import static java.lang.Integer.\*;

public class TegnTrekant {

public static void main(String[] args) {

int bredde = 5;

// Lag innlesing fra brukeren selv!

trekant(bredde);

}

public static void trekant(int antallLinjer) {

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

skrivStjerner(linjeNr);

}

public static void skrivStjerner(int antall) {

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

out.print('\*');

out.println();

}

}

### Bruker metode til å skrive ut et pil-mønster

/\* Programmet bruker metoden skrivTegn

\* til å lage og teste en ny metode skrivPil

\* som skriver ut et "pil-mønster". \*/

import static java.lang.System.\*;

import static java.lang.Integer.\*;

import static javax.swing.JOptionPane.\*;

public class Pil {

public static void main(String[] args) {

int nivå = 3;

// Lag innlesing fra brukeren selv!

skrivPil(nivå);

}

public static void skrivPil(int nivåLinje) {

// Først lengre og lengre linjer...

for (int linje=1; linje<=nivåLinje; linje++)

skrivTegn('>', 2\*linje-1);

// Deretter kortere og kortere linjer igjen...

for (int linje=nivåLinje-1; linje>=1; linje--)

skrivTegn('>', 2\*linje-1);

}

public static void skrivTegn(char t, int antall) {

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

out.print(t);

out.println();

}

}

### Bruker metoder for omregning, en for lengder i cm og en i tommer.

/ \* Programmet tester to metoder for omregning

\* mellom lengder i cm og i tommer. \*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Double.\*;

public class OmregnLengde {

public static void main(String[] args) {

double lengdeCm = parseDouble(showInputDialog("Gi lengede i cm:"));

double lengdeTommer = omregnFraCm(lengdeCm);

String utTxt = "Lengde i cm: " + lengdeCm + "\n" +

"Lengde i tommer: " + lengdeTommer + "\n" +

"Kontroll i cm: " + omregnTilCm(lengdeTommer);

showMessageDialog(null, utTxt);

}

public static double omregnFraCm(double lengde) {

return lengde/2.54;

}

public static double omregnTilCm(double lengde) {

return lengde \* 2.54;

}

}

### Bruker metode fakultet, beregner noe ved bruk av en løkke

/ \* Programmet bruker metoden fakultet som via ei

\* løkke beregner: n! = n x n-1 x . . . x 2 x 1 \*/

import static java.lang.Integer.\*;

import static javax.swing.JOptionPane.\*;

public class Fakultet {

public static void main(String[] args) {

boolean ulovlig;

int n;

do { // Leser inn lovlig n-verdi

String tallTxt = showInputDialog("Gi n =");

n = parseInt(tallTxt);

ulovlig = (n < 0 || 12 < n);

if (ulovlig)

showMessageDialog(null, "Ugyldig n-verdi!");

} while (ulovlig);

showMessageDialog( null, n + "! = " + fakultet(n) );

}

public static int fakultet(int maxTall) {

int nFak = 1;

for (int tall=1; tall<=maxTall; tall++)

nFak = nFak \* tall;

return nFak;

}

}

### Leser inn et heltall, sjekker at dette ligger i et angitt invervall

/\* Programmet leser inn et heltall og sjekker at

\* dette ligger i et angitt intervall. \*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

public class Innlesing {

public static void main ( String [] args ) {

int heltall = lesHeltall (2,10);

showMessageDialog (null ," Heltall : " + heltall );

}

private static int lesHeltall (int nedre, int øvre) {

int tall;

do {

String tallTxt = showInputDialog (" Gi heltall ("+nedre+"-"+øvre+"):");

tall = parseInt ( tallTxt );

} while ( tall < nedre || øvre < tall );

return tall ;

}

}

### Leser inn et tegn, bruker metoder for å avgjøre om dette er en bokstav, stor eller liten, vokal eller konsonant.

/ \* Programmet leser et tegn og bruker metoder til å

\* avgjøre om dette er en bokstav, og i så fall stor

\* eller liten, og vokal eller konsonant. Ser kun på

\* engelske bokstaver (dvs. ikke æ,ø,å). \*/

import static java.lang.Integer.\*;

import static javax.swing.JOptionPane.\*;

public class BokstavSjekk {

public static void main(String[] args) {

String tegnTxt = showInputDialog("Gi et tegn");

char tegn = tegnTxt.charAt(0);

String ut = "Tegnet " + tegn + " er";

if ( erBokstav(tegn) ) {

ut += " en";

if ( erStorBokstav(tegn) )

ut += " stor";

else

ut += " liten";

ut += " bokstav og";

if ( erVokal(tegn) )

ut += " vokal";

else

ut += " konsonant";

}

else

ut += " ikke en bokstav";

showMessageDialog(null, ut);

}

public static boolean erBokstav(char t) {

return ('a'<=t && t<='z' || 'A'<=t && t<='Z');

}

public static boolean erStorBokstav(char t) {

return ('A'<=t && t<='Z');

}

public static boolean erVokal(char t) {

if ( erStorBokstav(t) )

t = (char)(t + 32); // slår om til liten bokstav

return (t=='a'||t=='e'||t=='i'||t=='o'||t=='u'||t=='y');

}

}

### Beregner summen av en sekvens med påfølgende heltall

/\* Programmet beregner summen av en sekvens

\* med påfølgende heltall. \*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

public class SumAvSekvens {

public static void main ( String [] args ) {

int svar = sum (3, 7);

// De to tallene bør leses inn!

showMessageDialog(null,"Sum fom 3 tom 7 = " + svar );

}

private static int sum (int fra, int til ) {

int summen = 0;

for (int tall = fra; tall <= til; tall++)

summen += tall;

return summen ;

}

}

### Easygraphics – Tegner et antall lastebiler i en kolonne, lastebilene blir større lengre frem i kolonnen

/\* Programmet tegner et antall lastebiler i en kolonne.

\* Lastebilene blir større lengre fremme i kolonnen,

\* men de kjører alle på den 'samme vegen'. \*/

import java.awt.\*;

import java.applet.\*;

public class Lastebiler extends EasyGraphics {

public static void main (String[] args) {

launch (args);

}

public void run() {

makeWindow ("Lastebilkolonne",900,500);

int antBiler = 5;

// Kan leses inn fra brukeren

int xPos=20, yPos=200, str=90;

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

tegnLastebil(xPos, yPos, str);

xPos += str + str/2;

str += 15;

}

}

public void tegnLastebil(int x, int y, int lengde) {

// Først tegnes underdelen

int yUnder = y-lengde/3-lengde/5;

drawRectangle(x, yUnder, lengde, lengde/3);

// Så tegnes lasten

int yLast = y-2\*lengde/3-lengde/5;

drawRectangle(x, yLast, 3\*lengde/4, lengde/3);

// Deretter tegnes førerhuset

int yHus = y-lengde/5-lengde/3-lengde/5;

drawRectangle(x+3\*lengde/4, yHus, lengde/6, lengde/5);

// Tre like store hjul, forskjøvet i x-retningen

int r = lengde/8;

drawCircle(x+r, y-r/2, r);

drawCircle(x+3\*r, y-r/2, r);

drawCircle(x+7\*r, y-r/2, r);

}

}

### Easygraphics – Tegner ut en husrekke med takutstikk

/\* Programmet tegner ut en husrekke slik som i læreboken,

\* men det er nå lagt til et lite takutstikk. \*/

import java.awt.\*;

import java.applet.\*;

public class Husrekke extends EasyGraphics {

public static void main (String[] args) {

launch (args);

}

public void run() {

makeWindow ("Husrekke",800,300);

tegnHusrekke(5, 50, 100, 90, 50, 25);

}

private void tegnHusrekke(int antallHus,

int startX, int startY, int bredde, int høyde, int avstand) {

for (int husNr=1; husNr<=antallHus; husNr++) {

tegnHus(startX, startY, bredde, høyde);

startX = startX + bredde + bredde/3 + avstand;

// bredde/3 er tillegget pga takutstikket, se nedenfor

}

}

private void tegnHus(int x, int y, int bredde, int høyde ) {

int veggHøyde = høyde\*2/3;

int toppVegg = y - veggHøyde;

drawRectangle(x, toppVegg, bredde, veggHøyde);

// Takutstikket er valgt til 1/6 av bredden på huset.

// Ett på hver side gjør til sammen grunnlinja i taket

// 1/3 lengre enn bredden på huset.

int ut = bredde/6;

tegnTrekant(x-ut, toppVegg, bredde + 2\*ut, høyde/3);

}

private void tegnTrekant(int x, int y, int bredde, int høyde) {

drawLine(x, y, x+bredde, y);

drawLine(x+bredde, y, x+bredde/2, y-høyde);

drawLine(x+bredde/2, y-høyde, x, y);

}

}

### Easygraphics – Tegner en famile av fyrstikkfigurer, antall leses inn, to foreldre resten (små) barn.

/\* Programmet tegner en familie av fyrstikkfigurer.

\* Antall figurer leses inn, to store foreldre og

\* resten er (mindre, alle like små) barn. \*/

import static java.lang.Integer.\*;

public class FyrstikkFamilie extends EasyGraphics {

public static void main(String[] args) {

launch(args);

}

public void run() {

makeWindow("Strekfigurer", 900, 500);

int ant = parseInt( getText("Gi ant. figurer:") );

// Plassering av midtpunktet til det første hodet

int xPos = 50, yPos = 100;

// Setter størrelsen på en voksen, og på et barn

int stor = 150, liten = stor/2;

// Setter avstand mellom figurene

int avstand = liten;

// Tegner barna lengre ned for at alle 'skal stå på gulvet'

int littNed = stor/2;

for (int i=0; i<ant; i++) {

if ( i < 2)

tegnFigur(xPos+i\*avstand,yPos,stor);

else

tegnFigur(xPos+(i+2)\*avstand/2, yPos+littNed, liten);

// Tegner barna to hakk lengre mot høyre, derfor (i+2)

pause(400);

}

}

// Ulike hjelpemetoder for å tenge figurene

private void tegnFigur(int x, int y, int str) {

int r = str/5;

tegnHode(x,y,r);

tegnKropp(x,y+r,r);

tegnBein(x,y+3\*r,r);

}

private void tegnHode(int x, int y, int str) {

drawCircle(x,y,str);

// Tegner to øyne

fillCircle(x-str/4,y-str/3,str/6);

fillCircle(x+str/4,y-str/3,str/6);

// Tegner en smilemunn....

drawArc(x,y,str/2,str/2,190,160);

}

private void tegnKropp(int x, int y, int str) {

drawLine(x,y,x,y+2\*str);

drawLine(x-str,y+str,x+str,y+str);

}

private void tegnBein(int x, int y, int str) {

drawLine(x,y,x-str,y+2\*str);

drawLine(x,y,x+str,y+2\*str);

}

}

### Samling av metoder for geometriske beregninger

/\* Verktøyklasse for geometriske beregninger.

\* Areal og omkrets av diverse figurer. \*/

public class Geometri {

// Sirkler

public static double arealSirkel(double radius) {

return Math.PI\*radius\*radius;

}

public static double omkretsSirkel(double radius) {

return 2\*Math.PI\*radius;

}

// Rektangel

public static double arealRektangel(double bredde, double høyde) {

return bredde\*høyde;

}

public static double omkretsRektangel(double bredde, double høyde) {

return 2\*bredde + 2\*høyde;

}

// Kun for rettvinklet trekant

public static double arealTrekant(double bredde, double høyde) {

return bredde\*høyde/2;

}

public static double omkretsTrekant(double bredde, double høyde) {

return bredde + høyde + Math.sqrt(bredde\*bredde + høyde\*høyde);

}

}

### Tester metodene i programmet over

/ \* Programmet tester bruk av verktøyklassen Geometri.\*/

import static javax.swing.JOptionPane.\*;

public class TestAvGeometri {

public static void main ( String [] args ) {

double rad = 3.5;

double bredde = 8.5, høyde = 10.0;

showMessageDialog (null,

"Radius: r = " + rad + "\n" +

"Areal: " + Geometri.arealSirkel(rad) + "\n" +

"Omkrets: " + Geometri.omkretsSirkel(rad) + "\n" +

"Bredde: " + bredde + ", høyde: " + høyde + "\n" +

"Rektangel - Areal: " + Geometri.arealRektangel(bredde, høyde) + "\n" +

"Rektangel - Omkrets: " + Geometri.omkretsRektangel(bredde, høyde) + "\n" +

"Trekant - Areal: " + Geometri.arealTrekant(bredde, høyde) + "\n" +

"Trekant - Omkrets: " + Geometri.omkretsTrekant(bredde, høyde) );

showMessageDialog (null, "Overflate av prisme: " + overflatePrisme(1,2,3) );

}

public static double overflatePrisme(double lengde, double bredde, double høyde) {

return 2\*Geometri.arealRektangel(lengde, bredde)

+ 2\*Geometri.arealRektangel(lengde, høyde)

+ 2\*Geometri.arealRektangel(bredde, høyde);

}

}

### Bruker metode for å avgjøre om et heltall er et primtall eller ikke.

/\* Programmet bruker en egen metode for å avgjøre om et heltall

\* er et primtalll eller ikke, for å finne alle primtall i 1-100. \*/

import static javax.swing.JOptionPane.\*;

public class Primtall {

public static void main ( String [] args ) {

String utTxt = "Primtall (1-100): ";

for (int heltall=2; heltall<=100; heltall++)

if ( erPrimtall(heltall) )

utTxt += heltall + " ";

showMessageDialog (null, utTxt);

}

public static boolean erPrimtall(int tall) {

if ( tall % 2 == 0 )

return ( tall == 2 ); // 2 er eneste partallige primtall

boolean prim = true;

for (int faktor=3; faktor<=tall/2; faktor+=2)

if ( tall % faktor == 0 )

return false;

// Ingen faktorer funnet, dvs. primtall

return true;

}

}

### Easygraphics – Tegner husrekke med takutstikk

/ \* Programmet tegner ut en husrekke slik som i læreboken,

\* men det er nå lagt til et lite takutstikk. \*/

import java.awt.\*;

import java.applet.\*;

import static java.lang.Double.\*;

public class HusrekkeUtvidet extends EasyGraphics {

public static void main (String[] args) {

launch (args);

}

public void run() {

makeWindow ("Husrekke",800,300);

double f = parseDouble( getText("Høydeforhold:") );

// f = forhold mellom høyden på husveggen og hele huset

tegnHusrekke(5, 50, 100, 90, 80, 25, f);

}

private void tegnHusrekke(int antallHus,

int startX, int startY, int bredde,

int høyde, int avstand, double andel) {

for (int husNr=1; husNr<=antallHus; husNr++) {

tegnHus(startX, startY, bredde, høyde, andel);

startX = startX + bredde + bredde/3 + avstand;

// bredde/3 er tillegget pga takutstikket, se nedenfor

}

}

private void tegnHus(int x, int y,

int bredde, int høyde, double andelHøyde) {

int veggHøyde = (int)(høyde\*andelHøyde);

int takHøyde = høyde - veggHøyde;

int toppVegg = y - veggHøyde;

drawRectangle(x, toppVegg, bredde, veggHøyde);

// Takutstikket er valgt til 1/6 av bredden på huset.

