Obligatorisk oppgave nr. 2 (av 4) i INF1000

Leveringsfrist
Oppgaven må leveres senest **fredag 2. mars kl 16.00** via det elektroniske innleveringssystemet.

Viktig: Se siste side av oppgaveteksten for detaljerte leveringskrav.

Formål
Formålet med denne oppgaven er å gi trening i bruk av forgreninger, løkker, arrays, metoder, kommunikasjon med bruker via terminal og god programmeringsskikk som innrykk og kommentering av programkode.

Generell informasjon
Denne obligatoriske oppgaven skal løses individuelt. Besvarelsen må godkjennes som bestått for at du skal kunne gå opp til eksamen, men bidrar ikke til selve eksamenskarakteren. Studenter som har godkjent obligatorisk oppgave 2 fra et tidligere semester trenger ikke levere på nytt, men bør sjekke at de er oppført i listen over tidligere godkjente innleveringer. Gjennomføring og innlevering av oppgaven skal skje i henhold til gjeldende retningslinjer, se

http://www.ifi.uio.no/studinf/skjemaer/erklaring.pdf (norsk)
http://www.ifi.uio.no/studinf/skjemaer/declaration.doc (engelsk)

Enhver innlevering av besvarelse på denne oppgaven tas som en bekreftelse på at retningslinjene er lest og forstått.

Oppgave
Etter forelesningen 19. februar har du lært **alt** du trenger for å gjøre denne obligatoriske oppgaven, men mesteparten av arbeidet kan gjøres tidligere enn dette.

Oppgaveteksten er delt inn i fem forskjellige metoder. Disse kan programmeres uavhengig av hverandre, men skal sys sammen til slutt slik at brukeren av programmet kan velge hvilken av metodene han vil kjøre. Det endelige programmet skal ved oppstart skrive til skjermen hvilke valgmuligheter brukeren har. Eksempel på skjermutskrift:

```
Velg en av bokstavene A, B, C, D, E eller Q. Velger du:

A: kjøres metoden fakultet
B: kjøres metoden adderTilOppgittSum
C: kjøres metoden trekkEtKort
D: kjøres metoden blackJack
E: kjøres metoden DNA
Q: avslutter programmet.
```

Husk også å håndtere ulovlige menyvalg. Når et valg er gjort og metoden er ferdig, skal programmet igjen gi deg mulighet til å velge en av de fem metodene.

Metoden fakultet

Skriv en metode som multipliserer alle tallene fra 1 til n. Dette kalles å beregne “n fakultet” og skrives n!, for eksempel vil 4! = 4*3*2*1=24. Ved kjøring skal brukeren bli bedt om å oppgi et heltall n, deretter skal programmet skrive ut:

n! = <verdi fra utregningen>

Eksempel på utskrift:

```
5!  = 120
```

Metoden adderTilOppgittSum

Skriv en metode som adderer tallene fra 1 til n (n ukjent) inntil summen er større eller lik en verdi oppgitt av brukeren. Når metoden kjøres skal brukeren bli spurt om å oppgi en maksimal verdi. Programmet skal så skrive ut setningen:

Summen av 1, 2, 3, …, <n> er <sum> som er større eller lik <maksverdi>.

Eksempel på utskrift:

```
Oppgi en verdi større enn 10:
11

Summen av 1, 2, 3,…, 5 er 15 som er større eller lik 11.
```

Metoden trekkEtKort

Lag en metode som trekker et kort fra en kortstokk og skriver ut det kortet du trakk.

En kortstokk har 52 kort: 13 kort (2, 3, 4, 5, 6, 7, 8, 9, 10, Kn, D, K, A) i hver av de fire fargene (ruter, hjerter, kløver og spar) .

Tips: For å trekke et tilfeldig kort bruk random-generatorene i java. Et kall på metoden `Math.random()` vil returnere et flyttall x slik at $0 \leq x < 1$. For å beregne verdien til kortet kan dette tallet så multipliseres med 13, bruk kommandoen:

```
int n = (int)(13.0 * Math.random());
```

Du får da et tall i mengden {0, 1, 2, …, 12}. Husk at du også må trekke fargen på kortet.

Metoden blackJack

Lag en metode som spiller Black Jack. Metoden `blackJack` skal kalle på metoden `trekkEtKort`. Black Jack er et pokerspill hvor hver spiller får utdelt to kort, deretter kan spilleren trekke så mange kort han vil, men summen av kortenes verdi må ikke overstige 21. Den spilleren med sum nærmest 21 har vunnet. Verdien til kortene er:

- Tallkortene (2, 3, 4, 5, 6, 7, 8, 9) har verdi som tallet viser.
- Bildekortene (Kn, D, K) har verdi 10.
- Ess (A) har verdien 10. (Som regel spilles spillet slik at et ess kan ha verdien 1 eller 10, og at spiller selv kan bestemme hvilken verdi som skal gjelde. Vi forenkler spillet litt og sier at ess alltid har verdien 10).

Du kan anta at vi spiller med uendelig mange kortstokker, slik at samme kort kan trekkes flere ganger.

Etter å ha delt ut to kort til spilleren, oppgitt disse to kortene og summen av dem, skal blackJack metoden spørre om spilleren vil trekke enda et kort. Hvis han svarer ja, skal programmet trekke et nytt kort. Deretter skal det oppgi hvilket kort som ble trukket og summen av kortene som er trukket hittil. Programmet sjekker så om spilleren har tapt (det vil si at summen har overskredet 21), og rapporterer eventuelt tap.

