Skolan för datavetenskap och kommunikation

Objektorienterad Programkonstruktion, DD1346

Tentamen 2016–06–11, kl. 9.00–12.00

Tillåtna hjälpmedel: Papper, penna och radergummi.

Notera: Frågorna besvaras på separat papper. För del I kan flera frågor behandlas på samma blad, men behandla högst en uppgift per sida i del II. Kom ihåg att skriva namn och personnummer på alla inlämnade blad. Skriv tydligt!

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

Ansvarig: Christian Smith (ccs@kth.se)

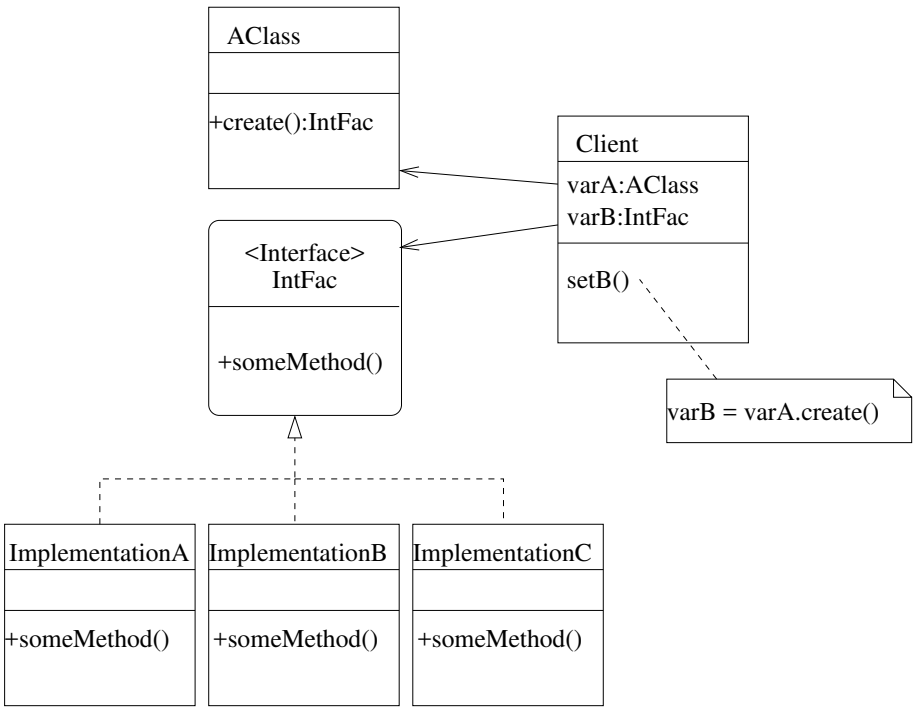
Lycka till!

Del I - flervalsfrågor

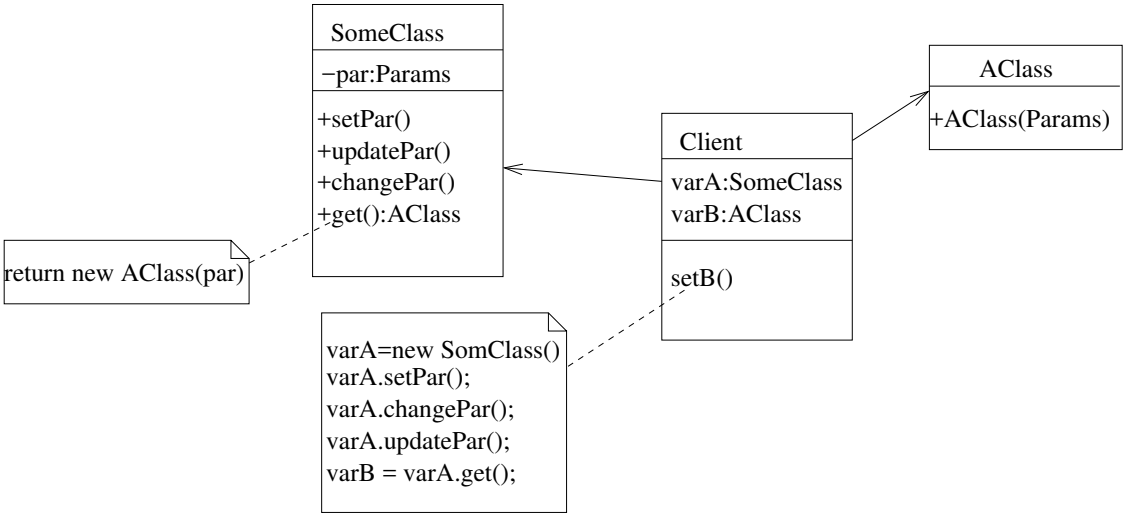
1. I denna uppgift finns 5 uppsättningar UML-diagram. *Generiska namn används i stället för de mer vanliga namn som avslöjar vilket mönster det är.* För varje diagram, ange det designmönster som bäst beskriver det. Välj från listan nedan. Varje korrekt angivet designmönster ger 1 p. (5 p)