// Ett på hver side gjør til sammen grunnlinja i taket

// 1/3 lengre enn bredden på huset.

int ut = bredde/6;

tegnTrekant(x-ut, toppVegg, bredde + 2\*ut, takHøyde);

}

private void tegnTrekant(int x, int y, int bredde, int høyde) {

drawLine(x, y, x+bredde, y);

drawLine(x+bredde, y, x+bredde/2, y-høyde);

drawLine(x+bredde/2, y-høyde, x, y);

}

}

## Leksjon 4 – Roy

### Tegner trappetrinn figur i konsoll ved bruk av metode med ulike tegn

/\* Leksjon 4: Jobb nr 1

\* Utviklet av RMI - Oktober 2017

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

\* Tegner en slags trappetrinn-figur ved bruk av hjelpemetode for

\* å lage linjer (trinn) med ulike tegn. \*/

import static java.lang.System.\*;

import static java.lang.Integer.\*;

import static javax.swing.JOptionPane.\*;

public class Trappetrinn {

public static void main(String[] args) {

// Innlesing av antall trinn

String tallTekst = showInputDialog("Gi antall trinn ("+min+"-"+max+"): ");

int antTrinn = parseInt(tallTekst);

// Utskrift av trapp til konsoll

for (int trinn=1; trinn<=antTrinn; trinn++) {

skrivTrinn('T', 2\*trinn);

out.println();

}

}

// Hjelpemetode (private) til bruk i dette programmet.

// Tegner en sekvens av tegn i konsollet

private static void skrivTrinn(char t, int antall) {

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

out.print(t);

}

}

### Beregner gjennomsnittlig tverrsum for et antall tosifrede tall. Trekk av tall via metoder

/\* Leksjon 4: Jobb nr 2

\* Utviklet av RMI - Oktober 2017

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

\* Programmet beregner gjennomsnittlig tverrsum for et

\* innlest antall tosifrede heltall. Det er en egne

\* metoder for å trekke tilfeldige tall, for å beregne

\* tverrsum og for å styre programgjentak. \*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

import static java.lang.Math.\*;

public class Tverrsum {

public static void main(String[] args) {

do { // Bruker do-løkke til programgjentak

// Leser inn antall tosifrede tall

String innTxt = showInputDialog("Gi ant. tall: ");

int antTall = parseInt(innTxt);

// Trekker de tresifrede heltallene

int totalsum = 0;

for (int tallNr=1; tallNr<=antTall; tallNr++) {

int tall = trekkTall(100,999);

totalsum += tverrsum(tall);

}

// Snitt beregnet, og avrundet til nærmeste heltall vha Math.round

int snitt = (int)( round(1.0\*totalsum/antTall) );

// Beregner og skriver ut gjennomsnitt av alle tverrsummene

showMessageDialog(null, "Antall heltall: " + antTall + "\n"

+ "Snitt tverrsum: " + snitt);

} while ( ønskerFortsettelse() );

// Slutt på main-metoden

}

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

\* Her kommer hjelpemetoder til bruk i dette programmet

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

// Metode for å spørre brukeren om det er ønske om programgjentak.

private static boolean ønskerFortsettelse() {

// Brukerdialog Ja/Nei for å fortsette

String innTekst = showInputDialog("Fortsett? (Ja=1/Nei=0): ");

int svar = parseInt(innTekst);

return (svar == 1);

}

// Metode for å plukke ut hvert siffer i et 3-sifret heltall,

// legge dem sammen til tverrsum og returnere denne.

private static int tverrsum(int heltall) {

int s1 = heltall / 100;

int s2 = (heltall / 10) % 10;

int s3 = heltall % 10;

int sum = s1 + s2 + s3;

return sum;

}

// Metode for å trekke et tilfeldig heltall i området min - max

private static int trekkTall(int min, int max) {

return min + (int)( random()\*(max-min+1) );

}

// Slutt på hjelpemetodene

}

### Leser ønsket antall ord, skriver ut setning med trekte bokstaver

/\* Leksjon 4: Jobb nr 3

\* Utviklet av RMI - Oktober 2017

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

\* Programmet leser et ønsket antall ord (2-10), og skriver ut

\* en setning med dette antallet ord (i ordlengder 1-5) som er

\* skrevet i tilfeldig valgte store (engelske) bokstaver. \*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

import static java.lang.Math.\*;

public class Tegntesting {

public static void main(String[] args) {

do {

// Leser inn (lovlig) ønsket antall ord

int antOrd = lesAntallOrd(2, 10);

// Trekker tilfeldig det ønskede antallet ord

String setning = "Setning: ";

for (int ordNr=1; ordNr<=antOrd; ordNr++) {

int ordlengde = trekkTall(1,5);

for (int bokstav=1; bokstav<=ordlengde; bokstav++)

setning += trekkBokstav();

setning += " "; // Mellomrom mellom ordene

}

// Skriver ut den aktuelle setningen

showMessageDialog(null, setning);

} while ( ønskerFortsettelse() );

// Slutt på main-metoden

}

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

\* Her kommer hjelpemetoder til bruk i dette programmet

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

// Metode for å lese inn og kontrollere at antall ord er i et angitt

// område (parameterliste i metoden).

private static int lesAntallOrd(int min, int max) {

int antall;

do {

String melding = "Gi antall ord ("+min+"-"+max+"):";

antall = parseInt( showInputDialog(melding) );

if (antall < min || antall > max)

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

} while ( antall < min || antall > max );

return antall;

}

// Metode for å spørre brukeren om det er ønske om programgjentak.

private static boolean ønskerFortsettelse() {

// Brukerdialog Ja/Nei for å fortsette

String innTekst = showInputDialog("Fortsett? (Ja=1/Nei=0): ");

int svar = parseInt(innTekst);

return (svar == 1);

}

// Metode for å trekke et tilfeldig heltall i området (min,max)

private static int trekkTall(int min, int max) {

return min + (int)( random()\*(max-min+1) );

}

// Metode for å trekke en tilfeldig stor engelsk bokstav (A-Z)

private static char trekkBokstav() {

return (char)trekkTall(65,90);

}

// Slutt på hjelpemetodene

}

### Samling av metoder

/\* Leksjon 4: Jobb nr 4 (Utviklet av RMI Oktober 2017)

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

\* Min egen verktøyklasse - dvs. metodesamling. NB! Alle metoder må

\* være public for at de skal kunne brukes utenfor denne klassen.

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

\* NB! Min.java kan ligge på toppnivået i mappestrukturen for emnet.

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

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

import static java.lang.System.\*;

import static java.lang.Math.\*;

public class Min {

// Metoden skriver en sekvens av tegn i en String

public static String skrivTegn(char t, int antall) {

String sekvens="";

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

sekvens += t;

return sekvens;

}

// Metoden avgjør om et tegn er en (engelsk) bokstav

public static boolean erBokstav(char tegn) {

return ('A'<=tegn && tegn<='Z') || ('a'<=tegn && tegn<='z');

}

// Metoden slår om fra liten til stor bokstav

public static char stor(char tegn) {

if ('a'<= tegn && tegn<= 'z')

return (char)(tegn - 32);

return tegn; // Slår inn om tegn ikke er liten bokstav

}

// Metoden slår om fra stor til liten bokstav

public static char liten(char tegn) {

if ('A'<= tegn && tegn<= 'Z')

return (char)(tegn + 32);

return tegn; // Slår inn om tegn ikke er stor bokstav

}

// Metoden avrunder et tall til én desimal

public static double avrund1(double tall) {

return (int)Math.round(tall\*10)/10.0;

}

// Metoden avrunder et tall til to desimaler

public static double avrund2(double tall) {

return (int)Math.round(tall\*100)/100.0;

}

// Metoden avrunder et tall til et gitt antall desimaler

public static double avrund(double tall, int ant) {

double tiPot = Math.pow(10,ant);

return (int)Math.round(tall\*tiPot)/tiPot;

}

// Metoden leser inn et heltall i området (min-max)

public static int lesHeltall(int min, int max, String ledetekst) {

int antall=0;

do {

String tallTekst = showInputDialog(ledetekst + " ("+min+"-"+max+"): ");

antall = parseInt(tallTekst);

if ( antall < min || antall > max )

showMessageDialog(null,"Ulovlig verdi!");

} while (antall < min || antall > max);

return antall;

}

// Metoden trekker et tilfeldig heltall i området min - max

public static int trekkTall(int min, int max) {

return min + (int)( random()\*(max-min+1) );

}

}

### Tester alle metodene i programmet over

/\* Leksjon 4: Jobb nr 4 (Utviklet av RMI - Oktober 2017)

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

\* Programmet tester alle hjelpemetodene i min egen verktøyklasse (Min). \*/

import static java.lang.System.\*;

import static javax.swing.JOptionPane.\*;

public class TestMetodeklasse {

public static void main(String[] args) {

int tall = Min.lesHeltall(1,9, "Gi heltall");

out.println( Min.skrivTegn('\*', tall) );

tall = Min.trekkTall(3,10);

out.println(Min.skrivTegn('+', tall) + " (" + tall + " stk.)" );

double desimaltall = 12.34567;

out.println("Verdi avrundet til en desimal: "

+ Min.avrund1(desimaltall) );

out.println("Verdi avrundet til to desimaler: "

+ Min.avrund2(desimaltall) );

out.println("Verdi avrundet til 4 desimaler: "

+ Min.avrund(desimaltall,4) );

// Trekker tilfeldig et tegn

char tegn = (char)Min.trekkTall(48,122);

out.print("Tegnet <" + tegn + "> er ");

if ( !Min.erBokstav(tegn) )

out.print("ikke ");

out.println("en (engelsk) bokstav!");

}

}

### Easygraphics – Terningkast, trekker tall, skriver tilfeldig plassert på skjermen som en terning

/\* EKSTRAJOBB: Terningkast i EasyGraphics

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

\* Trekker tall 1-6 og skriver dem tilfeldig plassert på skjermen

\* som øyne på en terning.

\* Et arkiv over alle terningkast tegnes helt nederst i vinduet. \*/

import static java.lang.Integer.\*;

import static java.lang.Math.\*;

public class TerningkastEG extends EasyGraphics {

public static void main(String[] args) {

launch(args);

}

public void run() {

int wX = 1000, wY = 750;

int mX = wX/2, mY = wY/2;

makeWindow("Terningkast", wX, wY);

int n = 0;

do {

n = parseInt( getText("Gi antall terningkast:") );

} while ( n <= 0 || n > wX/51 );

// Ikke plass til flere enn n små-terninger nederst i vinduet

pause(400); // Kort pause før første kast...

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

int str = wX/10;

int øyne = (int)(Math.random()\*6 + 1);

int pX = 15 + (int)(Math.random()\*(wX-str-25));

int pY = 15 + (int)(Math.random()\*(wY-str-85));

switch (øyne) {

case 1: visEn(pX,pY,str); visEn(i\*50, wY-40, 30); break;

case 2: visTo(pX,pY,str); visTo(i\*50, wY-40, 30); break;

case 3: visTre(pX,pY,str); visTre(i\*50, wY-40, 30); break;

case 4: visFire(pX,pY,str); visFire(i\*50, wY-40, 30); break;

case 5: visFem(pX,pY,str); visFem(i\*50, wY-40, 30); break;

case 6: visSeks(pX,pY,str); visSeks(i\*50, wY-40, 30);

}

pause(500); // Evt. stadig kortere pauser: pause(900-5\*i);

// Visker ut (dekker over) kastet terning

setColor(255,255,255);

fillRectangle(pX-9,pY-9,str+18,str+18);

}

}

// Hjelpemetoder for å tegne terning med et antall øyne

// NB! Ikke static på hjelpemetoder i EasyGraphics

private void visEn(int x, int y, int s){

setColor(230,230,230);

fillRectangle(x-8,y-8,s+16,s+16);

setColor(0,0,0);

drawRectangle(x-8,y-8,s+16,s+16);

fillCircle(x+s/2, y+s/2, s/6);

}

private void visTo(int x, int y, int s){

setColor(230,230,230);

fillRectangle(x-8,y-8,s+16,s+16);

setColor(0,0,0);

drawRectangle(x-8,y-8,s+16,s+16);

fillCircle(x+s/6, y+s/6, s/6);

fillCircle(x+5\*s/6, y+5\*s/6, s/6);

}

private void visTre(int x, int y, int s){

setColor(230,230,230);

fillRectangle(x-8,y-8,s+16,s+16);

setColor(0,0,0);

drawRectangle(x-8,y-8,s+16,s+16);

fillCircle(x+s/6, y+s/6, s/6);

fillCircle(x+s/2, y+s/2, s/6);

fillCircle(x+5\*s/6, y+5\*s/6, s/6);

}

private void visFire(int x, int y, int s){

setColor(230,230,230);

fillRectangle(x-8,y-8,s+16,s+16);

setColor(0,0,0);

drawRectangle(x-8,y-8,s+16,s+16);

fillCircle(x+s/6, y+s/6, s/6);

fillCircle(x+5\*s/6, y+s/6, s/6);

fillCircle(x+5\*s/6, y+5\*s/6, s/6);

fillCircle(x+s/6, y+5\*s/6, s/6);

}

private void visFem(int x, int y, int s){

setColor(230,230,230);

fillRectangle(x-8,y-8,s+16,s+16);

setColor(0,0,0);

drawRectangle(x-8,y-8,s+16,s+16);

fillCircle(x+s/6, y+s/6, s/6);

fillCircle(x+5\*s/6, y+s/6, s/6);

fillCircle(x+s/2, y+s/2, s/6);

fillCircle(x+5\*s/6, y+5\*s/6, s/6);

fillCircle(x+s/6, y+5\*s/6, s/6);

