

**Oppgave 1**

<i>Oppgavenr</i>	<i>A</i>	<i>B</i>	<i>C</i>
1.1	O		
1.2		O	
1.3			O
1.4			O
1.5	O		
1.6		O	
1.7		O	
1.8	O		
1.9			O
1.10			O
1.11		O	
1.12	O		
1.13		O	
1.14	O		
1.15	O		
1.16			O
1.17			O
1.18	O		
1.19		O	
1.20		O	

## Oppgave 2 – Programmering (70 %)

a) (5%) Skriv metoden *lag\_tomt\_brett* som genererer (lager) et tomt kvadratisk spillbrett (tabell bestående av bare 1'ere) av en gitt størrelse bestemt av inn-parameteren *storrelse*.

```
int[][] lag_tomt_brett(int storrelse) {
    int[][] tabell = new int[storrelse][storrelse];
    for (int i=0;i<storrelse;i++) {
        for (int j=0;j<storrelse;j++) {
            tabell[i][j] = 1;
        }
    }
    return tabell;
}
```

b) (10%) Skriv metoden *legg\_ut\_epler* som tar inn spillbrettet (en tabell av vilkårlig størrelse), samt antallet epler som skal legges ut på vilkårlige posisjoner på brettet. Et eple blir representert av tallet "5". Merk at metoden ikke skal legge flere epler ("5" tall) på samme posisjon.

```
void legg_ut_epler(int[][] tabell, int antall) {
    while (antall>0) {
        int x = (int) (Math.random()*tabell.length);
        int y = (int) (Math.random()*tabell.length);
        if (tabell[y][x]==1) {
            tabell[y][x] = 5;
            antall = antall - 1;
        }
    }
}
```

c) (5%) Skriv metoden *sjekk\_eple* som tar inn en x- og en y-posisjon (typisk hodet på ormen), samt spillbrettet. Metoden skal returnere *true* hvis et eple befinner seg på den angitte posisjonen (dvs. tallet "5"), og *false* hvis det ikke gjør det.

```
boolean sjekk_eple(int x, int y, int[][] table) {
    if (table[y][x]==5) {
        return true;
    } else {
        return false;
    }
}
```

d) (15%) Skriv metoden *tegn\_orm* som tar inn spillbrettet (en tabell av vilkårlig størrelse) samt de to tabellene som representerer ormen (*orm\_x* og *orm\_y*) og tegner ormen inn i spillbrettet representert med 0'ere. Halen på ormen skal tegnes som en 1'er.

```
void tegn_orm(int[][] table, int[] orm_x, int[] orm_y) {
    for (int i=0; i<orm_x.length-1;i++) {
        table[orm_y[i]][orm_x[i]] = 0;
        // Tegner 0 for ormen
    }
    table[orm_y[orm_y.length-1]][orm_x[orm_x.length-1]] = 1;
    // Tegner inn 1 for siste del av ormen
}
```

e) (15%) Skriv metoden *sjekk\_krasj* som tar inn tabellene for ormen (*orm\_x* og *orm\_y*), samt retningen ormen skal bevege seg i ("N", "E", "S" eller "V"). Metoden skal sjekke om ormen krasjer i seg selv hvis den beveger seg i angitt retning. Metoden skal returnere *true* hvis den krasjer og *false* hvis den ikke gjør det.

```
boolean sjekk_krasj(int[] orm_x, int[] orm_y, String retning) {
    int sjekk_x=orm_x[0];
    int sjekk_y=orm_y[0];
    if (retning.equals("N")) {
        sjekk_y--;
    } else if (retning.equals("E")) {
        sjekk_x++;
    } else if (retning.equals("S")) {
        sjekk_y++;
    } else if (retning.equals("V")) {
        sjekk_x--;
    }
    boolean krasj = false;
    for (int i=0;i<orm_x.length-1;i++) {
        if (sjekk_x==orm_x[i] && sjekk_y==orm_y[i]) {
            krasj = true;
        }
    }
    return krasj;
}
```

f) (10%) Skriv metoden *flytt\_orm* som tar inn tabellene for ormen (*orm\_x* og *orm\_y*), samt retningen ormen skal bevege seg i ("N", "E", "S" eller "V"). Metoden skal flytte posisjonene i de to tabellene slik at ormen beveger seg i riktig retning. Hvis vi tar utgangspunkt i verdiene gitt i tabellene *orm\_x* og *orm\_y* ovenfor, så vil for eksempel en forflytning i retning "N" (Nord) gi tabellene: *orm\_x* = {3,3,3,3,2} og *orm\_y* = {1,2,3,4,4}. Dette betyr at alle posisjonene for x og y i ormen flyttes bakover i tabellen (foruten siste posisjon) og at *orm\_x[0]* og *orm\_y[0]* får ny posisjon som er lik forflytningen.

```
void flytt_orm(int[] orm_x, int[] orm_y, String retning) {
    int temp_x = orm_x[0]; // ta vare på fremste x-posisjon på ormen
    int temp_y = orm_y[0]; // ta vare på fremste y-posisjon på ormen
    for (int i=orm_x.length-1;i>0;i--) {
        orm_x[i] = orm_x[i-1]; // flytt ormen framover x-posisjon
        orm_y[i] = orm_y[i-1]; // flytt ormen framover y-posisjon
    }
    orm_x[0] = temp_x; // Gi tilbake original fremste posisjon x
    orm_y[0] = temp_y; // Gi tilbake original fremste posisjon y
    if (retning.equals("N")) {
        orm_y[0]--;
    } else if (retning.equals("E")) {
        orm_x[0]++;
    } else if (retning.equals("S")) {
        orm_y[0]++;
    } else if (retning.equals("V")) {
        orm_x[0]--;
    }
}
```

g) (10%) Bruk metodene ovenfor for å skrive kode som gjør følgende (trenger ikke å definere en egen metode for denne koden):

