

**Objektorienterad Programkonstruktion, DD1346**

Tentamen 2013-03-12, kl. 09.00-12.00

**Tillåtna hjälpmedel:** Papper, penna och radergummi.

**Notera:** Frågorna i del I ska besvaras på för ändamålet lämnad plats i tentamenslydelsen. Frågorna i del II besvaras på separat papper. Använd gärna både fram- och baksida, men behandla högst en uppgift per sida. Kom ihåg att skriva namn och personnummer på alla inlämnade blad. Skriv tydligt!

**Betygsgränser:** Betyg XF:  $\geq 17$ p i del I  
Betyg E:  $\geq 20$ p i del I  
Betyg D:  $\geq 20$ p i del I **och**  $\geq 5$ p i del II  
Betyg C:  $\geq 20$ p i del I **och**  $\geq 10$ p i del II  
Betyg B:  $\geq 20$ p i del I **och**  $\geq 15$ p i del II  
Betyg A:  $\geq 20$ p i del I **och**  $\geq 20$ p i del II

**Ansvarig:** Christian Smith (ccs@kth.se)

*Lycka till!*

---

## Del I - flervalfrågor

1. I denna uppgift finns 8 program(-delar), antingen i form av Java-kod eller i UML-diagram. *Generiska namn används i stället för de mer vanliga namn som avslöjar vilket mönster det är.* För varje program, ange det designmönster som bäst beskriver det. Välj från listan nedan. Varje korrekt angivet designmönster ger 1 p. (8 p)

**Singleton**   **MVC**     **Composite**   **Proxy**   **Adapter**   **Flyweight**  
**Lock**     **Observer**   **Factory**   **Builder**   **Prototype**   **Facade**

a)

```
public class A{

    MyType1 myObject1;
    MyType2 myObject2;

    public A(){
        myObject1 = null;
        myObject2 = null;
    }

    public void setObject1(Object o){
        myObject1 = new MyType1();
        myObject1.doSomeInitialStuff(o);
    }

    public void modifyObject1(Object o){
        myObject1.doSomeShallowStuff(o);
    }

    public void setObject2(Object o){
        myObject2 = new MyType2();
        myObject2.doSomeInitialStuff(o);
    }

    public void modifyObject2(Object o){
        myObject2.doSomeDeeperStuff(o);
    }

    public MyType3 getType3(){
        MyType3 myOutputObject = new MyType3(this);
        return myOutputObject;
    }
}
```

b)

```
public class B{

    MyType1 myObject1;
    MyType2 myObject2;
    MyType3 myObject3;
    MyType4 myObject4;

    public B(){
        myObject1 = new MyType1();
    }

    public int doStuff(Object o){
        myObject1.doSomeInitialStuff(o);
        myObject2 = myObject1.getType1();
        myObject2.setSomeParameters(0,1,2,3,4);
        myObject1.setSomeOtherParameters(myObject2.getSomeParameters());
        myObject3 = new MyType3(myObject1, myObject2);
        myObject4 = new MyType4();
        myObject4.setChildObjects(myObject1,myObject3);
        return myObject4.calculateHairyStuff();
    }

}
```

c)

```
public class C{

    MyType1 myObject1;

    public C(){
        myObject1 = new MyType1();
    }

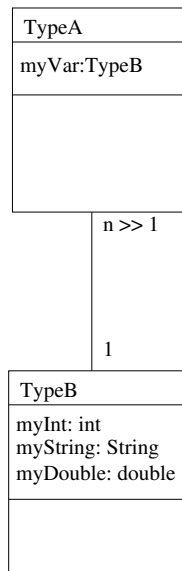
    public int doOtherStuff(float f){
        return myObject1.doMyStuff((double) f);
    }

}
```

