# DAT100

# Grunnleggende Programmering - H14

## Øving 6

### Gruppedeltakarar:

|  |  |
| --- | --- |
| Navn | Studentnr |
| Jonas Holme | 146176 |
| Nils Terje Krumsvik | 507017 |
| Martin Anaton Robertsen | 146167 |
| Mounir Idrissi Jazouli | 145174 |

Dato: 09.10.14

# Oppg 1

## Beskrivelse:

Her har vi ein main-metode som kallar på dei tre metodane i kvar si klasse. I den første metoden skriv vi inn eit tilfeldig ord eller setning. Deretter har vi brukt StringBuffer-klassen, med kommandoen .reverse() for å skrive tekststrengen inn i ein ny streng baklengs.

I neste metode skriv vi inn tal frå tastatur, og ved hjelp av ei for-løkke blir dei sett inn i ein tabell. Vidare sjekkar vi med ei ny for-løkke om kvart tal er mindre enn det neste. Dersom det ikkje stemmer, vil ein boolsk variabel bli satt til false. Kva utskrift vi får, kjem an på om den er true eller false.

I den siste metoden går vi fram mykje på same måten som den førre, med å føre inn fem ord i ein tabell. Så har vi brukt Arrays.sort til å sortere strengtabellen alfabetisk. Vi skriv deretter ut det første ordet i tabellen.

### Kode:

**package** øving6;

**import** java.util.Arrays;

**import** java.util.Scanner;

**import** easyIO.Out;

**public** **class** Oppgave1 {

**public** **static** **void** main(String[] args) {

baklengs word = **new** baklengs();

word.lesBaklengs();

array streng = **new** array();

streng.stigendeArray();

alfSort ord = **new** alfSort();

ord.sorterFørste();

}

}

**class** baklengs {

Scanner tastatur = **new** Scanner(System.***in***);

Out skjerm = **new** Out();

**void** lesBaklengs() {

skjerm.out("Tast inn ord/setning: ");

String word = tastatur.nextLine();

String reverse = **new** StringBuffer(word).reverse().toString();

skjerm.outln("Original String: " + word);

skjerm.outln("Reversed String: " + reverse);

}

}

**class** array {

Scanner tastatur = **new** Scanner(System.***in***);

Out skjerm = **new** Out();

**void** stigendeArray() {

**int** heltall[] = **new** **int**[5];

**boolean** sort = **true**;

skjerm.outln(" ");

skjerm.outln("Tast inn en array på 5 tall: ");

**for** (**int** i = 0; i <= 4; i++){

heltall[i] = tastatur.nextInt();

}

**for** (**int** i = 0; i < heltall.length - 1 && sort; i++){

**if** (heltall[i] > heltall[i + 1]){

sort = **false**;

}

}

skjerm.outln();

**if**(sort){

skjerm.outln("Tabellen er sortert i stigende rekkefølge");

} **else**{

skjerm.outln("Tabellen er ikke sortert i stigende rekkefølge");

}

}

}

**class** alfSort {

Scanner tastatur = **new** Scanner(System.***in***);

Out skjerm = **new** Out();

**void** sorterFørste() {

String ord[] = **new** String[5];

skjerm.outln(" ");

skjerm.outln("Tast inn en array med 5 ord: ");

**for** (**char** i = 0; i <= 4; i++){

ord[i] = tastatur.nextLine();

}

Arrays.*sort*(ord);

skjerm.outln(" ");

skjerm.outln("Første ordet alfabetisk er: " + ord[0]);

}

}

## Eksempel på Utskrift:

Tast inn ord/setning: kjøttkaker

Original String: kjøttkaker

Reversed String: rekakttøjk

Tast inn en array på 5 tall:

4

6

9

11

8

Tabellen er ikke sortert i stigende rekkefølge

Tast inn en array med 5 ord:

eple

pære

banan

kiwi

appelsin

Første ordet alfabetisk er: appelsin

# Oppg 2

## Beskrivelse:

Her har vi importert java.util.Random for å generere tilfeldige tal mellom 0 og 1. Desse tala blir sett inn i tabellen en etter en ved hjelp av ei for-løkke. Når tabellen er fylt opp, brukar vi System-klassen til å finne ei starttid i millisekundar. Deretter brukar vi Arrays-klassen til å sortere tala i tabellen i stigande rekkefølge. Når det er ferdig, setter vi ei sluttid, då kan vi trekke starttida frå sluttida, og få antal millisekund sjølve sorteringa tok. Denne varigheten skriv vi så ut. Programmet er lite, og vi kjenner ikkje til nokon enklare måte å gjere det på. Vi finner heller ingen svakhet i programmet.

