UNIVERSITETET I AGDER

FAKULTET FOR ØKONOMI OG SAMFUNNSVITENSKAP

E K S A M E N

Emnekode:	IS-211
Emnenavn:	Algoritmer og datastrukturer
Dato:	23 Mai 2016
Varighet:	0900 – 1300
Antall sider inkl. forside:	5
Målform:	Norsk
Tillatte hjelpemidler:	Alle trykte og skrevne

Innledning

Denne oppgaven handler om et fiktivt system for håndtering av pasienter som står på venteliste for å få en operasjon for å bli kvitt en sykdom (som også er fiktiv).
Les gjennom alle oppgavene før du begynner å svare. Det er to vedlegg til oppgaven som inneholder ufullstendige klassedefinisjoner. Du skal skrive ferdig noen av metodene. Noen metoder er beskrevet med signatur og javadoc kommentarer. Disse metodene kan du bruke fritt.
Du kan legge til så mange felt og metoder som du mener er nødvendig for å løse oppgaven, men husk at de metodene du legger til også må skrives ferdig.
Hvis du får dårlig tid, kan du beskrive metodene med pseudokode eller lignende, men husk at du må beskrive hvordan metodene skal gjøre det de gjør, i motsetning til metodene som er gitt i oppgaven (der står det ingenting om hvordan de er/kan bli implementert).

Oppgave 1

Den enkleste måten å håndtere pasientkøen på, er å behandle pasienten i den rekkefølgen de ble henvist til sykehuset.

a) Hva slags datastruktur vil du bruke til å holde på pasientene som venter på operasjon. Begrunn svaret.
b) Skriv de nødvendige deklarasjonene, og skriv ferdig konstruktoren til klassen Sykehus (Se vedlegg 1)
c) Skriv ferdig metodene nyHenvisning() og hentNestePasient() i klassen Sykehus

Oppgave 2

Den enkle håndteringen av ventelisten som er beskrevet i oppgave 1 er ikke særlig tilfredsstillende. Pasienter som kunne blitt friskmeldt og vært tilbake i jobb, blir gående sykmeldt, og kan få smerter og ubehag mens de venter på operasjonen. I verste tilfelle kan de gå over i en kronisk tilstand hvor det ikke lenger er mulig å kurere sykdommen. Derfor er det nødvendig å prioritere pasienten. Pasientene på ventelisten kan deles i tre grupper:

- A: Pasienten er rammet av sykdommen, men kan fortsatt være i jobb hvis han er yrkesaktiv, og har i liten grad plager.
- B: Pasienten har alvorlige plager, og må sykmeldes
- C: Tilstanden er i ferd med å bli kronisk, og må behandles umiddelbart for å unngå at pasienten får sterke plager.

Pasienter i gruppe C skal alltid gå foran alle andre, hvis flere gruppe C pasienter står på ventelisten skal de behandles i samme rekkefølgen som de ble henvist (samme opplegg som i oppgave 1),
Pasienter i gruppe B går foran pasienter i gruppe A, men yrkesaktive pasienter i gruppe A og B blir prioritert foran pasienter som ikke er i arbeid innenfor samme gruppe. Du skal nå lage et forbedret system, som tar hensyn til denne prioriteringsrekkefølgen

a) Hvilke datastruktur(er) vil du nå bruke til å holde på pasientene på ventelisten. Begrunn svaret.
b) Skriv ferdig metoden prioritertForan() i klassen pasient (vedlegg 2)
c) Skriv nye versjoner av metodene nyHenvisning() og hentNestePasient() i klassen Sykehus, som bruker datastrukturene fra oppgave 2a, og tar hensyn til prioriteringene som er beskrevet over.

Vedlegg 1: class Sykehus

```
public class Sykehus {
    // Deklarasjoner av datstruktur(er)
    // opg 1b

    public Sykehus() {
        // opg 1b
    }

    /**
     * Legger pasienten p inn i datastrukturen sammen med de
     * andre som venter på operasjon.
     * @param p pasienten som skal legges inn i datastrukturen.
     */
    public void nyHenvisning(Pasient p) {
        // opg 1 c
    }


    /**
     * Henter neste pasient som skal behandles fra datastrukturen.
     * Pasienten skal fjernes fra datastrukturen. (En pasient skal
     * bare behandles en gang)
     */
    public Pasient hentNestePasient() {
    }


    /**
     *  Denne er bare for å vise hvordan metodene over er tenkt brukt.
     *  Anta at systemet har ei løkke som kaller denne metoden så mange
     *  ganger som kirgurgene klarer å operere.
     *
     */
    public void behandleNestePasient() {
        Pasient p = hentNestePasient();
        opererPaa(p);
        skrivUt(p);
    }
```

Vedlegg 2 class Pasient

```
public class Pasient {
    /**
     * Returnerer “A”, “B”, eller “C” avhengig av hvilken gruppe
     * pasienten tilhører
    public String getGruppe();

    /**
     * Returnerer true hvis pasienten er yrkesaktiv, false hvis
     * ikke er yrkesaktiv, f.eks. Pensjonist, trygdet,...
     */
    public boolean erYrkesaktiv();


    /**
     * Returnerer true hvis denne pasienten, har stått lengre på
     * ventelisten en pasienten p,
     * dvs. denne pasienten ble henvist først
     */
    public boolean henvistFoer(Pasient p);

    /**
     * Returnerer true hvis denne pasienten skal prioriteres
     * foran pasienten p,
     */
    public booelan prioritertForan(Pasient p) {
        // opg 2 c
    }
}
```

SLUTT

UNIVERSITETET I AGDER