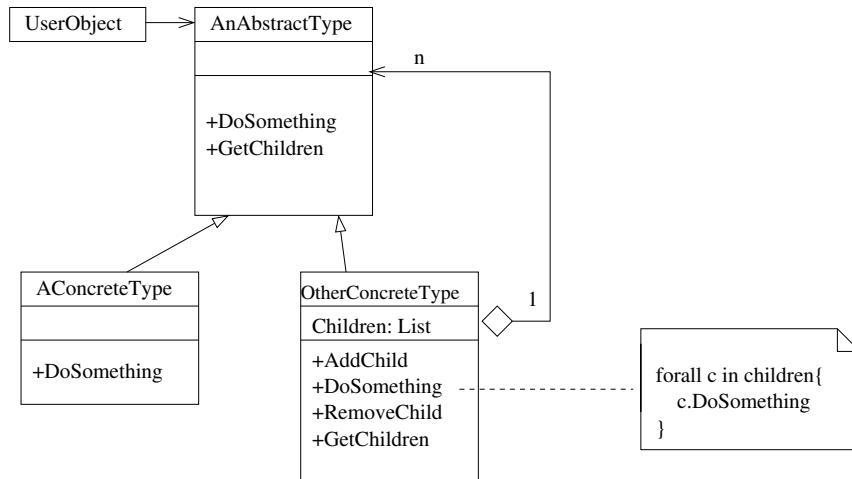
d)

```
public class D{  
  
    public static GenericType getObject(int a){  
        if(decisionFunction(a)){  
            return new SpecificType1();  
        }else{  
            return new SpecificType2();  
        }  
    }  
}
```

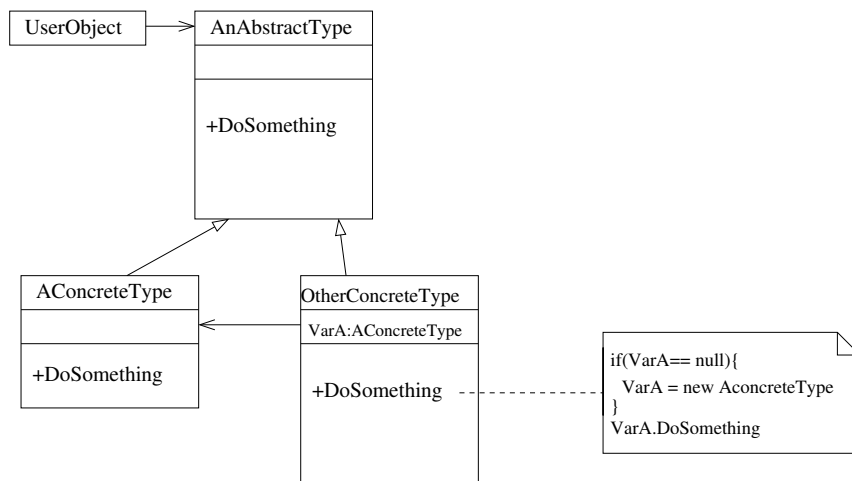
e)



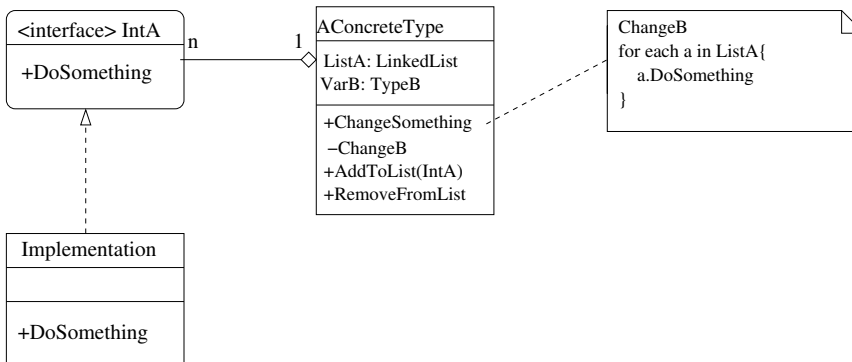
f)



g)



h)



