

# Tentamen i Grundläggande programmering STS, åk 1 fredag 2003-05-23

**Skrivtid:** 09.00 – 14.00

**Hjälpmedel:** Inga

**Lärare:** Anders Berglund. Anders besöker tentan vid två tillfällen: cirka kl. 10.30 samt cirka kl. 12.00.

**Observera:**

- Programkod ska skrivas enligt god programmeringssed för Java och objektorienterad programmering.
- Uppgifterna är inte ordnade efter svårighetsgrad.
- Läs igenom hela tentan först.
- Kom ihåg att fylla i försättsbladet ordentligt.
- Om du är osäker på någon fråga eller uppfattar någon uppgift som oklar, gör ett antagande som du motiverar och förklarar väl.
- På de uppgifter där du ska skriva programkod, kan du också skriva en förklaring, om du inte är säker på att din lösning är riktig. En sådan ger visserligen inga poäng, men kan hjälpa till vid rättningen att förstå tankarna bakom en lösning
- Skriv programkod rimligt indenterad och med bra variabelnamn. På så sätt underlättas rättningen.
- 50% av tentans totala poängantal garanterar betyget godkänt
- Skriv endast en uppgift per blad och skriv bara på ena sidan av papperet, dvs. ett nytt blad för varje uppgiftsnummer, medan deluppgifter (a, b, c etc.) får skrivas på samma blad.
- Fyll i försättsbladet ordentligt!
- Skriv namn på alla papper.
- Referera inte mellan olika svar.

*Lycka till!*

# FÖRSÄTTSBLAD

## Grundläggande programmering, STS, 23 maj 2003

Denna sida ska ifyllt placeras först bland de papper du lämnar in

Uppgift	Besvarad ange i förekommande fall vilka deluppgifter du har löst	Poäng
1		
2		
3		
4		
5		
6		
7		
8		
9		
bonus	<i>ifylles av institutionen</i>	

Namnteckning: \_\_\_\_\_

Namn: \_\_\_\_\_

Personnummer: \_\_\_\_\_

### 1. Ord, bara ord (0,5 + 0,5 + 1 + 1 + 1)

Förklara kortfattat men precist följande termer:

- syntax (i samband med programmeringsspråk)
- semantik (i samband med programmeringsspråk)
- objekt (i samband med programmeringsspråk)
- klass (i samband med programmeringsspråk)
- inkapsling (i samband med objektorienterad programmering)

### 2. Klassvariabel (2)

Förklara kortfattat och koncist begreppet *klassvariabel*. Vad är syftet med begreppet? Ge ett relevant exempel (såsom kod, som kan vara syntaktiskt felaktig utan du för den skull får poängavdrag) eller noggrant i ord.

### 3. Starta din Bil (3 + 2 + 3 + 3 + 4 + 3)

Studera klassen `Person`, som finns i bilagan till denna tenta. `Person` får inte modifieras i denna uppgift. I denna uppgift, som gäller bilar med motorer och förare, är en förare en sorts person som känner till en bil. En bil har en motor.

- Skriv klasshuvuden för de tre klasserna `Forare`, `Bil` och `Motor`. Skriv också de variabeldeklarationer som anger de samband mellan klasser/objekt som anges ovan.
- Klassen `Motor` har en metod `public void starta()`. Denna metod skriver ut texten "startar". Skriv metoden.
- Varje bil har ett bilmärke. Bilmärket kan vara olika för olika bilar och anges när bilen skapas. Skriv den eller de metoder (inklusive eventuella konstruktörer) och/eller deklaration(er) som du anser behövs för att kunna skapa en bil av ett visst märke.
- Skriv en metod `starta` till klassen `Bil`. Metoden skriver ut bilens märke, samt anropar därefter `starta` i bilens motor.
- Skriv en konstruktor för klassen `Forare`. Förare har namn. Du ska på ett relevant sätt använda dig av att en förare är en sorts person.
- Skriv ett "huvudprogram" i klassen `Kora`. Huvudprogrammet ska "fungera som" det anges i denna text, men kan därutöver vara mycket enkelt, vad gäller formateringar av utskrifter etc. Programmet ska på ett relevant sätt använda sig av (objekt ur) klasserna som diskuteras i uppgiften, dvs. `Bil`, `Motor` och `Forare`. Huvudprogrammet ska skapa två bilar, en BMW och en Ferrari (båda med motorer), samt två förare, en till vardera bilen. Se till att förarna verkligen känner till sina respektive bilar. Om du anser att det kräver nya metoder i någon klass, måste du skriva dessa. Du avgör själv vad förarna heter.

*forts på nästa sida*

*forts från föregående sida*

- g) Skriv i samma huvudprogram de rader som krävs för att följande text ska skrivas ut, två gånger, en gång för vardera föraren:

<förarens namn> kör iväg. Hans <bilmärke> startar

<förarens namn> är en egenskap hos föraren och ska behandlas på så sätt. Bilmärket ska skrivas ut av den metod som tillhör en viss förarens bil. Utnyttja att en förare känner till sin bil. ”Startar” ska skrivas ut av varje enskild bils motor.

#### 4. Flax (3)

Fundera på vad du vet om fåglar.

