



Inf1000 (Uke 13)

IT og samfunn og Black Jack eksempel

Grunnkurs i programmering
Institutt for Informatikk
Universitetet i Oslo

Anja Bråthen Kristoffersen og Are Magnus Bruaset



Dagens plan

- IT og samfunn
- Et stort eksempel:
Black Jack med kun en kortstokk.



Et teknologi-perspektiv

- Dampmaskinen er ca. 200 år
- Elektrisitet (dynamo/motor) ca. 150 år
- Bilen ca. 100 år
- Datamaskinen (edb) ca. 50 år



Er datateknologi lik annen teknologi ?

- IT kommer i alle sektorer og yrker
- Vedvarende prisfall
- Elektromotorer: Automatisering av fysisk arbeid i alle produkter
- Integreerte kretser: Automatisering av beslutninger
- IT-produkter slites 'ikke'
- Innføringstakten høyere og mer vedvarende



Samfunnsmessige virkninger

- Internett
- Jobbenes innhold
- Arbeidsløshet ?
- Overvåking



Internett -virkninger

- Elektronisk post utkonkurrerer klart Posten
- Internett som tillegg til andre media
- Jobbe hjemme
- Nye dataløsninger: Intranett
- Effektivisering av forskning & utvikling
- Banktjenester, kjøp/salg mellom bedrifter, e-handel

Ikke mye folkeopplysning, men business og underholdning



Eksempler på IT-rasjonalisering

- Mange enkelt-eksempler
 - bankene, posten
 - Televerkene
 - Dataprodusentene (som IBM, Dec, ND, Compaq ..)
 - Mekanisk industri,
 - Bryggerier, bilindustri
 - Erstatte mekaniske deler med elektronikk (printere, telefonsentraler)



Overvåking, oversikt

- Ekstremt effektivisert av IT
- (Hverdags) eksempler:
 - Sentralt samtaleregister i Telenor
 - Kjøpe/rabatt-kort 'elektroniske spor'
 - Salg av person/medlemsregistre
 - Drosjer (og fanger) har GPS e.l.
 - Mobiltelefon som er slått på gir nøyaktig 'spor'
 - Spionsatellitter-<10 cm. oppløsning
 - Telefonavlytting -dels hel automatisert
 - Helsedata (DNA-profil) til forsikringsselskapet?
 - Videokameraer over alt (automatisk personidentifikasjon)
 - Kombinering av registre
- Storebror ser deg (Schengen-avtalen, Ecelon)
 - Forbrytere, vitner og overvåkete (kan) registreres, all e-post 'avlyttes'
- Lillebror ser deg
 - Alle kan overvåke alle (BigBrother: TV-underholdning)



Lov om behandling av personopplysninger – trådd i kraft 1 jan. 2001

- **Hovedprinsipper:**

- **Samtykke:**

du må ha gitt lov til registrering

- **Informasjonsplikt:**

virksomheten har plikt til å informere den registrerte. Virksomheten kan ikke ta betalt for å gi informasjon eller innsyn. Du skal informeres eller gies et foreløpig svar innen 30 dager.

- **Meldeplikt for behandling:**

Virksomheten må melde til Datatilsynet (på forhånd) hvilke systemer de har og hva disse gjør.

- **Gjelder bare fysisk personer:**

Ikke firma og lignende.



Paragrafer

- **§ 2 –3 Definisjoner. Loven gjelder for elektroniske personregistre og for annen behandling av personopplysninger. ..**
- §2 nr.8: Sensitive opplysninger er opplysninger om:
 - Rase, etnisk, politiske .., osv forhold
 - Om man er dømt, mistenkt .el.
 - Helse- og seksuelle forhold
 - medlemskap i fagforeninger
- **§ 8 Behandling av personopplysninger skal bare gjøres etter samtykke og være saklig begrunnet, må være nødvendig før:**
 - a) Oppfylle avtale med den registrerte
 - c) Ivareta den registrertes interesser
 - e) å utøve offentlig myndighet
 - f) å ivareta en berettiget interesse som overstiger den registrertes interesse
- **§9 Behandling av sensitive opplysninger, bare hvis**
 - a) den registrerte samtykker
 - b) fastsatt i lov
 - g – h) nødvendig av helse, statistiske undersøkelser



§ (forts)