}

private void visSeks(int x, int y, int s){

setColor(230,230,230);

fillRectangle(x-8,y-8,s+16,s+16);

setColor(0,0,0);

int r = (int)(0.95\*s/6);

drawRectangle(x-8,y-8,s+16,s+16);

fillCircle(x+s/6, y+s/6, r);

fillCircle(x+5\*s/6, y+s/6, r);

fillCircle(x+s/6, y+3\*s/6, r);

fillCircle(x+5\*s/6, y+3\*s/6, r);

fillCircle(x+5\*s/6, y+5\*s/6, r);

fillCircle(x+s/6, y+5\*s/6, r);

}

}

# Leksjon 5

## Leksjon 5 – Lærebok

### Leser inn ett siffer og teller antall mindre enn dette i et tallsett lagret i en tabell

/ \* Programmet leser inn ett siffer og teller antall

\* forekomster som er mindre enn dette i et tallsett

\* som allerede er lagret i en tabell.\*/

import static java.lang.Integer.\*;

import static javax.swing.JOptionPane.\*;

public class AntallMindre {

public static void main(String[] args) {

// Tallsett: Legger noen siffer i en tabell

int[] siffer = {1,2,1,2,3,1,7,1,8,2};

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

int søkesiffer = parseInt(tallTxt);

int antall = 0;

for (int i=0; i<siffer.length; i++)

if ( siffer[i] < søkesiffer )

antall++;

String ut = "Antall siffer mindre enn " + søkesiffer + ": " + antall;

showMessageDialog(null, ut);

}

}

### Sorterer tallsett i avtagende rekkefølge

/ \* Programmet sorterer et tallsett i avtagende rekkefølge,

\* og skriver det ut.\*/

import static javax.swing.JOptionPane.\*;

import java.util.\*; // Trenger Arrays-klassen

public class SorterAvtagende {

public static void main(String[] args) {

int[] tall = {1,2,1,2,9,8,7,1,8,2};

Arrays.sort(tall); // Sortert stigende - feil ordning

int[] kopi = Arrays.copyOf(tall, tall.length);

int sistePos = tall.length - 1;

for (int i=0; i<=sistePos; i++)

tall[i] = kopi[sistePos-i];

String ut = "Sortert tallsett: " + Arrays.toString(tall);

showMessageDialog(null, ut);

}

}

### Teller antall tall i et tallsett som ikke flytter seg etter stigende sortering

/ \* Programmet teller antall tall i et tallsett

\* som står i ro (er uflyttede) når tallsettet

\* blir sortert (stigende).\*/

import static javax.swing.JOptionPane.\*;

import java.util.\*; // Trenger Arrays-klassen

public class AntallUflyttede {

public static void main(String[] args) {

int[] tall = {1,2,1,2,9,8,7,1,8,2};

int[] kopi = Arrays.copyOf(tall, tall.length);

Arrays.sort(kopi);

int antallRo = 0;

for (int i=0; i<tall.length; i++)

if ( tall[i] == kopi[i] )

antallRo++;

String ut = "Antall uflyttede tall er " + antallRo + "\n" +

"Original: " + Arrays.toString(tall) + "\n" +

"Sortert : " + Arrays.toString(kopi);

showMessageDialog(null, ut);

}

}

### Registrerer temperatur for hver dag i en uke, skriver ut oversikt, gjennomsnitt, maks og min.

/ \* Program for å registrere temperatur for hver dag i en

\* uke. Skriver ut en samlet oversikt over temperaturene,

\* inkludert gjennomsnittstemperatur, min- og makstemperatur. \*/

import static java.lang.System.\*;

import static java.lang.Integer.\*;

import static java.lang.Double.\*;

import static javax.swing.JOptionPane.\*;

public class Temperatur {

public static void main(String[] args) {

// Koblingstabell: dagnr -> dagnavn

String[] navn = {"man", "tir","ons", "tor","fre", "lør","søn"};

double[] temperatur = new double[7]; // Antal dager i uka er 7

for (int dag=0; dag<7; dag++) {

String melding = "Reg. temp. for " + navn[dag] +"dag:";

String tallTxt = showInputDialog( melding );

temperatur[dag] = parseInt(tallTxt);

}

// Utskrift til konsollet

out.println("Registrerte temperaturer:");

double tempSum = 0;

double minTemp = 100, maxTemp = -100;

int minDag, maxDag;

for (int dag=0; dag<7; dag++) {

out.println( navn[dag] + "dag: " + temperatur[dag] );

tempSum += temperatur[dag];

if ( temperatur[dag] < minTemp ) {

minTemp = temperatur[dag];

minDag = dag;

}

if ( temperatur[dag] > maxTemp ) {

maxTemp = temperatur[dag];

maxDag = dag;

}

}

double tempSnitt = tempSum / 7;

out.println("Gjennomsnittstemperatur: " + tempSnitt);

out.println("Minimumstemperatur: " + minTemp);

out.println("Maksimumstemperatur: " + maxTemp);

}

}

### Tar to sorterte tabeller og fletter dem til en sortert tabell

/ \* Programmet tar to sorterte tabeller og fletter dem til en

\* sortert tabell av eksakt samme lengde som de to til sammen. \*/

import static javax.swing.JOptionPane.\*;

import java.util.\*; // Trenger Arrays-klassen

public class Fletting {

public static void main(String[] args) {

int[] tabEn = {1,2,1,2,9,8,7,1,8,2};

int[] tabTo = {1,1,3,2,4};

Arrays.sort(tabEn);

Arrays.sort(tabTo);

int sumLengde = tabEn.length + tabTo.length;

int[] tabSamlet = new int[sumLengde];

int iEn = 0, iTo = 0, i = 0;

while ( iEn < tabEn.length && iTo < tabTo.length )

if ( tabEn[iEn] < tabTo[iTo] )

tabSamlet[i++] = tabEn[iEn++];

else

tabSamlet[i++] = tabTo[iTo++];

if ( iEn == tabEn.length )

while ( i < sumLengde )

tabSamlet[i++] = tabTo[iTo++];

else

while ( i < sumLengde )

tabSamlet[i++] = tabEn[iEn++];

String utTxt = "Tabell nr 1: " + Arrays.toString(tabEn) + "\n" +

"Tabell nr 2: " + Arrays.toString(tabTo) + "\n" +

"Samlet: " + Arrays.toString(tabSamlet);

showMessageDialog(null, utTxt);

}

}

### Leser inn et antall tallkarakterer, beregner gjennomsnitt, skriver ut som bokstavkarakter

/\* Programmet leser inn et antall tallkarakter og

\* beregner gjennomsnittskarakter. Skriver ut denne

\* snittkarakteren som bokstavkarakter. \*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

public class KarakterSnitt {

public static void main(String[] args) {

char[] bokstav = {'F','E','D','C','B','A'};

int sumKarakter = 0, nr = 0;

String svar;

do {

nr++;

int karakter;

do{

String tallTxt = showInputDialog("Gi kar. nr. "+nr+":");

karakter = parseInt(tallTxt);

if ( karakter < 0 || 5 < karakter)

showMessageDialog(null,"Ugyldig! Gi ny karakter.");

} while ( karakter < 0 || 5 < karakter);

sumKarakter += karakter;

svar = showInputDialog("Flere karakter? (Ja/Nei):");

svar = svar.toUpperCase();

} while ( svar.equals("JA") );

int snittKarakter = (int)( (sumKarakter\*1.0/nr + 0.5) );

// Vanlig avrunding opp til nærmeste heltall.

String utTxt = "Antall karakterer: " + nr + "\n" +

"Snittkarakter: " + bokstav[snittKarakter];

showMessageDialog(null, utTxt);

}

}

### Tester metode som skriver ut heltallsvariabel til konsollet

/\*Programmet tester ut en metode som skriver ut en

\* heltallstabell til konsollet. Dvs. utskriften er

\* et alternativ til out.println på Arrays.toString.\*/

import static java.lang.System.\*;

public class SkrivUtTabell {

public static void main(String[] args) {

int[] tall = {4, 3, 7, 1, 2, 5};

skrivUt(tall);

}

public static void skrivUt(int[] tab) {

String txt = "[" + tab[0];

for (int i=1; i<tab.length; i++)

txt += ", " + tab[i];

txt += "]";

out.println(txt);

}

}

### Tester metode som fyller en tabell med et gitt heltall

/\* Programmet tester ut en metode som fyller

\* en tabell med et gitt heltall. \*/

import static java.lang.Integer.\*;

import static javax.swing.JOptionPane.\*;

public class FyllTabell {

public static void main(String[] args) {

String tallTxt = showInputDialog("Gi antall tall i serien: ");

int antall = parseInt(tallTxt);

int[] tall = new int[antall];

fyllOpp(tall, 5);

String utTxt = "Tallserie (alle like): " + skriv(tall);

showMessageDialog(null, utTxt);

}

// Metoden fyller tabellen med den gitte verdien

public static void fyllOpp(int[] tab, int verdi) {

for (int i=0; i<tab.length; i++)

tab[i] += verdi;

}

// Hjelpemetode: Alternativ til Arrays.toString

public static String skriv(int[] tab) {

String txt = "[" + tab[0];

for (int i=1; i<tab.length; i++)

txt += ", " + tab[i];

return txt + "]";

}

}

### Tester metode som reverserer heltallstabell

/\* Programmet tester ut en metode som reverserer

\* en heltallstabell. \*/

import static javax.swing.JOptionPane.\*;

public class Reverser {

public static void main(String[] args) {

int[] tall = {4, 3, 7, 1, 2, 5};

String utTxt = "Tallserien: " + skriv(tall) + "\n";

reverser(tall);

utTxt += "Reversert: " + skriv(tall);

showMessageDialog(null, utTxt);

}

// Metoden snur rekkefølgen av tabellverdiene

public static void reverser(int[] tab) {

int sistePos = tab.length - 1;

// Plassbytte kun inn til midten av tabellen!

for (int i=0; i<=sistePos/2; i++) {

int hjelp = tab[i];

tab[i] = tab[sistePos-i];

tab[sistePos-i] = hjelp;

}

}

// Hjelpemetode: Alternativ til Arrays.toString

public static String skriv(int[] tab) {

String txt = "[" + tab[0];

for (int i=1; i<tab.length; i++)

txt += ", " + tab[i];

return txt + "]";

}

}

### Tester en metode som sjekker om en tegnsekvens er palindrom

/\* Programmet tester ut en metode som sjekker om en

\* tegnsekvens er et palindrom, dvs. samme sekvens

\* uansett om den er lest fra høyre eller venstre. \*/

import static javax.swing.JOptionPane.\*;

import java.util.\*;

public class Palindrom {

public static void main(String[] args) {

/\*

// Først en enklere versjon: Kontroll av tegnsekvens

String innTxt = showInputDialog("Gi en tekst:");

String kopiTxt = "";

int sistePos = innTxt.length()-1;

for (int i=0; i<=sistePos; i++)

kopiTxt += innTxt.charAt(sistePos-i);

String utTxt = "Original: " + innTxt + "\n" +

"Reversert: " + kopiTxt + "\n" +

"Dvs. teksten er ";

if ( innTxt.equals(kopiTxt) )

utTxt += "et palindrom";

else

utTxt += "ikke et palindrom";

showMessageDialog(null, utTxt);

\*/

// Versjon 2: Reverser tabellen

String innTxt = showInputDialog("Gi en tekst:");

char[] tekst = new char[innTxt.length()];

for (int i=0; i<tekst.length; i++)

tekst[i] = innTxt.charAt(i);

char[] kopiTekst = Arrays.copyOf(tekst, tekst.length);

reverser(kopiTekst);

String utTxt = "Inntekst: " + skriv(tekst) + "\n" +

"Reversert: " + skriv(kopiTekst) + "\n" +

"Dvs. teksten er ";

if ( Arrays.equals(tekst,kopiTekst) )

utTxt += "et palindrom";

else

utTxt += "ikke et palindrom";

showMessageDialog(null, utTxt);

}

// Metoden snur rekkefølgen av tabellverdiene

public static void reverser(char[] tab) {

int sistePos = tab.length - 1;

// Plassbytte kun inn til midten av tabellen!

for (int i=0; i<=sistePos/2; i++) {

char hjelp = tab[i];

tab[i] = tab[sistePos-i];

tab[sistePos-i] = hjelp;

}

}

// Hjelpemetode: Alternativ til Arrays.toString

public static String skriv(char[] tab) {

String txt = "[" + tab[0];

for (int i=1; i<tab.length; i++)

txt += ", " + tab[i];

return txt + "]";

}

}

### Tar to tabeller, slår dem sammen til en tabell og sorterer

/\* Programmet tar to tabeller og slår dem sammen til

\* en tabell som sorteres ved hjelp av Arrays.sort. \*/

import static javax.swing.JOptionPane.\*;

import java.util.\*; // Trenger Arrays-klassen

public class Flettemetode {

public static void main(String[] args) {

int[] tabEn = {1,2,1,2,9,8,7,1,8,2};

int[] tabTo = {1,1,3,2,4};

int[] tabSamlet = samle(tabEn, tabTo);

Arrays.sort(tabSamlet);

String utTxt = "Tabell nr 1: " + Arrays.toString(tabEn) + "\n" +

"Tabell nr 2: " + Arrays.toString(tabTo) + "\n" +

"Samlet: " + Arrays.toString(tabSamlet);

showMessageDialog(null, utTxt);

}

public static int[] samle (int[] tabA, int[] tabB) {

int nyLengde = tabA.length + tabB.length;

int[] nyTab = new int[nyLengde];

for (int i=0; i<tabA.length; i++) {

nyTab[i] = tabA[i];

}

for (int j=0; j<tabB.length; j++) {

nyTab[tabA.length+j] = tabB[j];

}

return nyTab ; // Returnerer referanse til ny tabell

}

}

### Tar inn tegnsekvenser, skriver ut en og en sekvens med info om det er et palindrom

/\* Programmet tar inn kommandolinjeparametre (tegnsekvenser)

\* som legges i args. Skriver ut igjen en og en sekvens med

\* info om tegnsekvensen er et palindrom, eller ikke.