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

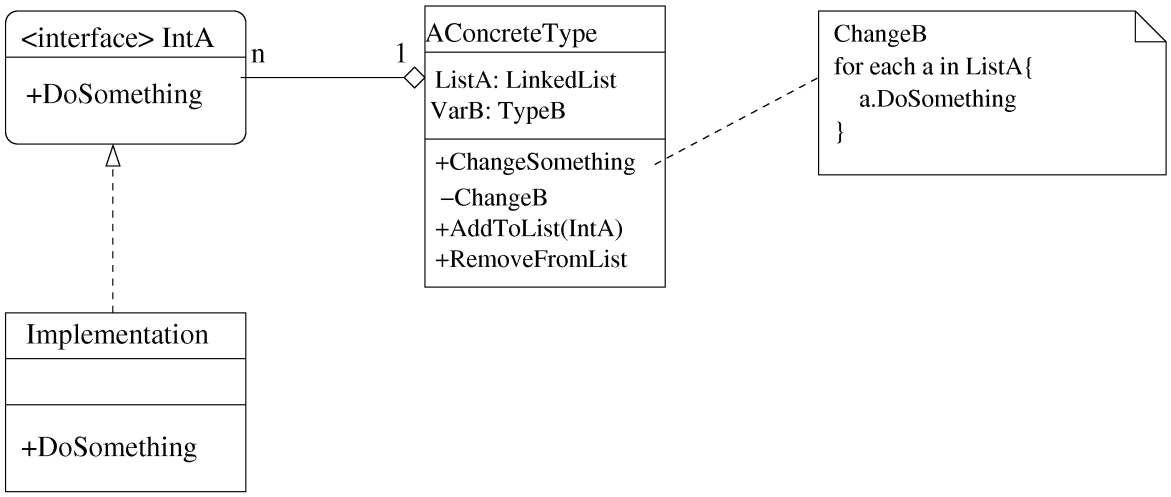
a) Factory



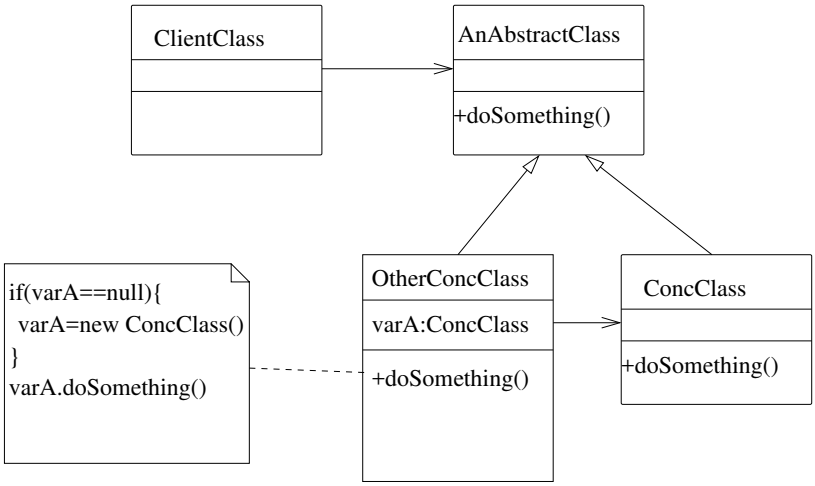
b) Builder



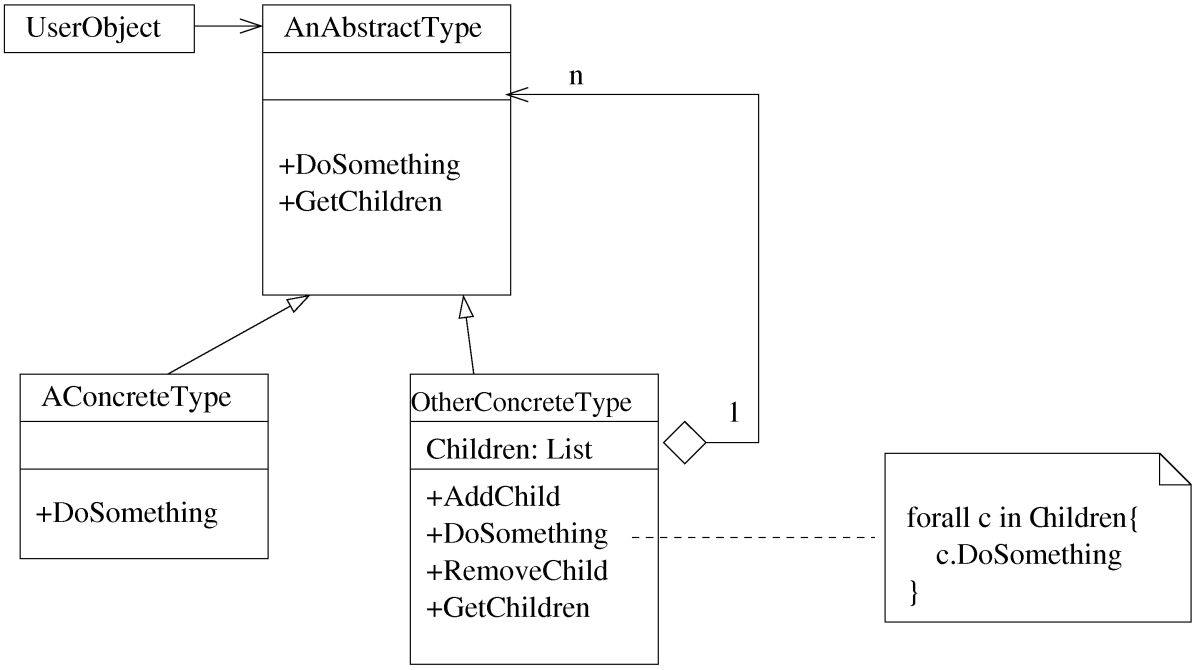
c) Observer



d) Proxy



e) Composite



2. Här följer 5 st kodlistningar i Java[1]. För varje kodlistning, ange om den går att kompilera eller inte. Anta att varje klass **X** finns i en fil som heter '`X.java`', och kompileras med kommandot '`javac X.java`'. (5 p)

a) Kompilerar EJ

```
public class A{

    public static void main(String[] args){
        Aint myA = new A();
    }

    static class Asub extends A implements Aint{}

    interface Aint{}

}
```

b) Kompilerar

```
public class B{

    public static void main(String[] args){
        Bint myB = new Bsub();
    }

    static class Bsub extends B implements Bint{}

    interface Bint{}

}
```