- **§ 18 Rett til innsyn i registrerte opplysninger**
- **§ 19 Informasjonsplikt til den registrerte**
- **§20. Informasjonsplikt**
når det samles inn opplysninger fra andre enn den registrerte
- **§ 22 Rett til info om automatiske avgjørelser – og rett til manuell behandling (§25)**
- **§ 26. Rett til å reservere seg mot direkte markedsføring**
- **§ 27 Rett til å få rettet mangelfulle/uriktige personopplysninger**
- **§ 29 Data kan overføres til et utland** (som følger et EF direktiv) eller får unntak etter § 30 f.eks pkt. g) nødvendig eller følger av lov for å beskytte en viktig samfunnsinteresse
- **§ 31 - 32 Meldeplikt til Datatilsynet** for registre (30 dager)
- **§ 33 Konsesjon for sensitive personopplysninger**



§ (forts)

- Fjernsyns (video) overvåking (§ 36-41)
 - må ha særskilt behov
 - må gi varsel
 - kan utlevere til politiet, følger av lov
- § 42 – 47 Datatilsynets oppgaver
- § 48 – straff inntil 1 år - 3 år ved særdeles skjerpene omstendigheter
- § 49 – Erstatning ved feil, brudd på lov begått av de som behandler / utleverer data. Ved skade ved feilaktige kredittopplysninger skal skade erstattes uansett om det er begått feil eller ikke fra behandlers side.



Du må kjenne loven om behandling av personopplysninger.

- Fordi
 - Du har nå lært så mye programmering at du kan begynne å bryte loven!
 - Husk alltid å hent inn samtykke ved registrering av personer
 - Meld fra til datatilsynet



Eksempler på brudd på loven.

Publisert: 03. oktober 2005-Oppdatert: 03. oktober 2005 kl.17:05

Redningsselskapet anmeldt for lovbrudd

Datatilsynet anmeldte i dag Redningsselskapet til politiet for flere brudd på personopplysningsloven.

- ❑ Selskapet søkte gjennom e-posten til flere medarbeidere.
- ❑ Saken endte blant annet med at generalsekretær Monica Kristensen Solås måtte gå.



Dagbladet 17.08.2005:

Blir anmeldt for epost-lesing

Styreleder Harald Arnkværn i Vinmonopolet undersøkte de ansattes epost.

For å oppklare smøreskandalen i vinmonopolet brøt styrelederen Harald Arnkværn personvernloven. Han leide inn dataeksperter fra selskapet IBAS for å granske polets epostservere og to filservere.

Saken endte blant annet med at Harald Arnkværn trakk seg som styreleder i Vinmonopolet.

13.03.2006

Datatilsynet: Rekrutteringsfirma har ulovlig database

Av [Therese Werenskiold](#)

Rekrutteringsselskapet IT Executive Search & Selection skryter av å inneha Norges største database over IT-ansatte. Selskapet har imidlertid ikke søkt Datatilsynet om konsesjon, noe som er ulovlig og kan medføre fengselsstraff.

Publisert 12.01.2006 13:48

Fortsetter utstedelsen av biometrisk pass



Av [Ann Kristin Bentzen Ernes](#)

(12.01.2006) - Nye sikkerhetskrav fra Datatilsynet på biometriske pass stopper ikke politiet i å gi ut nye.

Publisert 21.02.2006 08:41

Datatilsynet avslår biometrisk ID

Av [Ann Kristin Bentzen Ernes](#)

Datatilsynet har avslått alle forslag om biometrisk ID som er fremmet hittil.



Black Jack, basert på et program skrevet av Einar Fride Myskja

- I oblig 2 programmerte dere metoden blackJack.
- Som en forenkling antok vi at det ble spilt med uendelig mange kortstokker slik at vi kunne trekke samme kort mange ganger.
- Da oblig 2 skulle programmeres hadde vi ikke lært om objekter. Vi skal nå løse oppgaven med objektorientert programmering.



Organisering av dataene, noen spørsmål:

- Hvordan holde rede på kortene i kortstokken?
 - slik at vi ikke trekker samme kort to ganger.
- Hvordan gi kortene en verdi?
- Hvilke klasser og metoder trenger vi?



Hvilke klasser og metoder trenger vi?

- En løsning kan være klassene:
 - Kortspill
 - main metode
 - Meny
 - Metode for å starte blackJack evt. avslutte.
 - Spillebord
 - Summerings metode, spillBlackJack metode.
 - Kortstokk
 - Hvilke kort finnes fortsatt i kortstokken, Hashmap
 - trekkNyttKort metode
 - Kort
 - farge, valør, utskriftsmetode.



class Kort

- Bruker oppramsingstypen enum for
 - farge
 - valør
- Utskriftrutine for å skrive ut kortet til skjerm
- Konstruktør som sørger for at hvert objekt av typen kort har en farge og en valør.