2. Nedan följer 4 exempel på Java-program. För varje program, ange vad programmet skriver ut. Svara antingen med ett heltal mellan 0 och 20, eller ange om det är mindre än 0 eller större än 20. Varje korrekt analyserat program ger 1 p. (4 p).

```
a) public class A2{

    static int a = 0;

    public static void main(String[] args){

        for(int i = 0; i>a--; i++){
            i=++i+a--;
        }

        System.out.println(a);

    }
}
```

```
b) public class B2{

    static int a = 0;

    public static void main(String[] args){

        for(int i = 0; i<20; a++){
            i=a;
        }

        System.out.println(a);

    }
}
```

```
c) public class C2{

    static int a = 0;

    public static void main(String[] args){
        int a = 5;
        doStuff(a);
    }

    private static void doStuff(int a){
        for(int i = 0; i<10; i++){
            a++;
        }
        System.out.println(a);
    }
}
```

```
d) public class D2{

    static int a = 0;

    public static void main(String[] args){

        int a = 5;
        doStuff(a);
        System.out.println(myPrint());
    }

    private static void doStuff(int a){
        for(int i = 0; i<10; i++){
            a++;
        }
    }

    private static String myPrint(){
        return String.valueOf(a);
    }
}
```

3. För varje påstående om trådar, synkronisering och parallella program i Java, ange om det är sant eller falskt. 5 korrekta svar ger 4p, 4 korrekta svar ger 2p, 3 korrekta svar ger 1p, 2 eller färre korrekta svar ger 0 p (4 p)
- a) För att två olika trådar ska kunna anropa samma metod i samma objekt samtidigt, måste metoden vara deklarerad med nyckelordet `synchronized`.
  - b) En `threadpool` används till att minska risken för `deadlock`.
  - c) Ett `Lock` används till att minska risken för `thread interference`.
  - d) En `atomär operation` är säker mot `thread interference`.
  - e) Javakompilatorn kan hitta de flesta situationer där `livelock` uppstår, och ge felmeddelancen eller varningar.
4. För varje påstående om designmönster nedan, ange om det är sant eller falskt. 4 korrekta svar ger 2p, 3 korrekta svar ger 1p, 2 eller färre korrekta svar ger 0 p (2 p)
- a) I mönstret `Model View Controller` är det vanligt att man kan ha flera `View`-objekt men bara en `Controller`.
  - b) En `Iterator` behövs för att sortera arrayer.
  - c) Med en `Singleton` samlar man ihop ett större antal klasser och metoder till ett gemensamt objekt med en (eller ett fåtal) enkla metदानrop.
  - d) `Prototype`-mönstret kan användas internt i en `Factory`-metod för att skapa nya objekt.
5. För varje påstående om gränssnitt (`interface`) i Java nedan, ange om det är sant eller falskt. 4 korrekta svar ger 2p, 3 korrekta svar ger 1p, 2 eller färre korrekta svar ger 0 p (2 p)
- a) Gränssnitt specificerar hur en viss metod ska implementeras.
  - b) Utan gränssnitt skulle inte ett objekt av en klass kunna anropa en metod i ett objekt av en annan klass.
  - c) Gränssnittet `Comparable` behövs för att använda Javas inbyggda sorteringsalgoritmer.
  - d) Om en klass `A` ärver en annan klass `B`, och `B` implementerar gränssnittet `C`, så implicerar det att `A` också implementerar `C`.