\*

\* NB! Programmet kjøres via kommandovindu/konsoll \*/

import static java.lang.System.\*;

public class ParameterPalindrom {

public static void main(String[] args) {

for (int i=0; i<args.length; i++) {

String innTxt = args[i];

out.print("Sekvens nr. " + (i+1)+": " + innTxt);

String kopiTxt = "";

int sistePos = innTxt.length()-1;

for (int j=0; j<=sistePos; j++)

kopiTxt += innTxt.charAt(sistePos-j);

if ( innTxt.equals(kopiTxt) )

out.println(" (palindrom)");

else

out.println(" (ikke palindrom)");

}

}

}

### Leser inn fornavn og etternavn, skriver ut med små bokstaver, men stor forbokstav

/\* Programmet leser fornavn og etternavn, og skriver

\* dem ut igjen i små bokstaver, men stor forbokstav. \*/

import static javax.swing.JOptionPane.\*;

public class NavnSkriving {

public static void main(String[] args) {

String fornavn = showInputDialog("Gi fornavn:");

String etternavn = showInputDialog("Gi etternavn:");

etternavn = etternavn.toLowerCase();

// Viser to ulike måter å lage nytt navn av stor

// forbokstav, etterfulgt av kun små bokstaver

// Først: Fornavnet

char forbokstav = Character.toUpperCase(fornavn.charAt(0));

fornavn = fornavn.toLowerCase();

String nyFornavn = ""+forbokstav;

for (int i=1; i<fornavn.length(); i++)

nyFornavn += fornavn.charAt(i);

// Så: Etternavnet

etternavn = Character.toUpperCase(etternavn.charAt(0))

+ etternavn.substring(1);

// substring(1) betyr hele sekvensen f.o.m. posisjon nr 1

showMessageDialog(null, nyFornavn + " " + etternavn);

}

}

### Simulerer tilfeldig trekning av kast med en terning, teller antall av hvert resultat

/\* Programmet simulerer tilfeldig trekning som kast med

\* en terning. Teller antall av hvert kastresultat 1-6.\*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

public class Terningkast {

public static void main(String[] args) {

int[] antallAv = new int[6]; // 6 kastresultat skal telles

int antallKast = parseInt( showInputDialog("Gi antall kast:") );

for (int i=0; i<antallKast; i++) {

int øyne = (int)(Math.random()\*6); // Trekker 0-5

antallAv[øyne]++; // Teller opp ett hakk på kastresultatet

}

String utTxt = "Resultat fra " + antallKast + " kast:" + "\n";

for (int i=0; i<6; i++)

utTxt += "Ant. " + (i+1) + "-ere: " + antallAv[i] + "\n";

showMessageDialog(null, utTxt);

}

}

### Simulerer kast med to terninger og teller forekomster av hver sum.

/\* Programmet simulerer kast med to terninger og teller

\* antall forekomster av hver øyne-sum 2-12. \*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

public class ToTerningkast {

public static void main(String[] args) {

int[] antallAv = new int[11]; // 11 muligheter for sum øyne

int antallKast = parseInt( showInputDialog("Gi antall kast:") );

for (int i=0; i<antallKast; i++) {

int øyneEn = (int)(Math.random()\*6); // Kaster første terning

int øyneTo = (int)(Math.random()\*6); // Kaster andre terning

antallAv[øyneEn+øyneTo]++; // Teller opp dette kastresultatet

}

int[] andelAv = new int[11]; // Prosent av hver mulig sum øyne

for (int i=0; i<11; i++)

andelAv[i] = antallAv[i]\*100/antallKast;

String utTxt = "Resultat fra " + antallKast + " kast:" + "\n";

for (int i=0; i<=10; i++) {

if ( i < 8 ) utTxt += " ";

utTxt += (i+2) + ": ";

for (int j=0; j<andelAv[i]; j++)

utTxt += '\*';

utTxt += "\n";

}

showMessageDialog(null, utTxt);

}

}

### Trekker lotto-rekker. Teller rette i forhold til en vinnerrekke

/\* Programmet trekker tilfeldig ulike Lotto-rekker (7 tall, 1-34),

\* på en kupong (minst 2, maks 10 rekker). Programmet skal også kunne

\* telle opp antall rette på hver rekke i forhold til en vinnerrekke. \*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

import java.util.\*;

public class LottoKupong {

public static void main(String[] args) {

int[][] kupong = new int[10][7]; // 10 rekker a 7 tall

int antallRekker;

do {

antallRekker = parseInt( showInputDialog("Gi antall rekker (2-10):") );

} while ( antallRekker < 2 || 10 < antallRekker );

for (int i=0; i<antallRekker; i++) {

boolean fraFør;

int[] rekke;

do {

fraFør = false;

rekke = trekkRekke(34);

for (int j=0; j<i; j++)

if ( Arrays.equals(kupong[j],rekke) )

fraFør = true;

} while ( fraFør );

kupong[i] = rekke;

}

String utTxt = "Lotto-kupong:" + "\n";

for (int i=0; i<antallRekker; i++) {

if ( i < 9 ) utTxt += " ";

utTxt += (i+1) + ": " + skriv(kupong[i]) + "\n";

}

showMessageDialog(null, utTxt);

int[] vinnerRekke = {2,4,6,8,10,12,14};

utTxt = "Vinner-rekke: " + skriv(vinnerRekke) + "\n";

utTxt += "Antall rette på kupongen:" + "\n";

for (int i=0; i<antallRekker; i++) {

if ( i < 9 ) utTxt += " ";

utTxt += (i+1)+": "+ skriv(kupong[i]);

utTxt += " (" + sjekkRette(kupong[i],vinnerRekke)+ ")" + "\n";

}

showMessageDialog(null, utTxt);

}

public static int[] trekkRekke(int max) {

int[] ny = new int[7];

for (int i=0; i<7; i++) {

int tall;

boolean fraFør;

do {

fraFør = false;

tall = (int)(Math.random()\*34) +1 ;

for (int j=0; j<i; j++)

if ( tall == ny[j] )

fraFør = true;

} while ( fraFør );

ny[i] = tall;

}

Arrays.sort(ny);

return ny;

}

public static String skriv(int[] tab) {

String txt = " ";

for (int i=0; i<tab.length; i++) {

if ( tab[i] <= 9 ) txt += " ";

txt += tab[i] + " ";

}

return txt;

}

public static int sjekkRette(int[] enRekke, int[] fasit) {

int antRette = 0;

for (int i=0; i<enRekke.length; i++)

for (int j=0; j<fasit.length; j++)

if ( enRekke[i] == fasit[j] )

antRette++;

return antRette;

}

}

### Bruker en 2d tabell til å representere matrise, Blir speilet om hoveddiagonalen og skrevet ut

/\* Programmet bruker en to-dimensjonal tabell til å

\* representere en matrise. En metode speiler matrisa

\* om hoveddiagonalen, og en annen kan skrive ut matrisa. \*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

public class Matriser {

public static void main(String[] args) {

int[][] matrise = {{0,0,1,1,1,1,0,0},

{0,1,0,0,0,0,1,0},

{0,1,0,0,0,0,1,0},

{0,0,1,1,1,1,0,0},

{0,1,0,0,0,0,1,0},

{0,1,0,0,0,0,1,0},

{0,1,0,0,0,0,1,0},

{0,0,1,1,1,1,0,0}};

String utTxt = "Matrise:" + "\n" + skriv(matrise) + "\n";

speile(matrise);

utTxt += "Speilet matrise:" + "\n" + skriv(matrise);

showMessageDialog(null, utTxt);

}

public static void speile(int[][] tab) {

for (int i=0; i<tab.length; i++)

for (int j=0; j<i; j++) {

int h = tab[i][j];

tab[i][j] = tab[j][i];

tab[j][i] = h;

}

}

public static String skriv(int[][] tab) {

String txt = "";

for (int i=0; i<tab.length; i++) {

for (int j=0; j<tab.length; j++)

txt += tab[i][j] + " ";

txt += "\n";

}

return txt;

}

}

### Oppretter varelager med tabell pr vare-opplusning, leser inn varenummer, all info om varen blir funnet i tabellen og skrevet ut

/\* Programmet oppretter et lite varelager, med en tabell pr

\* vare-opplysning (varenr, -navn, -antall og -pris). Det skal

\* leses inn et varenummer, og all info om denne varen skal

\* gjenfinnes i tabellene og skrives ut. \*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

import static java.lang.System.\*;

public class Varelager {

public static void main(String[] args) {

int[] nrTab = {1,2,3};

String[] navnTab = {"Spade","Hakke","Spett"};

double[] prisTab = {220.5,199,170};

int[] antallTab = {1,2,3};

int vareNr = parseInt( showInputDialog("Gi varenr.:") );

String utTxt;

int nrPos = finn(vareNr, nrTab);

if ( nrPos < 0 )

utTxt = "Ulovlig varenr. " + vareNr;

else {

utTxt = "Varenr...... " + vareNr + "\n"

+ "Navn........ " + navnTab[nrPos] + "\n"

+ "Pris........ " + prisTab[nrPos] + "\n"

+ "Beholdning.. " + antallTab[nrPos];

}

out.println(utTxt);

}

public static int finn(int nr, int[] tab) {

int pos = -1;

for (int i=0; i<tab.length; i++)

if ( tab[i] == nr )

pos = i;

return pos;

}

}

### Easygraphics – Simulerer kast med to terninger, teller antall av hver sum og lager søyler

/\* Programmet simulerer kast med to terninger og teller

\* antall forekomster av hver øyne-sum 2-12. Resultatene

\* presenteres grafisk i et søylediagram. \*/

import static java.lang.Integer.\*;

public class GrafikkToTerninger extends EasyGraphics {

public static void main(String[] args) {

launch(args);

}

public void run() {

makeWindow("Kast med to terninger");

int[] antallAv = new int[13]; // 11 muligheter for sum øyne,

// lagret på indeksene 2 - 12

// Leser inn antall kast, ikke for få, og ikke flere enn det

// som er nødvendig for å få en rimelig fordeling av sum øyne.

int antallKast;

do {

antallKast = parseInt( getText("Gi antall kast (100-8000):") );

} while (antallKast < 100 || antallKast > 8000);

// Gjennomfører ønsket antall kast og registrerer sum øyne på

// de to terningene i en telletabell. Tabellen er litt for stor,

// og indeksene 0 og 1 står ubrukte som uaktuelle sum-verdier.

for (int i=0; i<antallKast; i++) {

int øyneEn = (int)(Math.random()\*6) + 1; // Kaster første terning

int øyneTo = (int)(Math.random()\*6) + 1; // Kaster andre terning

antallAv[øyneEn+øyneTo]++; // Teller opp dette kastresultatet

}

// Regner om antallsverdiene til prosenter, slik at ingen

// av de ulike søylene blir høyere enn det grafiske vinduet.

int[] andelAv = new int[13]; // Prosent av hver mulig sum øyne

for (int i=2; i<=12; i++)

andelAv[i] = antallAv[i]\*100/antallKast;

// Lager en overskrift på søylediagrammet

setFont("Arial",18);

drawString("Resultat fra " + antallKast + " kast.",300,50);

pause(150);

setFont("Arial",14);

fillRectangle(80,600,620,2); // Tegner x-aksen som tykk strek

fillRectangle(100,80,2,540); // Tegner y-aksen som tykk strek

// Setter enheter på x-aksen, vertikal strek 10 piksler.

// Skriver ulike sum øyne som enheter under aksemerkene.

for (int i=2; i<=12; i++) {

drawLine(50+i\*50,595,50+i\*50,605);

drawString(""+i, 50+i\*50-5, 630);

pause(90);

}

// Setter enheter på y-aksen, horisontal strek 10 piksler.

// Skriver prosentverdier 10-100 som aksemerker

for (int i=1; i<=5; i++) {

drawLine(95, 600-i\*100, 105, 600-i\*100);

drawString(""+i\*5+"%", 60, 600-i\*100+5);

pause(90);

}

// Legger på røde linjer for å lette lesing av %-verdi

setColor(255,0,0);

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

drawLine(100, 600-i\*50, 700, 600-i\*50);

pause(90);

}

// Tegner blå søyler, dvs. andelen av hver mulige sum øyne.

setColor(0,0,255);

for (int i=2; i<=12; i++) {

int x = 50+i\*50-20;

int y = andelAv[i]\*20;

fillRectangle(x, 600-y, 40, y);

pause(150);

}

}

}

### Easygraphics – Bruker 2d tabell, presenterer matrise, metode speiler og en annen skriver ut med ved svarte og hvite ruter

/\* Programmet bruker en to-dimensjonal tabell til å

\* representere en matrise. En metode speiler matrisa

\* om hoveddiagonalen, og en annen kan skrive ut matrisa

\* grafisk ved hvite (0) eller svarte ruter (1). \*/

public class GrafikkMatriser extends EasyGraphics {

public static void main(String[] args) {

launch(args);

}

public void run() {

makeWindow("Matriser");

int[][] matrise = {{0,0,1,1,1,1,0,0},

{0,1,0,0,0,0,1,0},

{0,1,0,0,0,0,1,0},

{0,0,1,1,1,1,0,0},

{0,1,0,0,0,0,1,0},

{0,1,0,0,0,0,1,0},

{0,1,0,0,0,0,1,0},

{0,0,1,1,1,1,0,0}};

skriv(matrise, 50, 50, 20);

speile(matrise);

skriv(matrise, 400, 50, 20);

}

public void speile(int[][] tab) {

for (int i=0; i<tab.length; i++)

for (int j=0; j<i; j++) {

int h = tab[i][j];

tab[i][j] = tab[j][i];

tab[j][i] = h;

}

}

public void skriv(int[][] tab, int x, int y, int str) {

// tab = matrisa, (x,y) = øvre venstre hjørne for utskrift,

// str = størrelse på smårute (tegnet 0, eller 1)

for (int i=0; i<tab.length; i++)

for (int j=0; j<tab.length; j++) {

if( tab[i][j] == 0 )

drawRectangle(x+j\*str, y+i\*str, str, str);

else

fillRectangle(x+j\*str, y+i\*str, str, str);

pause(50);

}

}

}

## Leksjon 5 - Roy

### Tallsett, beregner gjennomsnitt, finner tall mindre enn snitt, sjekker om snitt er i tallsett

\* Ser på et tallsett (med verdier i området 1-10), beregner

\* gjennomsnittet av dem og finner antall tall som er mindre

\* enn snittet.

\* Sjekker også om snittet finnes i tallsettet.

\* Avgjør hvilket tall som er hyppigst forekommende.