### Kode:

**package** øving6;

**import** java.util.Arrays;

**import** java.util.Random;

**public** **class** Oppg2 {

**public** **static** **void** main(String[] args) {

**double**[] tabell = **new** **double**[1000000];

Random tilfeldigTall = **new** Random();

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

tabell[i] = tilfeldigTall.nextDouble();

**long** startTid = System.*currentTimeMillis*();

Arrays.*sort*(tabell);

**long** sluttTid = System.*currentTimeMillis*();

**long** varighet = (sluttTid - startTid);

System.***out***.print("Sorteringa tok " + varighet + " millisekunder");

}

}

## Eksempel på Utskrift:

Sorteringa tok 334 millisekunder

Dersom vi setter av 10000000 (ein ekstra null) plassar i arrayen istadenfor, tek det mykje lenger tid:

Sorteringa tok 2289 millisekunder

# Oppg 3

## Beskrivelse:

Her brukar vi igjen java.util.Random til å generere eit tal mellom 1-6. Vi brukar ei for-løkke for å simulere og skrive ut 100 terningkast, og skriv dei ut med litt mellomrom. Vi har også lagt inn ei while-løkke for å ordne med linjeskift etter kvart tjuande kast.

For å telle kor mange vi får av kvar verdi, har vi deklarert seks tellarar, som ved hjelp av if-løkker aukar dersom deira terningverdi blir generert. Denne løysinga er ikkje akkurat plassbesparande og sikkert noko tungvint, men det var den måten vi fekk det til på, og den fungerar greit. Vi skriv ut alle tellarane.

For å finne gjennomsnittsverdien lagar vi ei ny for-løkke som summerar kvart kast med ein ny double-variabel «sum». Deretter tek vi summen delt på 100, og ved hjelp av easyIO skriv vi den ut med tre desimalar.

For å finne ut kor mange kast vi trengte før den første seksaren kom, så satte vi ein int-variabel «kastForÅFåSeks» til 1. Vi satte ei if-løkke inni ei for-løkke, og dersom neste kast ikkje blir seks, så aukar variabelen med ein. Finn vi ein seksar, stoppar vi for-løkka, og skriv ut «kastForÅFåSeks-variabelen.

For å finne ut kva terningverdi det var flest av, setter vi inn tellarane frå tidlegare inn i ein eigen tabell. Vi opprettar ein ny «mest» variabel som i ei for-løkke skal settast lik den største verdien i tabellen. Deretter brukar vi ei if-løkke for å sjå kva slags tellar «mest»-variabelen er lik på, og skriv ut deretter. Denne metoden er nok også ganske så tungvint å bruke. Den vil heller ikkje reagere dersom dei to største verdiane er like store. I så tilfelle vil den skrive ut den første av dei.

### Kode:

**package** øving6;

**import** easyIO.\*;

**import** java.util.Random;

**public** **class** Oppg3 {

**public** **static** **void** main(String[] args) {

Out skjerm = **new** Out();

**final** **int** BREDDE = 58;

skjerm.outln("TERNINGKASTSIMULATOR", BREDDE, Out.***CENTER***);

**int** linje = 19;

**int** tellerEn = 0;

**int** tellerTo = 0;

**int** tellerTre = 0;

**int** tellerFire = 0;

**int** tellerFem = 0;

**int** tellerSeks = 0;

Random terning = **new** Random();

**int**[] terningArray;

terningArray = **new** **int**[100];

// Utskrift av terningkast:

**for** (**int** i = 0; i < 100; i++) {

**int** kast = terning.nextInt(6) + 1;

skjerm.out(kast, 3);

terningArray[i] = kast;

// Linjeskift:

**while** (i == linje) {

skjerm.outln(" ");

linje = linje + 20;

}

// Antall av kvart tal:

**if** (terningArray[i] == 1)

tellerEn++;

**else** **if** (terningArray[i] == 2)

tellerTo++;

**else** **if** (terningArray[i] == 3)

tellerTre++;

**else** **if** (terningArray[i] == 4)

tellerFire++;

**else** **if** (terningArray[i] == 5)

tellerFem++;

**else** **if** (terningArray[i] == 6)

tellerSeks++;

**else**

skjerm.outln("Noko gjekk gale!");

}

skjerm.outln(" ");

skjerm.outln("Antall kast: 100");

skjerm.outln("Antall 6-arar: " + tellerSeks);

skjerm.outln("Antall 5-arar: " + tellerFem);

skjerm.outln("Antall 4-arar: " + tellerFire);

skjerm.outln("Antall 3-arar: " + tellerTre);

skjerm.outln("Antall 2-arar: " + tellerTo);

skjerm.outln("Antall 1-arar: " + tellerEn);

skjerm.outln(" ");