Programmet skal gjenta spørsmålet om nytt kort og etterfølgende behandling helt til spilleren ikke vil ha flere kort eller til han har tapt. Hvis spilleren har tapt, skal programmet gi beskjed om dette. Hvis trekkingen stoppet fordi han ikke ville ha flere kort, skal programmet skrive ut summene av de trukne kortene.

Metoden DNA

DNA opptrer i naturen som en dobbel helix, hvor den ene strengen i helixen er bestemt av den andre. De to strengene kalles sense og anti-sense strenger. Som regel får man kun oppgitt sekvensen til sense strengen, sekvensen til anti-sense strengen må man finne selv. I denne oppgaven skal du lage et program som finner sekvensen til anti-sense strengen når sekvensen til sense strengen er gitt. DNA sekvenser består av fire forskjellige nukleotider (symbolene A, T, C og G), bindinger mellom sense og anti-sense sekvensene er slik at A binder seg med T, T binder seg med A, C binder seg med G og G binder seg med C. Hvis sekvensen på sense strengen er AAAAATGGGACCC kan vi skrive opp denne med sekvensen til anti-sense strengen under slik:

5’ AAAAATGGGACCC 3’
3’ TTTTTTACCCTGGG 5’

DNA sekvenser leses alltid fra 5 merket enden til 3 merket enden. Dermed vil sekvensen på anti-sense strengen leses motsatt vei av sense strengen. Sekvensen av nukleotidene vi er ute etter er i dette eksemplet er altså: 5’ GGGTCCCATTTTT 3’.

Du skal lage en metode som kan brukes til å skrive inn sekvensen til sense strengen og få ut sekvensen til anti-sense strengen. Metoden skal gi feilmelding hvis andre bokstaver enn A, T, G eller C skrives inn. Metoden skal skrive ut sekvensen til den tilhørende anti-sense strengen.

Eksempel på utskrift når metoden kjøres:

```
Skriv inn DNA-sekvensen til sense strengen:
AAAGAAC

DNA sekvensen til anti-sense strengen er: GTTCTTT.
```

Eksempel med feil i inndata:

```
Skriv inn DNA-sekvensen til sense strengen:
AACGGH

Du har skrevet bokstaven H, H er ikke en av de fire nukleotidene A, T, C eller
G.

Skriv inn DNA-sekvensen til sense strengen:
…
```

Leveringskrav

For å få godkjent denne obligatoriske oppgaven må alle de fem metodene være programmert. Alle metodene må kunne kjøres fra en valgmeny som kommer opp når programmet `Oblig2.java` kjøres. Løsningskoden må følge god programmeringsskikk, det vil si ha innrykk og kommentarer. Kommentarer som `/* her slutter metode DNA */` eller `/* her finnes anti-sense strengen med riktig leseramme */` kan virke tåpelige i små programmer. Når vi senere skal lage større programmer, vil slike kommentarer hjelpe både deg og andre til å lettere finne fram i koden. Husk likevel at kommentarer ikke skal florere over alt, og at de skal være meningsfylte.

Du plikter å ha lest og forstått følgende krav til innleverte oppgaver ved institutt for informatikk:

http://www.ifi.uio.no/studinf/skjemaer/erklaring.pdf

Oppgaven skal leveres elektronisk. Første linje i programfilen (.java-filen) skal se slik ut:

```
// dittBrukernavn  g-gruppeNummer  o-2  k-inf1000/v07
```

hvor *dittBrukernavn* skal erstattes av ditt eget brukernavn (det navnet du oppgir når du logger deg inn på UiO's anlegg), og *gruppeNummer* skal erstattes av nummeret på den øvingsgruppa du er tatt opp på (hvis du har byttet øvingsgruppe, skal du altså oppgi gruppenummeret til den opprinnelige øvingsgruppa). Eksempel: hvis ditt brukernavn er anjab og du skal levere oblig2 på gruppe 3, så skal første linje i programfilen se slik ut:

```
// anjab  g-3  o-2  k-inf1000/v07
```

Send programfilen (.java-filen) og resultatet av en testkjøring av hvert av metoden (se nedenfor) til gruppelæreren din gjennom innleveringssystemet (Joly). Resultatet av testkjøringen skal vise utskriften på skjermen når du starter opp programmet ditt og tester ut de forskjellige delene som du er spurt om å lage i oppgaven. Kjører du Unix, kan du bruke et program som heter *photo* for å lagre på fil alt som skrives ut på skjermen under en testkjøring. Du gir da følgende kommando i xterm-vinduet rett før du starter java-programmet ditt:

```
photo testOblig2.txt
```

Etter at du har startet programmet ditt, testet det ut og fått det til å avslutte, gir du kommandoen exit (eller trykker Control-D) for å avslutte photo-programmet. Nå ligger all utskriften fra testen på filen *testOblig2.txt* (du kan kalle filen noe annet hvis du ønsker det).

Hvis du har spørsmål vedrørende leveringsmåte eller andre spørsmål til oppgaven, så kontakt gruppelæreren din i god tid før innleveringsfristen. Det er ditt ansvar at oppgaven kommer frem til gruppelæreren på riktig måte innen leveringsfristen

Lykke til!