\* NB! Programmet bruker metoder i Min-klassen (se leksjon 4)

import static java.lang.System.\*;

// Dette programmet skriver resultat i konsollet!

public class TallanalyseV2 {

public static void main(String[] args) {

// Versjon 1: Setter tabellen via en verdiliste

// int[] tall = {10, 1, 10, 3, 9, 8, 10, 6, 2, 1};

// Leser antall tall som skal analyseres

int antTall = Min.lesHeltall(1, 100);

// Setter tabellen via en lengde-deklarasjon

int[] tall = new int[antTall];

// Fyller tabellen med tilfeldige tall (1-10)

for (int i=0; i<antTall; i++)

tall[i] = Min.trekkTall(1,10);

// Skriver ut alle tallene for kontroll

for (int i=0; i<antTall; i++)

out.print(tall[i] + " ");

out.println(); // Linjeskift etter siste tall

// Beregner (heltallig) gjennomsnittet av tallene

int sum = 0;

// NB! Vi kan fortsatt bruke tabellens egen lengde

for (int i=0; i<tall.length; i++)

sum += tall[i];

int snitt = (int)(sum / tall.length + 0.5);

out.println("Snittet av tallene er " + snitt);

// Versjon 1:

// Lister opp tall som er mindre enn snittet

out.print("Tall mindre enn snittet: ");

for (int i=0; i<tall.length; i++)

if ( tall[i] < snitt )

out.print(tall[i] + " ");

out.println();

// Versjon 2:

// Teller antall tall som er mindre enn snittet

int antMindre = 0;

for (int i=0; i<tall.length; i++)

if ( tall[i] < snitt )

antMindre++;

out.println("Antall tall som er mindre enn snittet: " + antMindre);

// Finnes snitt som et av tallene?

boolean funnet=false;

int pos=0;

while ( !funnet && pos<tall.length )

if ( tall[pos] == snitt )

funnet = true;

else

pos++;

String svar = "Ja!";

if ( !funnet ) svar = "Nei...";

out.println("Snitt-verdien er et av tallene: " + svar);

// Hvilket tall er hyppigst forekommende?

// Teller først antall av hvert siffer

int[] antall = new int[11]; // Indeksert 0-10, skal IKKE bruke 0

for (int i=0; i<tall.length; i++)

antall[ tall[i] ]++; // NB! tall[i] er et tall 1-10

// Søker etter største verdi

int max = antall[1]; // Antar at største antall er fremst i tabellen

int hyppigst = 1; // Første posisjon er tallet 1

for (int i=2; i<10; i++)

if ( max < antall[i]) {

max = antall[i];

hyppigst = i;

}

out.println("Hyppigst forekommende tall er: " + hyppigst + " ("+max+" stk)");

// NB! Dersom det er flere tall som har samme max-antallet, så skrives kun

// ut det første av tallene som det er flest av.

}

}

### Easygraphics – Leser inn ant kast. Teller ant av hver sum, telles i tabell, presenteres i diagram med blokker.

/\* Leksjon 5: Jobb nr 3

\* Utviklet av RMI - November 2017

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

\* Ser på sum øyne ved kast av to terninger, og teller antall

\* forekomster av hver sum. Ønsket antall kast leses inn fra

\* brukeren. Antall forekomster telles i en tabell.

\* Resultatene fra kastene presenteres i diagram med blokker

\* som i lengden avspeiler forekomstene.

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

\* NB! Programmet bruker ikke metodekall til Min-klassen. Her

\* har vi hentet inn hele metodekoden. \*/

import static java.lang.System.\*;

import static java.lang.Integer.\*;

public class DiagramTerningkast extends EasyGraphics {

public static void main(String[] args) {

launch(args);

}

public void run () {

int winX=850, winY=650;

makeWindow("Sum øyne i to terningkast", winX, winY);

// Leser inn ønsket antall kast

String tallTekst = getText("Gi antall kast:");

int antKast = parseInt(tallTekst);

// Oppretter telletabellen for verdier 2-12

int[] antallAv = new int[13]; // Høyeste sum er 12

// Foretar (simulering av) terningkastene

for (int kast=1; kast<=antKast; kast++) {

// Kaster første terning

int øyne1 = trekkTall(1,6);

// Kaster andre terning

int øyne2 = trekkTall(1,6);

// Beregner sumen av øyne på terningene

int sumØyne = øyne1 + øyne2;

// Øker antallet av denne summen ett hakk

antallAv[sumØyne]++;

}

// Vindustørrelse minus to marger, delt på lengste blokk

double faktor = (double)(winX-200)/maximum(antallAv);

// Lager diagrammet

setFont("Arial", 40);

drawString("Kastresultat: Sum øyne", 100, 50);

for (int i=2; i<antallAv.length; i++) {

int inn = 50;

if (i < 10)

inn = 70;

setFont("Arial", 30);

setColor(0,0,0);

drawString(i+": ", inn, i\*50);

setColor(0,0,255);

fillRectangle(100, i\*50-30, (int)(faktor\*antallAv[i]), 35);

setFont("Arial", 20);

drawString(antallAv[i]+"", 100 + (int)(faktor\*antallAv[i]) + 10, i\*50);

// Skriver antallet kast litt (10 pix) bak hver blokk

}

}

// Hjelpemetoder

// Metoden trekker et tilfeldig heltall i området min - max

private static int trekkTall(int min, int max) {

return min + (int)(Math.random()\*(max-min+1) );

}

// Metoden finner det største tallet i en tabell

private static int maximum(int[] tab) {

int max = tab[0];

for (int i=2; i<tab.length; i++)

if (max < tab[i])

max = tab[i];

return max;

}

}

# Leksjon 6

## Leksjon 6 – Lærebok

### Ber om navnet på en tekstfil, teller totalt antall tegn.

/\* Programmet ber om navnet på en tekstfil, leser innholdet på

\* denne og teller totalt antall tegn (alle slags tegn). \*/

import static javax.swing.JOptionPane.\*;

import java.io.\*; // Diverse fil-klasser

import java.util.\*; // Spesielt Scanner-klassen

public class LesFraFil {

public static void main(String[] args) throws Exception {

// Åpner fil for innlesing

String filnavn = showInputDialog("Gi filnavn:");

File fil = new File(filnavn);

Scanner leser = new Scanner(fil);

int antallTegn = 0;

while ( leser.hasNextLine() ) {

String linje = leser.nextLine();

antallTegn += linje.length();

}

leser.close();

String utTxt = "Filnavn: " + fil + "\n"

+ "Totalt antall tegn: " + antallTegn;

showMessageDialog(null, utTxt);

}

}

### Leser heltall fra fil. Beregner summen per linje

/\* Programmet leser heltall fra en fil med ukjent antall

\* linjer, og ukjent antall tall pr linje. Telleresultat

\* skrives ut i en ny fil med 'Res' foran filnavnet. \*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

import java.io.\*; // Diverse fil-klasser

import java.util.\*; // Spesielt Scanner-klassen

public class LesHeltallFraFil {

public static void main(String[] args) throws Exception {

// Åpner fil med heltall for innlesing

String filnavn = showInputDialog("Gi filnavn:");

File fil = new File(filnavn);

Scanner leser = new Scanner(fil);

// Åpner resultatfil for skriving

PrintWriter skriver = new PrintWriter("Res"+fil);

int linjenr = 0;

skriver.println("Resultat fra: " + fil);

while ( leser.hasNextLine() ) {

String linje = leser.nextLine();

int antall, sum;

if (linje.length() > 0 ) {

// Linje med tall på, finn ut hvor mange...

String[] dataTab = linje.split(" ");

antall = dataTab.length;

// Beregn summen av tallen på linjen

sum=0;

for (int i=0; i<dataTab.length; i++) {

int tall = parseInt(dataTab[i]);

sum += tall;

}

}

else { // Tom linje, ingen heltall her...

antall = 0;

sum = 0;

}

linjenr++;

skriver.println(linjenr + ": " + antall + " - " + sum);

}

leser.close();

skriver.close();

showMessageDialog(null, "Filen er ferdigbehandlet!");

}

}

### Leser en tekstfil, teller forekomst av korte, halvlange og lange ord. Skrives ut som tall og prosent.

/\* Programmet leser en tekstfil og teller opp forekomsten

\* av hhv korte ord (1-4), halvlange ord (5-10) og lange

\* ord (>10). Resultatene skrives ut både som abslolutte

\* tall og som prosenter. \*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

import java.io.\*; // Diverse fil-klasser

import java.util.\*; // Spesielt Scanner-klassen

public class Ordlengder {

public static void main(String[] args) throws Exception {

String filnavn = showInputDialog("Gi filnavn:");

File fil = new File(filnavn);

Scanner leser = new Scanner(fil);

int kort = 0, middels = 0, lang = 0;

while ( leser.hasNextLine() ) {

String linje = leser.nextLine();

String[] ordliste = linje.split(" ");

for (int i=0; i<ordliste.length; i++) {

int ordlengde = ordliste[i].length();

if ( ordlengde <= 4 )

kort++;

else

if ( ordlengde <= 10 )

middels++;

else

lang++;

}

}

leser.close();

int antOrd = kort + middels + lang;

if ( antOrd > 0 ) {

int kortProsent = 100\*kort / antOrd;

int middelsProsent = 100\*middels / antOrd;

int langProsent = 100\*lang / antOrd;

String ut = "Resultat." + "\n"

+ "Ant. korte ord: " + kort

+ ", dvs. " + kortProsent + "%" + "\n"

+ "Ant. halvlange ord: " + middels

+ ", dvs. " + middelsProsent + "%" + "\n"

+ "Ant. lange ord: " + lang

+ ", dvs. " + langProsent + "%";

showMessageDialog(null, ut);

}

else

showMessageDialog(null, "Filen inneholdt ingen ord!");

}

}

### Fjerner alle blanke linjer fra en fil.

/\* Programmet fjerner alle blanke linjer fra en Java-fil.

\* Filnavnet er en kommandolinjeparameter. \*/

import static javax.swing.JOptionPane.\*;

import static java.lang.System.\*;

import java.io.\*; // Diverse fil-klasser

import java.util.\*; // Spesielt Scanner-klassen

public class FjernLinjer {

public static void main(String[] args) throws Exception {

// Filnavn er kommandolinjeparameter

String filnavn = args[0];

File fil = new File(filnavn);

Scanner leser = new Scanner(fil);

String filTxt = "";

while ( leser.hasNextLine() ) {

String linje = leser.nextLine();

if ( linje.length() > 0 )

filTxt += linje + "\n";

}

leser.close();

PrintWriter skriver = new PrintWriter(fil);

skriver.println(filTxt);

skriver.close();

showMessageDialog(null, "Filbehandlingne er ferdig!");

}

}

### Sammenligner to tekstfiler, finne og skriver ut linjer som er forskjellige

/\* Programmet sammenligner to tekstfiler. Finner

\* og skriver ut linjer som er forksjellige. \*/

import java.io.\*;

import java.util.\*;

import static javax.swing.JOptionPane.\*;

public class Diff {

public static void main(String[] args) {

// 'ut' er en felles sluttmelding til brukeren

String ut = "Sammenliknede filer" + "\n";

try {

String filnavn = showInputDialog("Gi filnavn (fil nr 1):");

File fil = new File(filnavn);

Scanner leser1 = new Scanner(fil, "utf8");

// NB! Det kan være norske bokstaver på filene, og da

// er info om filens tegnsett en mulig parameter til

// Scanner-konstruktøren.

ut += "Fil nr 1: " + filnavn;

filnavn = showInputDialog("Gi filnavn (fil nr 2):");

fil = new File(filnavn);

Scanner leser2 = new Scanner(fil, "utf8");

ut += "Fil nr 2: " + filnavn;

// Da er begge datastrømmene åpne

int antUlike = 0;

String ulikeLinjer = "";

int linjeNr = 0;

while ( leser1.hasNextLine() ) {

String linjeFra1 = leser1.nextLine();

String linjeFra2 = leser2.nextLine();

linjeNr++;

if ( !linjeFra1.equals(linjeFra2) ) {

antUlike++;

ulikeLinjer += linjeNr + ": " + linjeFra1 + "\n"

+ linjeNr + ": " + linjeFra2 + "\n\n";

}

}

if ( antUlike > 0 ) {

ut += "Følgende linjer var ulike: " +

"\n" + ulikeLinjer +

"\n" + "Antall ulike linjer: " + antUlike;

}

else

ut += "Ingen linjer var ulike! Ant.linjer = "+linjeNr;

}

catch (Exception e) {

ut = "Filsammenlikningen lot seg ikke utføre!";

}

showMessageDialog(null, ut);

}

}

### Leser en tekstfil med alfabetisk sorterte navn, skriver listen tilbake med motsatt sortering

/\* Programmet leser lese en tekstfil med alfabetisk sorterte

\* navn, ett navn pr linje, og skriver listen tilbake til den

\* samme filen men, med motsatt sortering. Filnavnet er en

\* kommandolinjeparameter. \*/

import static javax.swing.JOptionPane.\*;

import java.io.\*; // Diverse fil-klasser

import java.util.\*; // Spesielt Scanner-klassen

public class SnuSortertNavneliste {

public static void main(String[] args) throws Exception {

// Lager stor nok navneliste med plass alle navn på filen

final int ANT\_NAVN = 1000;

String[] navneliste = new String[ANT\_NAVN];

// Filnavnet er kommandolinjeparameter

String filnavn = args[0];

File fil = new File(filnavn);

Scanner leser = new Scanner(fil);

// Leser inn en alfabetisk sortert navneliste

int i = 0;

while ( leser.hasNextLine() )

navneliste[i++] = leser.nextLine();

leser.close();

int logiskLengde = i-1;

// Skriver listen tilbake til filen med motsatt sortering

PrintWriter skriver = new PrintWriter(filnavn);

for (i=0; i<=logiskLengde; i++)

skriver.println(navneliste[logiskLengde-i]);

skriver.close();

}

}

### Leser fornavnsliste fra fil, nytt navn leses inn og settes i liste, skrives tilbake alfabetisk sortert

/\* Programmet leser en fornavnsliste fra fil og et nytt

\* navn leses fra brukeren og settes inn i lista slik at

\* den kan skrives tilbake til filen alfabetisk sortert. \*/

import static javax.swing.JOptionPane.\*;

import static java.lang.System.\*;

import java.io.\*; // Diverse fil-klasser

import java.util.\*; // Spesielt Scanner-klassen

public class SettInnSortert {

public static void main(String[] args) throws Exception {

String filnavn = showInputDialog("Gi filnavn:");

File fil = new File(filnavn);

Scanner leser = new Scanner(fil);

// Hvor mange navn er det på filen?