6. Nedan följer 5 kodexempel i Java. För varje kodexempel, ange om koden går att kompilera. Eventuella fel kan förväntas finnas inom den listade koden, dvs du kan anta att alla `import`-påståenden är korrekta, och att alla anrop av externa metoder fungerar som man förväntar sig. Varje korrekt analyserad programkod ger 1p. (5 p)

a) `public class A6 implements Comparable<A6>{`

```
    private int a = 1;

    public int compareTo(A6 other){
        return this.a - other.a;
    }
}
```

b)

`public class B6{`

```
    private int a = 1;

    public static void main(String[] args){
        System.out.println(a);
    }

}
```

c) `public class C6{`

```
    public String myHej;

    public static void main(String[] args){
        C6 myC6;
        myC6.myHej = "Hej";
        System.out.println(myC6.myHej);
    }

}
```

```
d) import java.util.*;

public class D6{

    public static Object[] getArray(Iterable i){

        ArrayList<Object> myList = new ArrayList<Object>();

        for(Object o : i){
            myList.add(o);
        }

        return myList.toArray();
    }
}
```

```
e) import java.util.*;

public class E6 extends Date{

    public static E6 myE6;

    public static void assignDate(Date d){
        myE6 = d;
    }

}
```

## Del II - fördjupningsfrågor

Följande uppgifter besvaras på separat papper.

7. a) Förklara begreppet polymorfism, och på ge exempel på hur polymorfism kan underlätta för programmerare. (2 p)
- b) Hur hanterar Java polymorfism? Vilka är tillåtna typer för att deklarera resp. instansiera variabler med? Vilka metoodanrop är tillåtna? Hur/när avgörs vilken metod som ska anropas? (3 p)

8. Antag att du har fått i uppgift att skriva ett program för att simulera partikelflöden i rörsystem. I programmet ska man kunna rita upp sina rörsystem, bestående av sammansättningar av färdiga rördelar som användaren kan välja från ett bibliotek. Man ska kunna modifiera de färdiga rördelarnas sammansättning och egenskaper, och kunna spara både rördelar, delsystem och kompletta rörsystem till fil för att återanvända senare.

Du har hittat en uppsättning förenklade differensekvationer för beräkning av partiklarnas rörelser i diskret tid, som du tänker använda tills din kollega har härlett något bättre. I din förenklade modell kan varje partikel uppdateras oberoende av de andra i varje steg, men man måste ha med väldigt många partiklar för att få realistiska resultat. Efter att en simulering har körts ska man dels kunna spela upp en animering av flödet, och dels kunna visualisera olika statistiska mått, som t.ex lokala partikeltätheter.

Beskriv en bra struktur för programmet. Förklara vilka designmönster som används, i grova drag hur de implementeras (vilken information finns var, hur kommunicerar programmets olika delar med varandra, hur styrs olika programflöden, osv). Motivera varför din lösning är bra! För full poäng skall hänsyn ha tagits till alla delar som beskrivs ovan. Du får använda UML eller (pseudo-)kod i ditt svar om du tycker att det förenklar presentationen, men det är inte ett krav. (10 p)

9. Förklara hur man kan använda **Observer** i MVC (2 p)
10. Förklara skillnaden mellan ett vanligt **Observer**-mönster och en **publisher/subscriber**. Vilka fördelar har det senare? (3 p)
11. XML kan ibland vara ett bra format för att lagra data, men inte alltid. Förklara när det är bra, och när det passar dåligt, och ge exempel för båda fallen (2 p)

12. En vän ringer dig och ber dig hjälpa till med att fixa ett problem i ett program. Hen har försökt skriva om ett Matlab-program till Java, men något går snett, och beräkningarna ger helt andra resultat i Java-versionen. Du föreslår att hen ska skriva ut alla delresultaten av en körning i både Matlab- och Java-versionen, jämföra dessa och på så sett hitta var det börjar divergera. Efter en halvtimme ringer hen igen och säger att hen gjort det men inte kan se att resultaten skiljer sig åt någonstans, inte ens slutresultatet. Det var kanske inget fel i alla fall. 10 minuter senare ringer telefonen igen. Nu har hen tagit bort alla störande utskrifter, men nu fungerar det inte längre. Vad är det (troligen) som händer? Vad kallas fenomenet och hur kan det uppstå? (3 p)