- Ange tre klasser, som sinsemellan har ett samband av typen *är-en-sort* och som på något sätt är relevant för att beskriva några fåglar och samband mellan typer av fåglar. Ledning: Låt en av klasserna vara ”Fågel”. Du behöver enbart ange klassnamn i denna uppgift.
- Utöka din modell från a) med relationer av typen *känner-till* och *har-en*. Ledning: Du kan behöva införa nya klasser och/eller objekt ur klasser för att din beskrivning ska vara relevant och rimlig.

#### 5. Publikt och privat (2)

Metoder och data i objekt kan vara deklarerade privat eller publikt. Vad menas med dessa två begrepp? Varför används de? Förklara kortfattat men exakt, gärna med ett exempel.

#### 6. Integritet (3)

Beskriv en situation där det vid en IT-användning sker en integritetskränkning. Vilket är det moraliska problemet? Hur kan kränkningen undvikas?

#### 7. Rekursion (3)

Studera nedanstående program. Ange exakt vad utskriften blir, när programmet körs.

```
public class GorVadKlass {
    public static int gorVad(int x, int n) {
        if (n == 0) return 1;
        return x * gorVad(x, n - 1);
    }

    public static void main(String[] args) {
        for (int i = 3; i <= 6; i++) {
            Std.out.println(gorVad(i, i + 1));
        }
    }
}
```

## 8. *Gå på banken (2)*

Klassen `Bankkonto` har instansvariablerna `int nr` och `double saldo` samt metoderna `public int getNummer()`, `public void sättIn(double summa)`, `public void taUt(double summa)` och `public void saldobesked()`. `main`-metoden finns nedan. När programmet kompileras får man ett kompileringsfel i `main`. Visa vad som behöver ändras och förklara i ord varför det inte fungerar som det står nu.

```
public static void main(String [] args) {
    Bankkonto b1;
    b1.sättIn(3000);
    b1.saldobesked();
    b1.taUt(2500);
    b1.saldobesked();
}
```

## 9. *Fransk lyrik (8)*

Anna, som gillar att köpa böcker om fransk lyrik, men inte har tid att läsa alla (studierna vid STS är alltför betungande), vill ha bättre koll på sina nyinköpta böcker. Bokhyllan, som böckerna ska stå i, rymmer maximalt 50 böcker (oavsett storlek ☺). För att böckerna ska stå ordentligt är de sorterade efter sidantal, med de tunnaste först. Anna, som också är road av programmering, vill använda sin dator. Hon har redan skrivit klassen `Bok` (se nedan för skelettet till klassen) samt börjat skriva klassen `Bokhylla`. Hon har i `Bokhylla` skrivit `public void sättInBok(Bok b)` som sätter in en bok på rätt plats i bokhyllan. Metoden förutsätter att det finns plats i bokhyllan. Hon har också skrivit metoden `ut()` som skriver ut hela bokhyllainnehållet.

Din uppgift är nu att skriva färdigt `Bokhylla`. För det ska följande deluppgifter lösas:

- Skriv klasshuvudet för `Bokhylla` med eventuell(a) instansvariabel(ler) (såsom plats för böckerna).
- Skriv metoden `public void taUtBok(Bok b)`. Boken ska tas bort från hyllan och de övriga böckerna ska tryckas ihop för att stå stadigt.
- Skriv metoden `public boolean finnsMerPlats()` som avgör om det finns plats för ytterligare minst en bok i hyllan.
- Skriv en `main`-metod som testar klasserna genom att skapa ett par böcker och sätta in dem i bokhyllan med hjälp av `sättInBok()`. Metoden ska också skriva ut innehållet i bokhyllan, samt ta bort böcker, m h a `taUtBok()`. Därefter skrivs innehållet i bokhyllan ut igen. Vid något tillfälle ska det dessutom skrivas ut om det finns plats kvar eller inte. Dock behöver programmet inte testa om det finns plats kvar, innan böckerna sätts in. (`finnsMerPlats()` är tänkt för framtid bruk. Än så länge har Anna bara påbörjat sin samling, och har bara några få böcker i sin stora hylla.

Notera! Eftersom Anna har arbetat med denna klass, och skrivit kod för de metoder som anges ovan, fast koden inte redovisas här på tentan, får du anta att dessa metoder finns och fungerar som de bör.

```
import extra.*;
public class Bok {
    private String titel;
    private String författare;
    private String isbn;
    private int antal sidor;

    public Bok() här finns programkod

    public Bok(String title, String forf, Sting nr, int sidor) här finns programkod

    public String getTitel() här finns programkod
    public String geFörfattare() här finns programkod
    public String geIsbn() här finns programkod
    public int geSidantal() här finns programkod

    public void Ut() programkod, skriver ut info om en bok
    public void nyBok() programkod, matar in info om en ny Bok i dialog med användaren från tangentbordet

}
```

## Bilaga till uppgift 3, klassen Person

```
import extra.*;
public class Person {
    private String namn;
    private String adress;
    private int ålder;

    public Person(String n) {
        namn = n;
        adress = "";
        ålder = 0;
    }

    public Person(String n, String a, int å) {
        namn = n;
        adress = a;
        ålder = å;
    }

    public void fyllaÅr() {
        ålder = ålder + 1;
    }

    public void flytta(String nyAdress) {
        adress = nyAdress;
    }

    public String geNamn(){
        return namn;
    }

    public String geAdress(){
        return adress;
    }

    public int geÅlder() {
        return ålder;
    }
}
```