int antNavn = 0;

while ( leser.hasNextLine() ) {

String navn = leser.nextLine();

antNavn++;

}

leser.close();

// Lager navneliste med plass til ett nytt navn

String[] navneliste = new String[antNavn+1];

// Reåpner den samme filen

leser = new Scanner(fil);

int nr = 0;

for (int i=0; i<antNavn; i++)

navneliste[i] = leser.nextLine();

leser.close();

// Legger inn nytt navn bakerst i navnelisten

navneliste[antNavn] = showInputDialog("Gi nytt fornavn:");

Arrays.sort(navneliste);

PrintWriter skriver = new PrintWriter(filnavn);

for (int i=0; i<=antNavn; i++)

skriver.println(navneliste[i]);

skriver.close();

out.println("Filbehandling ferdig. Reåpne filen: "+filnavn);

}

}

### Leser tekstfil, teller forekomst av alle ord med lengder under 12. Ord lengre enn 12 telles samlet

/\* Programmet leser en tekstfil og teller opp forekomsten

\* av alle ord med lengder under 12 (ord av lengde 1, av

\* lengde 2, osv.) Ord som er lengre en 12 telles samlet. \*/

import static javax.swing.JOptionPane.\*;

import java.io.\*; // Diverse fil-klasser

import java.util.\*; // Spesielt Scanner-klassen

public class TellOrdlengder {

public static void main(String[] args) throws Exception {

final int MAX\_LENGDE = 12;

int[] antallAv = new int[MAX\_LENGDE];

String filnavn = showInputDialog("Gi filnavn:");

File fil = new File(filnavn);

Scanner leser = new Scanner(fil);

// Dersom det er norske bokstaver i teksten, og

// den er lagret i tegnsettet utf8, bruk:

// Scanner leser = new Scanner(fil, "utf8");

while ( leser.hasNextLine() ) {

String linje = leser.nextLine();

String[] ordliste = linje.split("( |\\.|,)+");

// Deler linjen i ord vha. mellomrom, punktum og komma.

for (int i=0; i<ordliste.length; i++) {

int ordlengde = ordliste[i].length();

if ( ordlengde < MAX\_LENGDE )

antallAv[ordlengde]++;

else

antallAv[0]++;

}

}

leser.close();

String utTxt = "RESULTAT" + "\n";

for (int i=1; i<MAX\_LENGDE; i++)

utTxt += "Ordlengde " + i + ": " + antallAv[i] + "\n";

utTxt += "Ordlengde >= " + MAX\_LENGDE + ": " + antallAv[0];

showMessageDialog(null, utTxt);

}

}

### Leser høydedata fra fil, jenter/gutter, median.

/\* Programmet leser høydedata fra fil, jentenes

\* høyde på første linje og guttenes på andre.

\* Både jentenes og guttenes medianhøyde beregnes

\* og skrives inn til slutt på samme fil.

\* Høyder er skilt av ett eller flere blanke tegn. \*/

import static java.lang.System.\*;

import static java.lang.Integer.\*;

import static javax.swing.JOptionPane.\*;

import java.io.\*;

import java.util.\*;

public class Median {

public static void main(String[] args) {

try {

String filnavn = showInputDialog("Gi filnavn:");

File fil = new File(filnavn);

Scanner leser = new Scanner(fil);

// Første linje inneholder jentenes høyder

String linje = leser.nextLine();

String[] dataTab = linje.split("( )+");

int[] jenteHøyder = new int[dataTab.length];

for (int i=0; i<dataTab.length; i++)

jenteHøyder[i] = parseInt(dataTab[i]);

// Andre linje inneholder guttenes høyder

linje = leser.nextLine();

dataTab = linje.split("( )+");

int[] gutteHøyder = new int[dataTab.length];

for (int i=0; i<dataTab.length; i++)

gutteHøyder[i] = parseInt(dataTab[i]);

leser.close();

FileWriter skrivelink = new FileWriter(filnavn, true);

PrintWriter skriver = new PrintWriter(skrivelink);

skriver.println("\n"+"Jentemedian: " + median(jenteHøyder)

+ "\n"+"Guttemedian: " + median(gutteHøyder) );

skriver.close();

}

catch (Exception e) {

out.println("Problemer med fildata: " + e.toString());

}

}

private static int median(int[] tab) {

int median = 0;

Arrays.sort(tab);

int midten = tab.length/2;

if ( tab.length % 2 == 1 )

median = tab[midten];

else

median = (tab[midten-1]+tab[midten])/2;

return median;

}

}

### Leser drivstoffsalg (4 ulike) fra fil for en enkelt dag. Skriver tilbake samlet omsetning.

/\* Leser drivstoffsalg (4 ulike typer) fra fil for en enkelt

\* dag. Beregner og skriver tilbake til den samme filen samlet

\* omsetning for hhv. bensin og diesel for denne dagen.

\* Evt. tomme linjer på filen blir bare hoppet over. \*/

import static java.lang.Integer.\*;

import static java.lang.Double.\*;

import static javax.swing.JOptionPane.\*;

import java.util.\*;

import java.io.\*;

public class Drivstoff {

public static void main(String[] args) {

String filnavn = showInputDialog("Gi filnavn:");

try {

// Inndatastrøm åpnes

File fil = new File(filnavn);

Scanner leser = new Scanner(fil);

double sumLB=0, sumLD=0, sumKrB=0, sumKrD=0;

while ( leser.hasNextLine() ) {

String linje = leser.nextLine();

linje = linje.trim();

// Fjerner tomme tegn foran og bak i tegnsekvensensen, og

// om den da blir tom så er den ikke en inndatalinje.

if ( linje.length() > 0 ) {

String[] data = linje.split("( )+");

// Skilletegn på filen er ett eller flere blanke tegn

// Splitter linjen i tre element: kode liter kr

char kode = data[0].toUpperCase().charAt(0);

double liter = parseDouble(data[1]);

double kr = parseDouble(data[2]);

if ( kode == 'L' || kode == 'H') {

sumLB += liter;

sumKrB += kr;

}

else { // koden er V eller F

sumLD += liter;

sumKrD += kr;

}

}

}

leser.close();

String ut = "\n" + "Omsetning" + "\n"

+ "Bensin: " + avrund(sumLB) + " liter "

+ avrund(sumKrB) + " kr " + "\n"

+ "Diesel: " + avrund(sumLD) + " liter "

+ avrund(sumKrD) + " kr " + "\n";

// Utdatastrøm åpnes

FileWriter skrivekobling = new FileWriter(fil, true);

// Her betyr Append=true innholdet skal legges til på slutten

PrintWriter skriver = new PrintWriter(skrivekobling);

skriver.println(ut);

skriver.close();

showMessageDialog(null, "Registrering vellykket!");

}

catch (FileNotFoundException e) {

showMessageDialog(null, "Problem, finner ikke filen!");

}

catch (NumberFormatException e) {

showMessageDialog(null, "Problem, innholdet kan ikke håndteres!");

}

catch (Exception e) {

showMessageDialog(null, "Uidentifisert problem!");

}

}

private static int avrund(double tall) {

// Hjelpemetoden avrunder til nærmeste heltall

return (int)(tall + 0.5);

}

}

### Leser personopplysninger fra fil, skriver ut navneliste sortert etter etternavn

/\* Leser personopplysninger fra en fil, der hver linje er på

\* formen: fornavn#etternavn#alder#yrke.

\* Programmet skriver ut en navneliste sortert på etternavn.

\* Forutsetter at alle inndatalinjene er korrekt oppsatt. \*/

import static javax.swing.JOptionPane.\*;

import java.util.\*;

import java.io.\*;

public class PersonOpplysninger {

public static void main(String[] args) {

String filnavn = showInputDialog("Gi filnavn:");

try {

File fil = new File(filnavn);

Scanner leser = new Scanner(fil);

String[] navneListe = new String[100]; // Nok plass?

int nr = 0;

while ( leser.hasNextLine() ) {

String linje = leser.nextLine();

String[] data = linje.split("#");

// Skilletegn på filen er ett grindtegn #

String fornavn = data[0];

String etternavn = data[1];

// alder og yrke er ikke aktuelle her!

navneListe[nr] = etternavn + ", " + fornavn;

nr++;

}

leser.close();

int antallNavn = nr;

Arrays.sort(navneListe, 0, antallNavn);

String utTxt = "Sortert navneliste:" + "\n";

for (int i=0; i<antallNavn; i++)

utTxt += navneListe[i] + "\n";

showMessageDialog(null, utTxt);

}

catch (Exception e) {

showMessageDialog(null, "Problem med filen!");

}

}

}

### Metoder for Person

public class Person {

String fornavn;

String etternavn;

int alder;

String yrke;

public Person(String fornavn, String etternavn,

int alder, String yrke) {

this.fornavn = fornavn;

this.etternavn = etternavn;

this.alder = alder;

this.yrke = yrke;

}

public String getNavn() {

return etternavn + ", " + fornavn;

}

public String toString() {

return getNavn() + " - " + alder + " - " + yrke;

}

}

### Leser personopplysninger fra fil, skriver ut liste sortert på etternavn av dem som har innlest alder.

/\* Leser personopplysninger fra en fil, der hver linje er på

\* formen: fornavn#etternavn#alder#yrke.

\* Programmet skriver ut en liste, sortert på etternavn,

\* over dem som har en innlest alder.

\* Forutsetter at alle inndatalinjene er korrekt oppsatt. \*/

import static javax.swing.JOptionPane.\*;

import static java.lang.Integer.\*;

import java.util.\*;

import java.io.\*;

public class PersonAlderInfo {

public static void main(String[] args) {

String filnavn = showInputDialog("Gi filnavn:");

try {

// Inndatastrøm åpnes

File fil = new File(filnavn);

Scanner leser = new Scanner(fil);

Person[] personListe = new Person[100];

int nr = 0;

while ( leser.hasNextLine() ) {

String linje = leser.nextLine();

String[] data = linje.split("#");

// Skilletegn på filen er ett grindtegn #

String fornavn = data[0];

String etternavn = data[1];

int alder = parseInt(data[2]);

String yrke = data[3];

personListe[nr] = new Person(fornavn, etternavn, alder, yrke);

nr++;

}

leser.close();

int antallNavn = nr;

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

int lestAlder = parseInt(tallTxt);

String utTxt = "Liste over dem som er " + lestAlder + " år:" + "\n";

for (int i=0; i<antallNavn; i++)

if ( personListe[i].alder == lestAlder )

utTxt += personListe[i].toString() + "\n";

showMessageDialog(null, utTxt);

}

catch (Exception e) {

showMessageDialog(null, "Problem med filen!");

}

}

}

### Statistiske metoder for innlesing av teksstrenger, heltall og kommatall via dialogvindu, og visning av meldinger via dialogvindu (ikke nødvendigvis pensum)

import java.lang.Double;

import java.lang.Integer;

import javax.swing.JOptionPane;

/\* Hjelpeklasse med statiske metoder for robust innlesing

\* av tekststrenger, heltall og kommatall via dialogvindu,

\* samt visning av meldinger via dialogvindu. \*/

public class Dialog {

public static void visMelding(String ledetekst) {

JOptionPane.showMessageDialog(null, ledetekst);

}

public static String lesString(String ledetekst) {

String innTxt = JOptionPane.showInputDialog(ledetekst);

return innTxt;

}

public static int lesInt(String ledetekst) {

int tall = 0;

boolean ok = true;

String feilmelding = "";

do {

String innTxt = JOptionPane.showInputDialog(feilmelding + ledetekst);

try {

tall = Integer.parseInt(innTxt);

ok = true;

}

catch (NumberFormatException e) {

ok = false;

feilmelding = "Ulovlig tallverdi! \n";

}

} while (!ok);

return tall;

}

public static double lesDouble(String ledetekst) {

double tall = 0;

boolean ok = true;

String feilmelding = "";

do {

String innTxt = JOptionPane.showInputDialog(feilmelding + ledetekst);

try {

tall = Double.parseDouble(innTxt);

ok = true;

}

catch (NumberFormatException e) {

ok = false;

feilmelding = "Ulovlig tallverdi! \n";

}

} while (!ok);

return tall;

}

}

### Registrering av skøytetider, lagret på fast tekstfil, registrert i to omganger. 2 dager Hvis tom dag 1.

import static java.lang.System.\*;

import java.lang.Double;

import java.lang.Integer;

import java.util.\*;

import java.io.\*;

/\* Hovedklasse for registrering av skøytetider.

\*

\* Resultatene blir lagret på en fast tekstfil og blir

\* registrert i to omganger (etter dag 1 og etter dag 2).

\*

\* Hvis datafilen er tom, så blir programmet kjørt for dag 1,

\* dvs. at nr, navn og tider for 500 og 5000 skal registreres.

\* Etter dag 2 skal data for dag 1 først leses inn fra fil,

\* og tider for 1500 og 10000 legges til.

\*

\* Etter dag 2 blir navn på vinnere skrevet ut, sammen med

\* navn, tider og poengsum for alle deltagere. \*/

public class Skoytelop {

// Skilletegn ved lagring til fil

private static String SKILLETEGN = ";";

// Navn på tekstfil der resultatene blir lagret.

private static String FILNAVN = "skoytelop.txt";

// Referansetabell for å ta vare på deltagere.

private static Deltager[] løperTab;

// Settes til true hvis datafilen er tom.

private static boolean dag1;

public static void main(String[] args) throws Exception {

lesInnFraFil();

lesInnResultat();

skrivTilFil();

if (!dag1) {

double best = bestePoengsum();

visResultater(best);

}

}

private static void lesInnFraFil() throws Exception {

// Inndatastrøm åpnes

File fil = new File(FILNAVN);

Scanner leser = new Scanner(fil);

// Første linje på filen inneholder antall deltagere.

