

TENTAMEN: Objektorienterad programmering

Läs detta!

- *Uppgifterna är inte ordnade efter svårighetsgrad.*
- Börja varje uppgift på ett nytt blad.
- Skriv ditt namn i rutan ovan samt namn och personnummer på varje blad (så att vi inte slarvar bort dem).
- **Skriv rent dina svar. Oläsliga svar r ä t t a s e j!**
- Programkod som finns i tentamenstesen behöver ej upprepas.
- Programkod skall skrivas i Java 5 och vara indenterad och kommenterad.
- Onödigt komplicerade lösningar ger poängavdrag.
- Givna deklARATIONER, parameterlistor etc. får ej ändras.
- Läs igenom tentamenstesen och förbered ev. frågor.

I en uppgift som består av flera delar får du använda dig av funktioner klasser etc. från tidigare deluppgifter, även om du inte löst dessa.


Lycka till!

Uppgift 1

Välj **ett** alternativ för varje fråga! Garderingar ger noll poäng. Inga motiveringar krävs. Varje korrekt svar ger två poäng. *Besvara direkt i tesen!*

1. Ett av villkoren är falskt, vilket?


```
Integer i1 = new Integer(123), i2 = new Integer(123), i3 = i1;
```

- a. `i1.equals(i2)`
- b. `i1 == i3`
-  c. `i1 == i2`
- d. `i1.intValue() == i2.intValue()`

2. Vilket påstående är korrekt om följande Javaklass?

```
public class C {  
    private int x = 123;  
    private static int y = 456;  
  
    public void f() { ... }  
    public static void g() { ... }  
};
```


Static variable associeras med en klass och inte som vanliga instance variable med en objekt som skapas av en klass. De är Global variable.

-  a. `f` får anropa `g` och ändra `x` och `y`
- b. `g` får anropa `f` och ändra `x` och `y`
- c. `g` får ändra `x`
- d. `g` får läsa av men ej ändra `x`
- e. `g` får anropa `f`

Static method can ONLY access static variable !


Static method can ONLY call static method !

3. Om A är basklass till B och B basklass till C och variablerna `a`, `b` och `c` har typerna A, B resp. C, vilken tilldelning är då tillåten i Java?

- a. `b = a;`
- b. `c = b;`
-  c. `b = c;`
- d. `c = a;`

Sub till super vehicle = car ;

4. En klass som präglas av hög kohesion

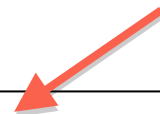
- a. har många abstrakta metoder
- b. har inga abstrakta metoder
- c. kan ha många olika ansvarsområden
-  d. har ett väl avgränsat ansvarsområde

5. Ett av kodavsnitten ger ett exekveringsfel, vilket?

```
a)  
C obj1;  
C obj2 = new C();  
obj1 = obj2;  
obj1.f();
```

```
b)  
C obj1 = new C();  
C obj2 = obj1;  
obj1 = null;  
obj2.f();
```

```
c)  
C obj1;  
C obj2 = new C();  
obj2 = obj1;  
obj2.f();
```



(10 p)

Uppgift 2

Utvidga klassen `ArrayList<Integer>` genom att definiera subklassen `MyArrayList` med metoden

```
public MyArrayList reverse()
```

som arrangerar om elementen i listan baklänges. Ex.

```
MyArrayList list = new MyArrayList();  
for ( int i = 0; i < 5; i++ )  
    list.add(i);  
  
list.reverse();  
// elementen i list skall nu ligga i omvänd ordning  
// t.ex. 4, 3, 2, 1, 0
```

Observera att metoden skall returnera en referens till den förändrade listan. Man skall alltså kunna skriva t.ex. `list.reverse().add(x)`;

(8 p)

Uppgift 3

En båtförsäkringsbolag har ett register över försäkrade fritidsbåtar. Registren har utvecklats och handhas av två olika avdelningar, en för motorbåtar och en för segelbåtar. I samband med en omorganisation vill man samordna registren till ett enda. I det gamla motorbåtsregistret finns bl.a. klasserna

```
public class Motorbåt {
    private String namn;
    private int längd;
    private int motorstyrka;
    private int maxfart;

    public void skrivInfo() {
        System.out.println( "Namn: " + namn + "\nLängd: " + längd +
            "\nMotorstyrka: " + motorstyrka + "\nMaxfart: " + maxfart );
    }
}

public class Motorbåtsregister {
    private ArrayList register;
    public void addera( Motorbåt b ) { ... }
    public void skrivInfo() { ... }
}
```

och i segelbåtsregistret klasserna

```
public class Segelbåt {
    private String namn;
    private int längd;
    private int segelyta;

    public void print() {
        System.out.println( "Namn: " + namn + "\nLängd: " + längd +
            "\nSegelyta: " + segelyta );
    }
}

public class Segelbåtsregister {
    private ArrayList register;
    public void addera( Segelbåt b ) { ... }
    public void print() { ... }
}
```

Refaktorera klasserna `Motorbåt` och `Segelbåt` m.h.a. arv på lämpligt sätt så att en enda registerklass, `Båtregister`, kan innehålla uppgifter om båda typerna av båtar. Klassen skall ha operationerna `addera` och `skrivInfo`. Förutom de refaktorerade klasserna skall lösningen innehålla en fullständig implementering av `Båtregister`.

(12 p)

Uppgift 4

a) Vad skriver programmet ut? Motivera!

```
public class A {
    public void f() { System.out.println( "A.f" ); }
    public void h() { f(); }
}

public class B extends A {
    public void g() { h(); }
}

public class C extends B {
    public void g( A obj ) { obj.h(); g(); }
    public void f() { System.out.println( "C.f" ); }
}

public class Main {
    public static void main(String[] args) {
        C c = new C();
        c.g( new B() );
    }
}
```

(7 p)

b) Vilka av metodanropen är tillåtna och vilka ger kompileringsfel? Motivera svaren!

```
Base obj1 = new Sub1();
obj1.f();
obj1.g();
obj1.h();
Base obj2 = new Sub2();
obj2.f();
obj2.g();
obj2.h();
Int obj3 = new Sub2();
obj3.f();
obj3.g();
obj3.h();
```

Oberoende vilken new vilken ... kolla bara typen
t.ex Base har bara f() och g()

```
public interface Int {
    public void h();
    public void f();
}

public class Base {
    public void f() {}
    public void g() {}
}

public class Sub1 extends Base {
    public void h() {}
}

public class Sub2 extends Base implements Int {
    public void h() {}
}
```

(9 p)

Uppgift 5

I ett system för hantering av studieresultat finns klassen

```
public class MiniLadok {  
    private ArrayList<Studentprestation> prestationer;  
    ...  
    public TreeMap<String, TreeMap<String, Integer>> // <--returtypen  
        sammanställPrestationer()  
  
    {
```

```
        } ... // övrig funktionalitet i klassen utelämnad  
    }
```

Du kan anta att listan `prestationer` innehåller en osorterad följd av objekt av typen

```
public class Studentprestation {  
    private String kurskod;  
    private int poäng;  
    private String pnr;  
    public String getKurskod() { return kurskod; }  
    public int getPoäng() { return poäng; }  
    public String getPnr() { return pnr; }  
}
```

Skriv färdigt metoden `sammanställPrestationer`. Metoden skall returnera en map som avbildar varje kurskod som hittats i `prestationer` på en annan map som avbildar personnummer på totalt antal avklarade poäng i kursen. Kurskoderna skall lagras i bokstavsordning och för varje kurs skall personnumren lagras i bokstavsordning. Fyll gärna i din lösning direkt i tesen!

(14 p)