[1]För tydlighets skull anses här Java 8 som finns i skolans datasalar

c) Kompilerar EJ

```
public class C{

     public static void main(String[] args){
        C myC = new Cint();
    }

    static class Csub extends C implements Cint{}

    interface Cint{}

}
```

d) Kompilerar

```
public class D{

    public static void main(String[] args){
        Dint myD = new Dint(){};
    }

    static class Dsub extends D implements Dint{}

    interface Dint{}

}
```

e) Kompilerar EJ

```
public class E{

    public static void main(String[] args){
        Esub myE = new E();
    }

    static class Esub extends E implements Eint{}

    interface Eint{}

}
```

3. Här följer 5 st kodlistningar i Java[2]. För varje kodlistning, ange vad den skriver ut. Om den inte ger en deterministisk utskrift, ange vad som är möjliga utskrifter, t.ex "programmet skriver antingen ut talet 5 eller ingenting alls" eller "ett heltal mellan 1 och 9". Anta att varje klass **X** finns i en fil som heter '`X.java`', och kompileras med kommandot '`javac X.java`', och körs med '`java X`'. (5 p)

   a) Utskrift: ett heltal ≥ 1000. Hypotetiskt sett, men extremt osannolikt, så skulle värdet på `a` kunna bli större än maxvärdet på en int, och alltså 'slå runt' och bli ett negativt heltal, så svaret ett heltal mellan -2^{31} och $2^{31} - 1$ är också godtagbart.

```
public class A{

    private static int a = 0;

    public static void main(String[] args){
       Thread myThread = new Thread(new Asub());
       myThread.start();
       while(a<1000){}
       System.out.println(a);
    }

    static class Asub implements Runnable{

        public void run(){
            while(true){
                a++;
            }
        }
    }

}
```

[2]För tydlighets skull anses här Java 8 som finns i skolans datasalar

b) Utskrift: ett heltal ≥ 0, eller ingenting alls. Hypotetiskt sett, men extremt osannolikt, så skulle värdet på `a` kunna bli större än maxvärdet på en int, och alltså 'slå runt' och bli ett negativt heltal, så svaret ett heltal mellan -2^{31} och $2^{31}-1$, eller ingenting alls är också godtagbart. Notera att **det utskrivna talet kan vara större än 1000**.

```
public class B{

    private static int b = 0;

    public static void main(String[] args) throws InterruptedException{
        Thread myThread = new Thread(new Bsub());
        myThread.start();
        if(b>=1000){
            myThread.join();
        }
        System.out.println(b);
    }

    static class Bsub implements Runnable{

        public void run(){
            while(true){
                b++;
            }
        }
    }

}
```

c) Utskrift: 0

```
public class C{

    int c = 0;
    static double cd = 0;

    public static void main(String[] args){

        C myC = new C();
        int c = 2;

        myC.inc();

        System.out.println(myC.c);
    }

    private void inc(){
        c += cd++;
    }


}
```

d) Utskrift: ett flyttal mellan 0.0 och 0.1

```
public class D{

    public static void main(String[] args){

        D myD = new Dsub();
        myD.printD(0.1);

    }

    private static void printD(double d){
        System.out.println(d*Math.random());
    }

    static class Dsub extends D{

        private static void printD(double d){
            System.out.println((int)(d*Math.random()));
        }
    }

}
```

e) Utskrift: -1

```
public class E{

    public static void main(String[] args){

        try{
            System.out.println(args[0]);
        }catch(Exception e){
            System.out.println("-1");
        }
    }

}
```

4. För varje påstående om designmönster, 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) Mönstret `Factory` är ett sätt att åstadkomma lös koppling. Sant

   b) Mönstret `Prototype` har som främsta syfte att ge snabbare exekvering. Sant

   c) Mönstret `Threadpool` har som främsta syfte att motverka deadlock. Falskt

   d) Mönstret `Singleton` har som främsta syfte att motverka polymorfism. Falskt

   e) Mönstret `Iterator` har som främsta syfte att ge snabbare exekvering. Falskt

5. För varje påstående om åtkomst i Java, 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) En inre klass har tillgång till fält och metoder som deklarerats som `private` i den omgivande klassen. Sant

   b) Två olika instanser av samma klass har tillgång till varandras privata fält och metoder. Sant

   c) `protected` ger en mer restriktiv åtkomst än `package private`. Falskt

   d) Ett fält som deklarerats som `static` i klass **A** kan inte ändras från instanser av klass **B**, som ärver från **A**. Falskt