Kort repetisjon av enum

- En variabel som skiller mellom ulike navngitte verdier.
- Kan iterere over elementene i enum ved å bruke `.values()`.
- Hvert navngitte element får en verdi ut fra plasseringen den kan finnes ved metoden `.ordinal()`.
- I tillegg kan hvert av elementene i enum få en tilhørende verdi oppgitt i parentes ved oppramsingen.



enum Farge

```
enum Farger {  
    Spar,Hjerter,Ruter,Kløver;  
    static Farger randomFarge () {  
        int enAv4 = (int) (4.0 * Math.random());  
        Farger farge = Spar;  
        for (Farger f: Farger.values()) {  
            if (f.ordinal() == enAv4){  
                farge = f;  
            }  
        }  
        return farge;  
    }  
}
```

```

enum Valorer {
    $2(2), $3(3), $4(4), $5(5), $6(6), $7(7), $8(8), $9(9), $10(10), J(10), D(10), K(10), E(10);
    private final int blackJackVerdi;

    Valorer (int v) {
        blackJackVerdi = v;
    }

    static Valorer randomValør () {
        int enAv13 = (int) (13.0 * Math.random());
        Valorer valør = E;
        for (Valorer v: Valorer.values()) {
            if (v.ordinal() == enAv13){
                valør = v;
            }
        }
        return valør;
    }

    int hentVerdi() {
        return blackJackVerdi;
    }
}

```



```
class Kort {
    // Variabeldeklarasjoner
    Farger farge;
    Valorer valør;

    Kort(Farger f, Valorer v) {
        farge = f;
        valør = v;
    }

    void skrivUtKort(Out skjerm) {
        skjerm.out(farge + " ", 8, Out.LEFT);
        skjerm.outln("" + valør, 3, Out.RIGHT);
    }
} // slutt class Kort
```



class Kortstokk

- Representer kortstokken som en HashMap.
 - Legg alle kortene i kortstokken (dette gjøres i Konstruktøren til klassen Kortstokk)
 - Fjern kortene fra kortstokken etter hvert som de blir trukket
 - Hold rede på hvor mange kort som fortsatt er i kortstokken, variabel antIKortstokken
- trekkNyttKort
 - Trekker et kort som ligger i kortstokken (har en nøkkel i HashMap-en)
 - Minker antIKortstokken med 1.

```
class Kortstokk {
    HashMap <String,Kort> stokken = new HashMap <String,Kort> ();
    int antIstokken;

    Kortstokk () {
        String nøkkel;
        Kort kort;
        for (Farger f: Farger.values()) {
            for (Valorer v: Valorer.values()) {
                kort = new Kort(f, v);
                nøkkel = kort.farge + ";" + kort.valør;
                stokken.put(nøkkel, kort);
            }
        } //legger alle kortene inn i kortstokken
        antIstokken = 52; }

    Kort trekkNyttKort() {
// denne metoden må skrives
    }
}
```

```
Kort trekkNyttKort() {
    Farger f;
    Valorer v;
    Kort kort = null; //Metoden returnerer null dersom ingen flere kort
    String nøkkel;
    if (antIstokken > 0) {
        do {
            f = Farger.randomFarge();
            v = Valorer.randomValør();
            nøkkel = f + ";" + v;
        } while(!stokken.containsKey(nøkkel)); // trekker
        inntil kort som fortsatt er i kortstokken blir
        trukket.
        kort = (Kort) stokken.remove(nøkkel); //fjerner det
        kortet som er trukket fra kortstokken.
        antIstokken--; //minker antall i kortstokken
    }
    return kort; //returnerer det trukne kortet.
}
```



class SpilleBord

- Variabler:
 - kortplass, tabell med kort til alle kortene som er trekt.
 - kortstokk av klassen Kortstokk
- Metoder:
 - sumPåBordet
 - summerer kortene trekt hittil.
 - skrivUtBordet
 - skriver ut kortene trekt hittil.
 - spillBlackJack
 - deler ut to kort til spilleren,
 - spør om han vil ha flere kort
 - fortsetter til tap evt. spiller ikke vil ha flere kort.

```
class Spillbord {
    // Variabeldeklarasjoner
    // 9 kortplasser fordi: 2*4 + 3*4 == 20
    Kort[] kortplass = new Kort[9];
    Kortstokk kortstokk = new Kortstokk();

    int sumPåBordet(int antKortPåBordet) {
        //denne metoden må skrives
    }
    void skrivUtBordet(int antKortPåBordet, Out skjerm) {
        //denne metoden må skrives
    }
    void spillBlackJack(In tastatur, Out skjerm) {
        //denne metoden må skrives
    }
}
```



