Tentamen

DD2385 Programutvecklingsteknik vt 2010

Torsdagen den 27 maj 2010 kl 9.00 – 12.00

Hjälpmedel: penna, suddgummi, linjal

Tentan har två delar om vardera 25 poäng

Maximala betygsgränser (gränserna kan bli lägre men inte högre):

Betyg FX:	18-19 poäng på del I
Betyg E:	20-25 poäng på del I
Betyg C/A:	20-25 poäng på del I och 13-20(för C)/21-25(för A) poäng på del II

Betyg FX får kompletteras till C eller A om
C- respektive A-nivån på del II är nådd på tentan.

Bonuspoäng från labbarna (max 6) läggs till del II-resultatet innan betyget sätts.

del I

(6p) **1.** Rita ett UML-klassdiagram som åskådliggör följande klasser och interface. Alla relationer ska vara med. För full poäng måste öppna och slutna pilspetsar samt streckade och heldragna linjer användas rätt. Fyllda romber behöver *inte* användas. Instansvariabler och metoder behöver *inte* vara med i diagrammet men ev. relationer som ges av variabler och metoder ska ritas ut. Alla klasser och interface utom ArrayList måste vara med i diagrammet. Långa namn får förkortas om det är helt klart vilket namn som avses.

```
interface Moderator { ... }
interface Wizard { ... }

abstract class AbstractMember {
  Moderator myMod;
  AbstractMember (Moderator m) { ... }
  ...
}

class Member extends AbstractMember { ... }
class PlusMember extends Member implements Wizard { ... }
class GuestMember extends Member { ... }

class Organizer implements Moderator {
  ArrayList<PlusMember> plusmembers;
  ArrayList<GuestMember> guests;
  ....
}

class Overseer implements Moderator { ... }
  PlusMember chair;
  ArrayList<Member> members;
  ...
}
```

(1p) **2.**Vilket designmönster används i uppgift 1?

(3p) **3.** Nedan följer fem korta beskrivningar av designmönster. Välj ut vilken beskrivning som stämmer för var och en av

Template Method **Observer** **Proxy**

Två beskrivningar blir alltså över!

A) Ett objekt kontrollerar åtkomsten till ett annat objekt. De båda implementerar samma gränssnitt.
B) En enda klass ger ett gränssnitt till ett komplext system.
C) Skjut upp delar av en algoritm till subklasser.
D) När tillståndet i ett objekt ändras så uppdateras ett antal andra objekt som prenumererarpå uppdateringsinfo.
E) En trädstruktur där löv och sammansatta objekt har samma interface och kan behandlas lika.

(2p) **4.** Vad menas med "Mock Object" och varför är en sådan teknik bra att använda ?

(2p) **5.** Ange fyra karakteristiska drag för *agila/lättviktiga* metodiker för programutveckling eller ange fyra av XP-reglerna som används inom *eXtreme Programming*.

6. Förutsätt följande klass- och interface-deklarationer i Java
("insidan" är oväsentlig och markeras som `...`).

```
interface C {...}

class A {...}

class B extends A implements C {...}
```

(1p) **a.** Vilken eller vilka typer av referenser kan referera till objekt av klassen `A`?

A) endast typen `A`
B) endast typerna `A` och `Object`
C) endast typerna `A` och `B`
D) endast typerna `A` och `C`

(1p) **b.** Vilken eller vilka typer av referenser kan referera till objekt av klassen `B`?

A) endast typen `B`
B) endast typerna `B` och `A`
C) endast typerna `B`, `A` och `C`
D) endast typerna `B`, `A`, och `Object`
E) endast typerna `B`, `A`, `C` och `Object`

(2p) **7.** Om variabeln `a` refererar till en Java-objektsamling, t.ex. en `ArrayList` eller `LinkedList`, så kan man under vissa förutsättningar få samlingen sorterad genom att anropa metoden `Collections.sort(a)`. Vad måste gälla för att sorteringen ska gå att genomföra?

(2p) **8.** Stegtävlingar som motionsaktivitet är populära just nu. Anställda på en arbetsplats delas in i lag, får var sin stegräknare och går och går och går så mycket de orkar och rapporterar varje kväll in dagens antal steg till en server skriven i Java. Ställningen mellan lagen redovisas kontinuerligt på en webbsida. Det lag som först har gått t.ex. hela sträckan från Stockholm till Berlin vinner. Birgit och Cecilia tillhör samma lag och promenerar tillsammans. Birgit går 12536 steg och Cecilia går 11980 steg den 27 maj 2010. De rapporterar in resultaten samtidigt. När båda gjort sina rapporteringar har lagets antal endast ökat med 11980 och inte 24516 som väntat. Vad är det som har hänt och hur ska serverprogammet åtgärdas för att kommande rapporteringar ska bli rätt? Svaret är mycket kortare än frågan. Ett svar med bara ett ord ger 1p (om det är rätt ord förstås).

(4p) **9.** Skriv om följande lilla programavsnitt så att det blir tydligare men har exakt samma funktion. *Koduppprepningen* ska elimineras (eller åtminstone minskas) men det finns fler förbättringar att göra. Full poäng kan ges även om svaret innehåller syntaxfel.

```
if (d==1) {
  if (p!=q) {
    int tmp = rad[q]; rad[q] = rad[p]; rad[p] = tmp;
    move = true;
  }
}
else if (d==-1) {
  if (p!=q) {
    int tmp = rad[q]; rad[q] = rad[p]; rad[p] = tmp;
    move = true;
  }
}
else if (d==2) {
  if (goahead) {
    if (p!=q) {
      int tmp = rad[q]; rad[q] = rad[p]; rad[p] = tmp;
      move = true;
    }
  }
}
else if (d==-2) {
  if (goahead) {
    if (p!=q) {
      int tmp = rad[q]; rad[q] = rad[p]; rad[p] = tmp;
      move = true;
    }
  }
}
```

(1p) **10.** Vad kallas det för när man inom programutveckling förbättrar programkoden utan att den yttre funktionen av programmet ändras?