// Finne gjennomsnittskast:

**double** sum = 0;

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

sum = sum + terningArray[j];

}

skjerm.out("Gjennomsnittskast: ");

skjerm.outln(sum / 100, 3);

skjerm.outln(" ");

// Antal kast for å få den første seksaren

**int** kastForÅFåSeks = 1;

**for** (**int** k = 0; k < terningArray.length; k++) {

**if** (terningArray[k] == 6)

k = terningArray.length;

**else**

kastForÅFåSeks++;

}

skjerm.outln("Antall kast for å få den første 6-aren: "

+ kastForÅFåSeks);

skjerm.outln(" ");

// Terningsverdien det var flest av

**int**[] antallNummer = { tellerEn, tellerTo, tellerTre, tellerFire,

tellerFem, tellerSeks };

**int** mest = antallNummer[0];

**for** (**int** l = 0; l < antallNummer.length; l++) {

**if** (antallNummer[l] > mest)

mest = antallNummer[l];

}

skjerm.out("Terningverdien det var flest av: ");

**if** (mest == tellerSeks)

skjerm.outln(6);

**else** **if** (mest == tellerFem)

skjerm.outln(5);

**else** **if** (mest == tellerFire)

skjerm.outln(4);

**else** **if** (mest == tellerTre)

skjerm.outln(3);

**else** **if** (mest == tellerTo)

skjerm.outln(2);

**else** **if** (mest == tellerEn)

skjerm.outln(1);

}

}

## Eksempel på utskrift:

TERNINGKASTSIMULATOR

5 4 3 2 3 3 2 5 4 5 3 5 3 6 1 1 1 1 1 4

3 2 6 4 1 4 3 5 3 3 2 2 6 6 6 5 3 3 1 6

2 2 5 3 2 4 3 3 2 1 5 2 4 3 2 5 3 5 1 4

4 3 2 3 3 3 3 2 1 6 1 3 6 1 4 1 1 1 2 4

6 1 2 2 6 2 1 1 3 3 4 4 3 5 5 6 6 1 4 6

Antall kast: 100

Antall 6-arar: 13

Antall 5-arar: 12

Antall 4-arar: 14

Antall 3-arar: 25

Antall 2-arar: 17

Antall 1-arar: 19

Gjennomsnittskast: 3.220

Antall kast for å få den første 6-aren: 14

Terningverdien det var flest av: 3

# Oppg 4

## Beskrivelse:

### Kode:

**package** øving6;

**import** easyIO.\*;

**public** **class** Oppgave4 {

**public** **static** **void** main(String[] args) {

In tast = **new** In();

Out skjerm = **new** Out();

skjerm.out("Sett varenr: ");

**int** varenr = tast.inInt();

skjerm.out("Sett varenavn: ");

String navn = tast.inLine();

skjerm.out("Sett pris på vare: ");

**double** pris = tast.inDouble();

Klasser vare = **new** Klasser(varenr, navn, pris);

vare.skriv();

}

}

**class** Klasser {

**private** **int** varenr;

**private** String navn;

**private** **double** pris;

Klasser() {

pris = 100;

varenr = 0;

}

Klasser(**int** varenr, String navn, **double** pris) {

**this**.varenr = varenr;

**this**.navn = navn;

**this**.pris = pris;

}

**private** **int** getVarenr() {

**return** varenr;

}

**private** String getNavn() {

**return** navn;

}

**private** **double** getPris() {

**return** pris;

}

**public** **double** setMoms() {

**double** moms = (pris / 100) \* 20;

**return** pris + moms;

}

**public** **boolean** billigereEnn() {

**boolean** billigere = **false**;

**if** (pris < 100) {

billigere = **true**;

}

**return** billigere;

}

**public** **void** skriv() {

Out skjerm = **new** Out();

skjerm.outln(" ");

skjerm.outln("Varenavn: " + navn);

skjerm.outln("Varenummer: " + varenr);

skjerm.out("Pris u/moms: ");

skjerm.outln(pris, 2);

skjerm.out("Pris m/moms: ");

skjerm.outln(setMoms(), 2);

**if** (billigereEnn()) {

skjerm.outln("Varen er billigere enn 100.00 kr");

} **else** {

skjerm.outln("Varen er ikke billigere enn 100.00 kr");

}

}

}

## Eksempel på utskrift:

Sett varenr: 50001

Sett varenavn: Grandis

Sett pris på vare: 52

Varenavn: Grandis

Varenummer: 50001

Pris u/moms: 52.00

Pris m/moms: 62.40

Varen er billigere enn 100.00 kr