Metoden sumPåBordet

```
int sumPåBordet(int antKortPåBordet) {  
    int sum = 0;  
    for (int i = 0; i < antKortPåBordet; i++) {  
        sum = sum + kortklass[i].valør.hentVerdi();  
    }  
    return sum;  
}
```



Metoden skrivUtBordet

```
void skrivUtBordet(int antKortPåBordet, Out skjerm) {  
    for (int i = 0; i < antKortPåBordet; i++) {  
        skjerm.out(i+1 + ". ", 4, Out.RIGHT);  
        kortplass[i].skrivUtKort(skjerm);  
    }  
}
```



```
void spillBlackJack(In tastatur, Out skjerm) {
    final int BLACKJACKMAX = 21;
    char valg;
    skjerm.outln("Spiller BlackJack!");
    int i = 0; //angir antall trekte kort
    kortplass[i] = kortstokk.trekkNyttKort();
    i = 1; // Ett kort på bordet
    kortplass[i] = kortstokk.trekkNyttKort();
    i = 2; // To kort på bordet
    skrivUtBordet(i, skjerm);
    skjerm.outln("Summen på bordet:" + sumPåBordet(i));
    skjerm.outln();
    skjerm.out("Vil du trekke et nytt kort (J/N):");
    valg = tastatur.inChar();
    tastatur.inLine();
    while ((valg=='J' || valg=='j') && kortstokk.antIstokken > 0) {
        //denne løkken må skrives. Her skal valget utføres
    }
}
```

```

while ((valg=='J' || valg=='j') && kortstokk.antIstokken > 0) {
    kortplass[i] = kortstokk.trekkNyttKort();
    i++;
    skrivUtBordet(i, skjerm);
    skjerm.outln("Summen på bordet:" + sumPåBordet(i));
    skjerm.outln();
    if (sumPåBordet(i) <= BLACKJACKMAX) {
        skjerm.out("Vil du trekke et nytt kort (J/N):");
        valg = tastatur.inChar();
        tastatur.inLine();
    } else {
        skjerm.out("Du har tapt!");
        skjerm.outln();
        valg = 'N';
    }
}
}

```



class Meny

- Skal deklarerere utskrift og innlesnings objekt.
- Metoden presenter skal presentere mulige valg.
- Metoden trekkEtKort, trekker et kort fra en ny kortstokk.
- Metoden blackJack, utfører spillet Black Jack.

```
class Meny {
    Out skjerm = new Out();
    In tastatur = new In();
    void presenter() {
        //Må skrives
    }
    void trekkEtKort() {
        skjerm.outln("Trekker et kort:");
        Kortstokk kortstokk = new Kortstokk();
        Kort kort = kortstokk.trekkNyttKort();
        kort.skrivUtKort(skjerm);
    }

    void blackJack() {
        Spillbord bord = new Spillbord();
        bord.spillBlackJack(tastatur, skjerm);
    }
}
```

```
void presenter() {
    char valg;
    do {
        skjerm.outln("Meny: (velg en bokstav)");
        skjerm.outln(" T: Trekk et kort          ");
        skjerm.outln(" S: Spill BlackJack          ");
        skjerm.outln(" Q: Avslutt          ");
        skjerm.out("Valg:");
        valg = tastatur.inChar();
        tastatur.inLine();
        switch (valg) {
            case 'T':
                trekkEtKort();
                break;
            case 'S':
                blackJack();
                break;
            case 'Q':
                break;
            default:
                skjerm.outln("Velg en bokstav i menyen. Husk stor
bokstav!");
                break;
        }
    } while (valg != 'Q');
}
```



class Kortspill

- Skal kun inneholde main og sette i gang spillet når javafilen kjøres.

```
class Kortspill {  
    public static void main(String[] args) {  
        Meny meny = new Meny();  
        meny.presenter();  
    }  
}
```



Black Jack