A) Reuniting **B)** Refactoring **C)** Resolving

del II

11. Uppgiften handlar om en prototyp för spel på en rutindelad spelplan. Prototypen är utan grafik. Klassen `Brade` innehåller en matris av objekt av klassen `Ruta`. Klassen `Ruta` ska ha två instansvariabler av typ `int` som anger rutans position (koordinater) på brädet, samma tal som indexen i matrisen i klassen `Brade`. Index-talen anges när man skapar ett objekt av `Ruta`. Dessutom ska `Ruta` ha egenskapen att när man ”skriver ut” objekt ska index-talen visas:

```
Ruta rutan = new Ruta(2, 5);
Ruta andraRutan = new Ruta(0, 1);
System.out.println(rutan + "  " + andraRutan);
```

ska ge en utskrift liknande `Ruta 2 5  Ruta 0 1`.

(3p) **a)** Skriv klassen `Ruta` så att den stämmer med exemplen ovan.

Klassen `Brade` innehåller en metod för att få en *iterator* horisontellt, vertikalt eller diagonalt över rutorna. Man anger startkoordinater och slutkoordinater och så skapas den rätta iteratorn. Skiss av prototypen för `Brade`:

```
class Brade{
  int n;
  Ruta[][] rutor;

  Brade(int n) { ... } // n*n-matrisen rutor skapas

  II iterator(int i1, int j1, int i2, int j2) { // Skapar iterator
    Ruta r1 = rutor[i1][j1];                    // från ruta (i1,j1)
    Ruta r2 = rutor[i2][j2];                    // till ruta (i2,j2)
    return new Iterator(this, r1, r2);
  }
}
```

Iteratorn kan implementera följande minimi-gränssnitt för iteratorer: Du *får* använda ett annat likvärdigt interatorinterface.

```
public interface II {
    public boolean hasNext();
    public Ruta next();
}
```

Exempel på användning:

```
Brade brade = new Brade(10);
II it = brade.iterator(2,6,5,3);
while (it.hasNext())
  System.out.println(it.next());
```

ger utskriften

```
Ruta 2 6
Ruta 3 5
Ruta 4 4
Ruta 5 3
```

(13p) **b)** Skriv klassen `Iterator`. Kontroller av indata, t.ex. att `(i1,j1)` och `(i2,j2)` anger en horisontell, vertikal eller diagonal riktning och håller sig inom brädets storlek behöver *inte* göras. Iterationen ska kunna göras i åtta olika riktningar: höger vänster, upp, ned och fyra diagonala riktningar. Din lösning måste förstås identifiera vilken riktning det är. Stor vikt läggs vid tydlig och bra struktur. Liten vikt läggs vid att Java-syntaxen är perfekt. Om metoden `next()` anropas efter att iteratorns slut är nått (`hasNext()` har givit `false`) ska iteratorns sista objektet returneras igen.

12. Ett program för glosförhör ska använda engelska och svenska ord. För varje engelskt ord ska ett godtyckligt antal svenska ord (översättningar) kunna lagras, detta utgör en glosa. Ett stort antal glosor ska hanteras av programmet. Glosorna bör lagras med det engelska ordet som nyckel så att när ett engelskt ord är givet ska glosan kunna slås upp mycket snabbt. Det är inte viktigt att glosorna är sorterade eller ens kan sorteras.

(3p) **a)** Föreslå ett sätt att representera en enskild glosa samt en stort antal glosor i ett Java-program. Kraven ovan ska uppfyllas. Skissa den eller de nya klasser som ska definieras och visa vilka bibliotheksklasser som ska användas. Svaret behöver inte *inte* ges som fullständiga Java-klasser, t.ex. konstruktorer behöver inte visas. Det går bra att använda pseudokod i svaret, dvs en kombination av programkod och vanlig text.

(3p) **b)** De gloslistor som är indata till programmet finns på flera olika format. Dels finns vanliga textfiler med olika principer för hur orden avskiljs och hur glosorna avskiljs och dels finns indatafiler på xml-format.

Man ska kunna byta indatafil medan programmet kör. Den del av programmet som läser från fil och bygger upp datastrukturen med glosorna ska vara utbytbar och löst kopplad till själva glosförhöret. Det ska också vara lätt att lägga till nya indataformat. Vilket designmönster bör användas ? Visa ett UML-klassdiagram för mönstret och beskriv kortfattat i ord hur det ska tillämpas. Du får skriva Java-kod men det behövs inte för att få full poäng. Tag med tre olika indataformat i UML-diagrammet. Obs! Inga detaljer om formaten behöver visas, ge dem bara olika namn.

(3p) **c)** Glosprogrammets XML-filer följer följande DTD.

```
<!DOCTYPE Gloslista [
    <!ELEMENT Gloslista (Glosa+)>
    <!ATTLIST Gloslista listid CDATA #REQUIRED>
    <!ELEMENT Glosa (EngOrd,SveOrd+)>
    <!ELEMENT EngOrd (#PCDATA)>
    <!ELEMENT SveOrd (#PCDATA)>
]>
```

Skriv välformad XML-kod enligt denna DTD för en gloslista med minst två engelska ord där åtminstone ett av dem har mer än en översättning. T.ex. har engelska *grade* betydelserna *betyg, klass, sluttning m.m* och engelska *mark* betyder *betyg, poäng, märke m.m.*