1. Opprett en heltalls variabel *score* som tar vare på antall poeng i spillet og sett denne variabelen lik 0
2. Opprett et *spillbrett* (tabell) med størrelsen 10 x 10 og fyll tabellen med 1'er
3. Opprett tabellene *orm\_x* og *orm\_y* for å representere ormen med størrelse 3
4. Legg inn følgende punkter i ormen: Hodet er punktet (3,4), midten er punktet (2,4) og halen er punktet (1,4)
5. Legg ut 5 epler på spillbrettet
6. Sjekk om posisjonen til hodet på ormen på spillbrettet har mat (tallet 5), og øk variabelen *score* med 10 hvis dette er tilfellet
7. Sjekk om ormen krasjer i seg selv hvis den beveger seg nordover ("N") og minsk variabelen *score* med 5 hvis dette er tilfelle
8. Flytt ormen en posisjon i retning nordover ("N")
9. Tegn ormen inn i *spillbrett* (tabellen)
10. Skriv ut spillbrettet (tabellen) til skjerm ved hjelp av *out.println()*

### Løsning:

```
<%
int score = 0; // #1
int hode = 0;

int[][] spillbrett = lag_tomt_brett(10); // #2
int[] orm_x = new int[3]; // #3
int[] orm_y = new int[3];
orm_x[0] = 3; orm_y[0] = 4; // #4
orm_x[1] = 2; orm_y[1] = 4;
orm_x[2] = 1; orm_y[2] = 4;

legg_ut_epler(spillbrett, 5); // #5

if(sjekk_eple(orm_x[hode],orm_y[hode], spillbrett)) { // #6
    score = score + 10;
}

if(sjekk_krasj(orm_x,orm_y,"N")) { // #7
    score = score - 5;
}

flytt_orm(orm_x,orm_y,"N"); // #8

tegn_orm(spillbrett, orm_x, orm_y); // #9

for (int i = 0 ; i<spillbrett.length; i++) { // #10
    for (int j = 0; j<spillbrett.length; j++) {
        out.println(spillbrett[i][j]);
    }
    out.println("<br>");
}
%>
```

### Oppgave 3: HTML (10%)

I denne oppgaven får du oppgitt en html-fil med 10 mangler. Du skal skrive HTML-koden for å lage en webside som vist under. Merk følgende opplysninger:

1. Alle innslag i tabellen er i utgangspunktet (default) venstrejustert horisontalt og midtjustert vertikalt.
2. Bildet *geek.jpg* skal brukes i venstre kolonne og har størrelsen 310 x 310 punkter (pixels).
3. Bildet *header.jpg* skal brukes i midtkolonnen og har størrelsen 825 x 105 punkter (pixels).
4. Venstre kolonne skal ha bredde på 120 punkter (pixels).
5. Header i midtkolonnen skal ha høyde på 70 punkter (pixels).

Du skal *kun* skrive HTML-kode for de tomme feltene slik at nettleseren vil vise koden slik som vist i skjermbildet under:

	<p><b>Hjalmar Pjalle's kuule webside!!!</b></p>	<p>Dato: 25.mai 2010</p>
<p><b>Meny</b></p> <p><a href="#">Verdens Gang</a>  <a href="#">Dagens Næringsliv</a>  <a href="#">Vårt Land</a>  <a href="#">Dagbladet</a>  <a href="#">NRK</a></p>	<p><b>Velkommen til hjemmesiden min!</b></p> <p>Her skal hoveddelen av websiden være.          Her kommer det mye rart...</p> <p>Lurer på om studentene i ITGK klarer layouten...          Håper det...</p>	
<p>Ta kontakt på epost: <a href="mailto:hjalmar.pjalle@mysilgoerra.uff">hjalmar.pjalle@mysilgoerra.uff</a></p>		

(oppgaven fortsetter på neste side)

Løsning:

## Html-filen

1	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2	<html>
3	<head>
4	<title>Hjemmeside layout</title>
5	</head>
6	<body>
7	<table border="1" cellpadding="5">
8	<tr>
9	<td rowspan="3" valign="top">
10	
11	<h3>Meny</h3>
12	<a href="http://www.vg.no">Verdens Gang</a> 
13	<a href="http://www.dn.no">Dagens Næringsliv</a> 
14	<a href="http://www.vl.no">Vårt Land</a> 
15	<a href="http://www.dagbladet.no">Dagbladet</a> 
16	<a href="http://www.nrk.no">NRK</a> 
17	</td>
18	<td>
19	
20	</td>
21	<td align="right">
22	Dato: 25.mai 2010
23	</td>
24	</tr>
25	<tr>
26	<td colspan="2" valign="top">
27	<h1>Velkommen til hjemmesiden min!</h1>
28	<p>Her skal hoveddelen av websiden være. Her kommer det mye rart... </p>
29	<p>Lurer på om studentene i ITGK klarer layouten... Håper det...</p>
30	</td>
31	</tr>
32	<tr>
33	<td colspan="2" align="center">
34	Ta kontakt på epost: <b><u>hjalmar.pjalle@mysilgoerra.uff</u></b>
35	</td>
36	</tr>
37	</table>
38	</body>
39	</html>