6. För varje påstående om gränssnitt 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) Om klassen **A** ärver från klassen **B**, och **B** implementerar gränssnittet **C**, så kan man alltid anta att **A** också implementerar **C**. Sant

   b) Ett gränssnitt som inte innehåller vare sig metoder eller fält implementeras implicit av alla klasser. Falskt

   c) Gränssnitt kan inte implementeras av abstrakta klasser. Falskt

   d) Det är inte tillåtet att använda gränssnitt som returtyp för en metod. Falskt

   e) Det är inte tillåtet att låta en klass ärva från ett gränssnitt. Sant

Del II - fördjupningsfrågor

Följande uppgifter besvaras på separat papper.

7. Förklara följande begrepp: **polymorfism**, **lös koppling** och **designmönster**. (3 p)

8. Vad är en **profilerare**? När, hur och varför används den? (3 p)

9. Det har visat sig att i den version av Java som finns på ditt företags nya serie av smarta kameradrönare så saknas en massa funktionalitet, och av licensskäl kan ni inte använda existerande externa bibliotek. Du får i uppgift att skriva ett bibliotek för kartläggning och positionsberäkningar. Ditt bibliotek ska stödja sparade trajektorier[3] av godtycklig storlek, samt ett stort antal olika operationer, som t.ex avståndsberäkningar, areaberäkningar, hastighetsuppskattningar, samt geometriska beräkningar av kamerapositioner, vinklar och så vidare.

   Din uppdragsgivare säger att beräkningsprestanda inte är viktigt, eftersom man räknar med att kunna licensiera ett effektivt bibliotek till generation två av produktserien när de mer krävande funktionerna skall lanseras, men man vet ännu inte vilket bibliotek det kommer att bli. Man behöver dock ett fungerande bibliotek för stunden, för att funktionaliteten i generation ett ska bli komplett.

   Beskriv hur du tänker när du utformar ditt bibliotek. Hur representerar du datatyperna? Hur gör du med klasshierarkier? Vilka designmönster kan vara tillämpliga, och hur? Vad bör du tänka på inför framtiden? (7 p)

10. Vad är **designmönster**, och hur skiljer de sig från **algoritmer**? Är det nödvändigt att vara expert på båda två för att skriva bra objektorienterade program? Motivera! (2 p)

11. En vän har kontaktat dig och bett om hjälp med ett problem i Java-kod som hen har skrivit. Hen har skrivit ett program med ett GUI gjort i Swing, med bland annat ett par `JButton`-knappar som heter "start" respektive "stop". När man trycker på "start"-knappen så startar en animation av en lavalampa i ett fönster, men det går inte att stoppa med "stopp"-knappen, eftersom programmet inte längre bryr sig om några knapptryckningar alls efter att man tryckt på "start". Alla andra knappar, som t.ex de som byter färg, tänder/släcker lampan eller som sparar bildfiler av hur lampan ser ut, fungerar i övrigt, så länge man inte startar animationen. Du har inte tid att titta igenom källkoden just nu, men ska ändå ge en trolig förklaring till vad som är felet och hur vännen bäst löser det. (4 p)

12. Hur fungerar designmönstrena **Flyweight** och **Prototype**? Vilka likheter har de, och vilka skillnader? Hur kan man implementera dem? Rita en UML och förklara! (6 p)

[3] En trajektoria består av en serie positions- och riktningsangivelser, som uppdateras löpande, med upp till hundra nya värden i sekunden