// Hvis filen er tom, leses antall deltagere inn fra bruker.

int antall;

if (leser.hasNextLine()) {

antall = Integer.parseInt(leser.nextLine());

dag1 = false;

}

else {

antall = Dialog.lesInt("Antall deltagere:");

dag1 = true;

}

løperTab = new Deltager[antall];

// Les inn resultater for alle deltagere

int pos = 0;

while (leser.hasNextLine()) {

String linje = leser.nextLine();

String[] data = linje.split(SKILLETEGN);

int nr = Integer.parseInt(data[0]);

String navn = data[1];

double tid5H = Double.parseDouble(data[2]);

double tid5T = Double.parseDouble(data[3]);

double tid15H = Double.parseDouble(data[4]); // Settes til 0.0 etter dag 1

double tid10T = Double.parseDouble(data[5]); // Settes til 0.0 etter dag 1

Deltager d = new Deltager(nr, navn, tid5H, tid5T, tid15H, tid10T);

løperTab[pos] = d;

pos++;

}

leser.close();

}

public static void skrivTilFil() throws Exception {

PrintWriter skriver = new PrintWriter(FILNAVN);

// Skriv antall deltagere på første linje

int antall = løperTab.length;

skriver.println(antall);

// Skriv ut alle deltagere

for (int nr=0; nr<antall; nr++) {

Deltager d = løperTab[nr];

String linje = d.getNr() + SKILLETEGN

+ d.getNavn() + SKILLETEGN

+ d.getTid5H() + SKILLETEGN

+ d.getTid5T() + SKILLETEGN

+ d.getTid15H() + SKILLETEGN

+ d.getTid10T();

skriver.println(linje);

}

skriver.close();

}

private static void lesInnResultat() {

if (dag1) {

// Les inn nr, navn og tider for 500m og 5000m

for (int nr=0; nr<løperTab.length; nr++) {

løperTab[nr] = lesDeltagerDag1();

}

}

else {

// Dag 2: Oppdater med tider for 1500m og 10000m

for (int nr=0; nr<løperTab.length; nr++) {

oppdaterDeltagerDag2(løperTab[nr]);

}

}

}

private static Deltager lesDeltagerDag1() {

int nr = Dialog.lesInt("Deltagernr:");

String navn = Dialog.lesString("Navn:");

double tid5H = Dialog.lesDouble("Tid 500m:");

double tid5T = Dialog.lesDouble("Tid 5000m:");

// Opprett nytt Deltager-objekt.

Deltager d = new Deltager(nr, navn, tid5H, tid5T);

return d;

}

private static void oppdaterDeltagerDag2(Deltager d) {

int dnr = d.getNr();

double tid15H = Dialog.lesDouble("Tid 1500m for deltager " + dnr + ":");

double tid10T = Dialog.lesDouble("Tid 10000m for deltager " + dnr + ":");

// Oppdater Deltager-objekt

d.setTid15H(tid15H);

d.setTid10T(tid10T);

}

private static double bestePoengsum() {

double bestePoengsum = Double.MAX\_VALUE;

for (int nr=0; nr<løperTab.length; nr++) {

// Ta vare på hvis ny beste poengsum

if (løperTab[nr].poengsum() < bestePoengsum) {

bestePoengsum = løperTab[nr].poengsum();

}

}

return bestePoengsum;

}

private static void visResultater(double best) {

String vinnere = "Vinnere:\n";

for (int lnr=0; lnr<løperTab.length; lnr++) {

// Skriv ut navn, tider og poengsum for alle

out.println(løperTab[lnr].toString());

// Ta vare på navn til de som har best poengsum

if (løperTab[lnr].poengsum() == best) {

vinnere += løperTab[lnr].getNavn() + " ";

}

}

out.println(vinnere);

}

}

### Metoder for neste program

public class Rektangel {

int x; // (x,y) er posisjon for

int y; // øvre venstre hjørne

int b; // rektangelets bredde

int h; // rektangelets høyde

public Rektangel(int x, int y, int b, int h ) {

this.x = x; this.y = y;

this.b = b; this.h = h;

}

public void størrelse(int b, int h) {

this.b = b; this.h = h;

}

public void flytt(int dx, int dy) {

this.x += dx; this.y += dy;

}

public double omkrets() {

return 2\*b + 2\*h;

}

public String vis() {

return "("+x+","+y+") - B="+b+", H="+h;

}

}

### Easygraphics - Leser data om rektangler fra en fil, opprettes i tabell, tegnes på en grafisk flate

/\* Programmet leser data om rektangler fra en

\* skilletegnseparert fil. Rektanglene opprettes

\* i en tabell, og tegnes ut på en grafisk flate. \*/

import static java.lang.Integer.\*;

import static java.lang.System.\*;

import java.io.\*;

import java.util.\*;

public class RektangelTegning extends EasyGraphics {

public static void main(String[] args) {

launch(args);

}

public void run() {

makeWindow("Rektangeltegning", 900, 500);

// Antar at det ikke er flere enn 50

// rektangler på den aktuelle filen.

Rektangel[] r = new Rektangel[20];

int i = 0;

try {

String filnavn = "rektangler.txt";

File fil = new File(filnavn);

Scanner leser = new Scanner(fil);

while ( leser.hasNextLine() ) {

String linje = leser.nextLine();

String[] data = linje.split("#");

int x = parseInt(data[0]);

int y = parseInt(data[1]);

int b = parseInt(data[2]);

int h = parseInt(data[3]);

r[i++] = new Rektangel(x,y,b,h);

}

leser.close();

int antRek = i;

for (i=0; i<antRek; i++) {

int rv = trekk(0,255);

int gv = trekk(0,255);

int bv = trekk(0,255);

setColor(rv,gv,bv);

fillRectangle(r[i].x,r[i].y,r[i].b,r[i].h);

pause(50);

}

}

catch ( Exception e) {

out.println("Problem med filen");

}

}

// Trekker et tilfeldig heltall i området: fra-til.

public static int trekk(int fra, int til) {

return (int)( Math.random()\*(til-fra+1) ) + fra;

}

}

## Leksjon 6 – Roy

### Tester permanent lagring på fil ved å skrive navneliste til en sekvensiell tekstfil

/\* Leksjon 6/Kap.7, jobb 1 a): Testing av filer

\* Utviklet av RMI - November 2017

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

\* Programmet tester permanenet lagring på fil ved å skrive

\* data (navneliste) til en sekvensiell tekstfil.

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

import java.io.\*; // Trenger PrintWriter

import static javax.swing.JOptionPane.\*;

public class Filtest {

public static void main(String[] args) throws Exception {

// Lager ei navneliste (for å slippe innlesing fra brukeren her)

String[] navn = {"Ole Olsen", "Ida Nilsen", "Ola Normann", "Medel Swenson", "Anne By"};

// Skriver navn til fil

// Kan gjøre det rett fram slik:

// PrintWriter skriver = new PrintWriter("navn.txt");

// Et bedre alternativ:

String filnavn = "navn.txt";

PrintWriter skriver = new PrintWriter(filnavn);

for (int i=0; i<navn.length; i++)

skriver.println(navn[i]);

// Legger ut ett navn pr linje på tekstfilen

skriver.close(); // "Lukker filen" - viktig for ikke å miste data

// Skriver ut kvittering til brukeren om at filskrivingen er ferdig

String utTekst = "Filutskrift er ferdig, ok!";

showMessageDialog(null, utTekst);

}

}

### Leser fil med navn, åpner ny fil og skriver navnene i omvendt rekkefølge, leser inn og skriver nytt navn nederst på den opprinnelige filen

/\* Leksjon 6/Kap.7, jobb 1 a): Testing av filer

\* Utviklet av RMI - November 2017

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

\* Programmet tester permanent lagring på fil ved å lese

\* inn data (navn) fra en sekvensiell tekstfil som har ett

\* navn pr tekstlinje. Antall navn er ukjent. NB! Navnefilen

\* må allerede ligge i samme mappe som kjørende program.

\* Programmet bygger opp ei navneliste i hurtigminnet i form

\* av en String-tabell.

\* En ny fil åpnes og navnene i tabellen skrives i omvendt

\* rekkefølgen på den nye filen. Filen lukkes.

\* Til slutt leses ett navn fra bruker, og dette legges til

\* bakerst/nederst på den opprinnelige navnefilen.

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

import java.io.\*;

import java.util.\*;

import static javax.swing.JOptionPane.\*;

public class FiltestV2 {

public static void main(String[] args) throws Exception {

// Vi kan opprette en "stor" nok tabell i programmet, og

// så fylle inn i den så lenge det er flere navn på filen.

// String[] navn = new String[100];

// Et bedre alternativ er å bestemme eksakt rett størrelse

// på tabellen som skal inneholde navnene.

// Åpner fil for lesing

String filnavn = "navn.txt";

File fil = new File(filnavn); // Filinnpakning av "tekst"

Scanner leser = new Scanner(fil);

// Teller antall navn på filen

int antallNavn = 0; // Teller antall linjer -> antall navn

// Fortsetter så lenge det er tekst på linjer

while ( leser.hasNextLine() ) {

String linje = leser.nextLine();

antallNavn++;

}

// Lukker fil etter tellelesing

leser.close();

// Oppretter tabellen

String[] navn = new String[antallNavn];

// Henter data fra fil og legger i en tabell i hurtigminnet

// Åpner filen på nytt

leser = new Scanner(fil);

int i=0; // Sjekker om det er flere tekstlinjer

while ( leser.hasNextLine() ) {

// Henter inn neste tekstlinje

navn[i] = leser.nextLine();

i++;

}

int antNavn = i;

// Lukker fil etter navnelesing

leser.close();

// Skriver navn i omvendt rekkefølge på en ny fil

PrintWriter skriver = new PrintWriter("snu"+filnavn);

for(int j=antNavn-1; j>=0; j--)

skriver.println(navn[j]);

skriver.close();

// Leser inn et nytt navn og skriver dette på den opprinnelige filen.

String nyttNavn = showInputDialog("Gi navn (Fornavn Etternavn):");

// Henger på en linje til filen 'navn.txt' // Append = true

FileWriter skrivekobling = new FileWriter(filnavn, true);

skriver = new PrintWriter(skrivekobling);

skriver.println(nyttNavn);

skriver.close();

// Skriver ut kvittering til brukeren om at filskrivingen er ferdig

String utTekst = "Filutskrift er ferdig, ok!";

showMessageDialog(null, utTekst);

}

}

### Leser inn fil med fornavn,etternavn, skriver ut til fil med etternavn,fornavn

/\* Leksjon 6, jobb 1 b): Navneliste

\* Utviklet av RMI - November 2017

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

\* Alternativ 1: Programmet leser inn navn på formen

\* 'Fornavn Etternavn' fra fil, og legger dem på en ny

\* fil i samme rekkefølge som: 'Etternavn, Fornavn'.

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

\* Alternativ 2: La navn på 'fornavn.txt' ha ett eller

\* flere fornavn, f.eks. Jens Kåre Brage Olav Hansen,

\* og da snus navnet til: Hansen, Jens Kåre Brage Olav

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

import java.io.\*;

import java.util.\*;

public class NavnefilTest {

public static void main(String[] args) throws Exception {

String filnavn = "fornavn.txt";

File fil = new File(filnavn);

Scanner leser = new Scanner(fil);

PrintWriter skriver = new PrintWriter("etternavn.txt");

while ( leser.hasNextLine() ) {

String navn = leser.nextLine();

Scanner navneleser = new Scanner(navn);

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

// Alternativ 1: Kun ett fornavn, og kun ett etternavn

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

// String fornavn = navneleser.next();

// String etternavn = navneleser.next();

// skriver.println(etternavn + ", " + fornavn);

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

// Alternativ 2: Ett eller flere fornavn, kun ett etternavn

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

String[] navneTab = new String[10];

int n=0; // Teller antall navneelement på linja

while ( navneleser.hasNext() ) {

navneTab[n] = navneleser.next();

// Legger hvert navneelement inn i en tabell

n++; // Teller opp antall navn ett hakk

}

// Legger etternavn først i snudd versjon av navnet

String snuNavn = navneTab[n-1] + ", ";

// Henger på alle fornavn i den rekkefølge de stod

for (int i=0; i<=n-2; i++)

snuNavn += navneTab[i] + " ";

skriver.println(snuNavn);

}

leser.close();

skriver.close();

}

}

### Leser desimaltall fra fil, avrunder til heltall som skrives på ny fil.

/\* Leksjon 6, jobb 2: Tallanalyse pkt a)

\* Utviklet av RMI - November 2017

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

\* Programmet leser desimaltall fra fil, avrunder til

\* heltall som skrives på ny fil - med maks fire tall

\* på hver linje.

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

\* På tekstfil med fastfont, kan vi skrive verdiene i

\* høyrejusterte kolonner med gitt bredde på kolonna.

\* De to 'skrivPent'-metodene her fungerer ikke særlig

\* bra i dialogvindu med proporsjonalfont.

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

import java.io.\*;

import java.util.\*;

public class Tallanalyse {

public static void main(String[] args) throws Exception {

String filnavn = "desimal.txt";

File fil = new File(filnavn);

Scanner leser = new Scanner(fil);

leser.useLocale(Locale.US); // Endrer fra komma til punktum

PrintWriter skriver = new PrintWriter("heltall.txt");

int antall = 0;

while ( leser.hasNextDouble() ) {

double des = leser.nextDouble();

int hel = (int)Math.round(des);

// Ulike utskriftsalternativ:

// skriver.print(hel + " ");

// skriver.print( skrivPent(hel) );

skriver.print( skrivPent2(hel, 7) );

// Skriver heltall i kolonner med bredde 7

antall++;

if ( antall % 4 == 0 )

skriver.println();

// Linjeskift etter hvert 4. tall

}

leser.close();

skriver.close();

} // Slutt på main

// Hjelpemetoder til bruk i dette programmet

private static String skrivPent(int tall) {

String ut = "";

if ( tall < 10 ) ut += " ";

if ( tall < 100 ) ut += " ";

if ( tall < 1000 ) ut += " ";

return ut+tall;

} // Slutt på skrivPent

# Eksamensoppgaver

## Hjelpemetode, teller antall siffer i et heltall, hvis positivt

/\* Eksamensoppgave 1d)

\* Hjelpemetode som teller antall siffer i et

\* heltall - forutsatt at det er positivt.

\*/

private static int antallSiffer(int tall) {

// tall = Math.abs(tall);

// Denne vil sikre korrekt siffertelling

// også for negative tall.

int ant = 0;

do {

ant++; // Alle tall har minst ett siffer

tall = tall / 10;

} while ( tall > 0);

return ant;

} // Slutt på antallSiffer

private static String skrivPent2(int tall, int bredde) {

String ut = "";

int antBlanke = bredde - antallSiffer(tall);

for (int i=0; i<antBlanke; i++)

ut += " ";

return ut+tall;

} // Slutt på skrivPent2

}

## Leser informasjon fra en telstfil med produktgruppenummer og sekundantall for tlf-samtale. Lagres i to tabeller, en teller antall tlf-samtaler for hvert gr.nr og en summerer tlf-tid for hvert gr.nr. Skriver til ny fil gjennomsnittlig tlf-tid i sekund for hvert gr.nr.

/\* Leksjon 6/ Kap.7: Jobb 3 (Eksamen 03.05.2011)

\* Utviklet av RMI - November 2017

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

\* Programmet leser informasjon fra en tekstfil (datalogg) som består av

\* et produktgruppenr og et sekundantall for tlf-samtale angående dette

\* produktgruppenummeret. Leste data lagres i to tabeller, en som teller

\* antall tlf-samtaler for hvert gr.nr. og en som summerer total tlf-tid

\* for hvert gr.nr.

\* Programmet beregner og skriver til ny fil gjennomsnittlig tlf-tid i

\* sekund for hvert gr.nr.

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

import static javax.swing.JOptionPane.\*;

import java.util.\*; // Scanner

import java.io.\*; // File og PrintWriter

public class Telefonbetjening {

public static void main(String[] args) throws Exception {

// Oppretter to telletabeller

final int ANT\_GRUPPER = 9;

// Kan indeksere 1-8 som produktgruppenummer, 0 er ubrukt

int[] antTlfsamtaler = new int[ANT\_GRUPPER];

int[] antTlfsekund = new int[ANT\_GRUPPER];

// Lesing fra fil (filnavn: ukedag.txt)

String filnavn = showInputDialog("Gi filnavn: ");

File fil = new File(filnavn);

Scanner leser = new Scanner(fil);

while ( leser.hasNextLine() ) {

String linje = leser.nextLine();

// Splitter inndatalinje i gruppenr og tlftid

Scanner linjeleser = new Scanner(linje);

linjeleser.useDelimiter("-");

// Skilletegn '-' mellom grNr og tlfTid på fil

int grpNr = linjeleser.nextInt();

int sekund = linjeleser.nextInt();

antTlfsamtaler[grpNr]++;

antTlfsekund[grpNr] += sekund;

}

leser.close();

// Overfører telleresultatene til fil

// 1) Åpne (ny) fil

PrintWriter skriver = new PrintWriter("res" + filnavn);

// 2) Overføre data til fil

for (int grpNr=1; grpNr<ANT\_GRUPPER; grpNr++)

if ( antTlfsamtaler[grpNr] > 0 ){

int snittTlftid = antTlfsekund[grpNr] / antTlfsamtaler[grpNr];

skriver.println(grpNr + ": " + snittTlftid);

}

else // kan ikke dele på 0 !

skriver.println(grpNr + ": 0");

// 3) Lukke fil

skriver.close();

String utTekst = "Resultat til fil er OK!";

showMessageDialog(null,utTekst);

}

}

# Min kode

## Obligatorisk oppgave

### Bankinfo (Main)

/\* Leksjon 4: Obligatorisk oppgave (Bankinformasjon)

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

\* Levert av: 216756, Robert Anholt

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

\* Programmet bruker verktøyklassen 'Hjelp' som inneholder relevante

\* metoder (innlesing, avrunding) og konstanter (kurs og meny).

\* NB! Filen lagres i mappen sammen med Hjelp, som: BankInfo.java

\*/

// Import av nødvendige bibliotekressurser

import static java.lang.Integer.\*;

import static javax.swing.JOptionPane.\*;

public class BankInfo {

public static void main(String[] args) {

// Variabel for programgjentak

boolean fortsett = true;

do {

// Leser inn ønsket info-tjeneste fra en meny

int valg = parseInt(showInputDialog(Hjelp.MENY) );

if (valg == 0)

fortsett = false;

else

switch ( valg ) {

case 1: veksling(); break;

case 2: sparing(); break;

case 3: låning(); break;

default: feilmelding(); break;

}

} while ( fortsett );

} // slutt på main-metoden

/\* Hjelpemetode for å gi info om valutaveksling fra et beløp

\* i USdollar til norske kroner ved en fastsatt valutakurs.

\*/

private static void veksling() {

// Les inn et dollarbeløp via kall på hjelpe-metode.

// Hvis lovlig beløp

// beregn beløp i norske kroner

// skriv ut beløpene avrundet til 2 siffer etter komma

// skriv ut vekslingskursen

// Skriv feilmelding ved ulovlig beløp

double valuta;

do {

valuta = Hjelp.lesInn("Gi et dollarbeløp:");

if (valuta < 0)

showMessageDialog(null, "Ulovlig beløp!");

} while (valuta < 0);

double bUSD = Hjelp.avrund(valuta);

double bNOK = Hjelp.avrund(Hjelp.USD\_NKR\*bUSD);

showMessageDialog(null, "1: Veksling" + "\n"

+ "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" + "\n"

+ "Beløp i USD: " + bUSD + "\n"

+ "Kurs: 1 USD = " + Hjelp.USD\_NKR + " NOK" + "\n"

+ "Beløp i NOK: " + bNOK + "\n"

+ "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

}

/\* Hjelpemetode for å gi info om sparing med ett innskud som

\* følges over et gitt antall år til en gitt årsrente (%).

\*/

private static void sparing() {

String utTekst = "";

// Les inn via kall på hjelpe-metode: et sparebeløp, en årsrente

double sparebeløp;

do {

sparebeløp = Hjelp.lesInn("Gi sparebeløp:");

if (sparebeløp < 0)

showMessageDialog(null, "Ulovlig Sparebeløp!");

} while (sparebeløp < 0);

double årsrente;

do {

årsrente = Hjelp.lesInn("Gi årsrente:");

if (årsrente < 0)

showMessageDialog(null, "Ulovlig Årsrente!");

} while (årsrente < 0);

// og et antall år i spareperioden.

int antallår;

do {

antallår = (int)Hjelp.lesInn("Gi antall år:");

if (antallår < 0)

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

} while (antallår < 0);

// Hvis lovlige inndata

// skriv ut (kvitter) de aktuelle inndataene

sparebeløp = Hjelp.avrund(sparebeløp);

// beregn år for år: påløpt rente og ny saldo (økt beløp)

utTekst += ( "2: Sparing" + "\n"

+ "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" + "\n"

+"Sparebeløp: " + sparebeløp + "\n"

+ "Årlig rente: " + årsrente + "%" + "\n"

+ "Antall år: " + antallår + "\n"

+ "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" + "\n"

+ "År Rente Saldo" + "\n");

double rente ;

double saldo = sparebeløp;

for(int i = 1; i<=antallår; i++){

rente = saldo\*årsrente/100;

saldo += rente;

// hvis mindre enn eller lik 10 år

// skriv ut år, renter og saldo

rente = Hjelp.avrund(rente);

saldo = Hjelp.avrund(saldo);

if (i <=10)

utTekst += (i + ": " + rente + " " + saldo + "\n");

// Etter beregningene skrives sluttsaldo ut hvis år er over 10

if (antallår > 10 && i == antallår)

utTekst += ("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" + "\n"

+ "Saldo etter " + antallår + " År:" + " " + saldo);

}

showMessageDialog(null, utTekst);

// Skriv feilmelding ved ulovlige inndata

}

/\* Hjelpemetode for å gi info om annuitetslån, dvs. et fast årlig

\* beløp for å betale et lån til en gitt rente over et antall år.

\*/

private static void låning() {

String utTekst = "";

// Les inn via kall på hjelpe-metode: et lånebeløp, en årsrente

double lånebeløp;

do {

lånebeløp = Hjelp.lesInn("Gi lånebeløp:");

if (lånebeløp < 0)

showMessageDialog(null, "Ulovlig lånebeløp!");

} while (lånebeløp < 0);

double årsrente;

do {

årsrente = Hjelp.lesInn("Gi årsrente:");

if (årsrente < 0)

showMessageDialog(null, "Ulovlig årsrente!");

} while (årsrente < 0);

// og et antall år for nedbetalingsperioden.

int antallår;

do {

antallår = (int)Hjelp.lesInn("Gi antall år for nedbetaling:");

if (antallår < 0)

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

} while (antallår < 0);

// Hvis lovlige inndata:

// skriv ut (kvitter) de aktuelle inndataene

utTekst = ("3: Lån" + "\n"

+ "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" + "\n"

+ "Lånebeløp: " + lånebeløp + "\n"

+ "Årlig rente: " + årsrente + "%" + "\n"

+ "Antall år: " + antallår + "\n"

+ "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" + "\n");

// beregn årlig betaling (nettsøk: formel for annuitetslån)

// bruk matematiske metoder (Math) til regnearbeidet

double rente = lånebeløp \* årsrente;

double eksponent = (1- 1 / Math.pow(1 + årsrente/100, antallår));

double betaling = rente / eksponent /100;

betaling = Hjelp.avrund(betaling);

// skriv ut årlig betaling

utTekst += ("Årlig betaling: " + betaling);

showMessageDialog(null, utTekst);

// Skriv feilmelding ved ulovlige inndata

}

/\* Hjelpemetode for å gi en standard feilmelding i situasjonene

\* som er aktuelle i dette programmet.

\*/

private static void feilmelding() {

// Skriv en generell feilmelding som kan brukes overalt her

showMessageDialog(null, "404: Brain not found!");

}

}

### Metoder for Bankinfo

/\* Leksjon 4: Obligatorisk oppgave (Bankinformasjon)

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

\* Levert av: 216756, Robert Anholt

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

\* Verktøyklasse med hjelpemetoder til Bank-programmet.

\* NB! Filen lagres i mappen med BankInfo, som: Hjelp.java

\*/

// Evt. import av nødvendige bibliotekressurser

import static javax.swing.JOptionPane.\*;

import static java.lang.Double.\*;

public class Hjelp {

// Dagens valutakurs settes utenfor bankprogrammet.

// final -> konstant

// static -> brukes uten objekt: Klasse.KONSTANT

final static double USD\_NKR = 8.28; // Må endre verdi

// Viser aktuelle brukervalg via en tekstkonstant.

final static String MENY = "Konkus bank" + "\n"

+ "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" + "\n"

+ "Her får du info om:" + "\n"

+ "[1] Valutaveksling" + "\n"

+ "[2] Sparing" + "\n"

+ "[3] Lån" + "\n"

+ "[0] Avslutt" + "\n"

+ "Skriv inn tjenestekode (0-3): "; // Må endres

public static double lesInn(String ledetekst) {

// Metode som ber om og leser inn en desimalverdi fra

// brukeren, presentert ved 'ledetekst' i inndialogvindu.

String tallTekst = showInputDialog(ledetekst);

double tall = parseDouble(tallTekst);

return tall;

}

public static double avrund(double tall) {

// Metode som avrunder et desimaltall til to siffer

// etter komma. Se læreboka side 58.

return (int)(tall\*100 + 0.5)/100.0;

}

}

# Fra internett

## Finne det tredje største tallet i en array. Første er en metode.

**public** **class** ThirdLargestInArrayExample{

**public** **static** **int** getThirdLargest(**int**[] a, **int** total){

**int** temp;

**for** (**int** i = 0; i < total; i++){

**for** (**int** j = i + 1; j < total; j++){

**if** (a[i] > a[j]){

                    temp = a[i];

                    a[i] = a[j];

                    a[j] = temp;

                }

            }

        }

**return** a[total-3];  // Endre denne basert på hvilket tall du skal finne. Ned til -1 for største tall!

}

**public** **static** **void** main(String args[]){

**int** a[]={1,2,5,6,3,2};

**int** b[]={44,66,99,77,33,22,55};

System.out.println("Third Largest: "+getThirdLargest(a,6));

System.out.println("Third Largest: "+getThirdLargest(b,7));

}}

## Finne det nest minste tallet i en array

**public** **class** SecondSmallestInArrayExample{

**public** **static** **int** getSecondSmallest(**int**[] a, **int** total){

**int** temp;

**for** (**int** i = 0; i < total; i++){

**for** (**int** j = i + 1; j < total; j++){

**if** (a[i] > a[j]){

                    temp = a[i];

                    a[i] = a[j];

                    a[j] = temp;

                }

            }

        }

**return** a[1];//2nd element because index starts from 0  // For å finne det minste tallet skal «1» være null(0), tredje minste «2» osv.

}

**public** **static** **void** main(String args[]){

**int** a[]={1,2,5,6,3,2};

**int** b[]={44,66,99,77,33,22,55};

System.out.println("Second smallest: "+getSecondSmallest(a,6));

System.out.println("Second smallest: "+getSecondSmallest(b,7));

}}

## Fjerne «duplicate element» i en array.

**public** **class** RemoveDuplicateInArrayExample{

**public** **static** **int** removeDuplicateElements(**int** arr[], **int** n){

**if** (n==0 || n==1){

**return** n;

        }

**int**[] temp = **new** **int**[n];

**int** j = 0;

**for** (**int** i=0; i<n-1; i++){

**if** (arr[i] != arr[i+1]){

                temp[j++] = arr[i];

            }

         }

        temp[j++] = arr[n-1];

        // Changing original array

**for** (**int** i=0; i<j; i++){

            arr[i] = temp[i];

        }

**return** j;

    }

**public** **static** **void** main (String[] args) {

**int** arr[] = {10,20,20,30,30,40,50,50};

**int** length = arr.length;

        length = removeDuplicateElements(arr, length);

        //printing array elements

**for** (**int** i=0; i<length; i++)

           System.out.print(arr[i]+" ");

    }

}

## Plusse to matriser med tre kolonner

**public** **class** MatrixAdditionExample{

**public** **static** **void** main(String args[]){

//creating two matrices

**int** a[][]={{1,3,4},{2,4,3},{3,4,5}};

**int** b[][]={{1,3,4},{2,4,3},{1,2,4}};

//creating another matrix to store the sum of two matrices

**int** c[][]=**new** **int**[3][3];  //3 rows and 3 columns

//adding and printing addition of 2 matrices

**for**(**int** i=0;i<3;i++){

**for**(**int** j=0;j<3;j++){

c[i][j]=a[i][j]+b[i][j];    //use - for subtraction, \* for multiplication and so on,.

System.out.print(c[i][j]+" ");

}

System.out.println();//new line

}

}}

## Print Odd and even numbers from array

**public** **class** OddEvenInArrayExample{

**public** **static** **void** main(String args[]){

**int** a[]={1,2,5,6,3,2};

System.out.println("Odd Numbers:");

**for**(**int** i=0;i<a.length;i++){

**if**(a[i]%2!=0){

System.out.println(a[i]);

}

}

System.out.println("Even Numbers:");

**for**(**int** i=0;i<a.length;i++){

**if**(a[i]%2==0){

System.out.println(a[i]);

}

}

}}