

Tentamen ITK:P1

torsdag 2005-11-03

Inga hjälpmedel tillåtna förutom papper och penna.
Skriv tydligt. Texta gärna. 10st frågor, 50 poäng.
Alla svar ska påbörjas på ett nytt blad.

30 poäng ger betyget G, 40 poäng ger VG. Lycka till!

1.)

Ett heltal av typen `int` kan i en klass deklarerats med olika modifierare. Vad blir det för skillnad i synlighet för detta heltal om det i deklarationen står :

- a) **`private`**
- b) **`protected`**
- c) **`public`**

(3 poäng)

2.)

Vad blir det för skillnad för denna heltalsvariabel om den även deklarerats som **`static`** ?

(2 poäng)

3.)

Vad blir utskriften när följande klass körs:

```
public class Fraga3 {  
  
    public static void main(String[] args) {  
        int x = 9;  
  
        for(int i = 9; i >= 0; i--) {  
            System.out.print(i + "*" + x);  
            System.out.println(":\t" + x*i);  
        }  
    }  
}
```

(6 poäng)

4.)

En klass **`Built`** har två specialiseringar:
`House` och **`Bicycle`**

I samtliga tre klasser finns en överskuggning av den **`public String toString()`** som finns deklarerad redan i Javas rotklass **`Object`**. I klassen **`Built`** är den definierad enligt:

```
public String toString() {
    return "This is a built object.";
}
```

De ärvande klasserna ska ha en static-deklarerad heltalsräknare. Denna klassvariabel ska räknas upp i de bägge subklassernas konstruktörer så att deras respektive toString-metoder sedan kan visa information om hur många instanser det finns av respektive klass. Detta kan se ut i stil med:

```
return "This is house number :" + counter;
```

Skissa upp en lösning för en fungerande klasshierarki där klassen *House* har ett flyttal som kan lagra information husets yta och *Bicycle* har en variabel för antalet växlar samt en annan boolesk variabel som anger om instansen är en tandem eller ej. Din lösning kan vara en kombination av klassdiagram, javakod och vanlig beskrivande text.

(7 poäng)

5.)

Varför gör man denna överskuggning i klasshierarkin i den föregående uppgiften och vad är **dynamisk bindning**?

(4 poäng)

6.)

Vad innebär begreppen **selektion** och **iteration**? Med hjälp av vilka konstruktioner byggs dessa i språket Java?

(6 poäng)

7.)

Vad är en **stack** och vad innebär algoritmen **LIFO**? Hur har du använt en stack i Laboration 4?

(4 poäng)

8.)

I Java samlar man ofta ihop data i **array/fält**. Hur fungerar denna konstruktion och hur gör man för att lägga in och plocka ut data när man använder en array/ett fält?

(4 poäng)

9.)

Vad är en **konstruktor**? Vad är en *överlagrad* konstruktor? Hur har du i laborationerna använt konstruktorer?

(6 poäng)

10.)

Beskriv i detalj hur följande klass är uppbyggd. För full poäng krävs även att du beskriver utskriftens struktur.

```
public class Fraga10 {

    public Fraga10() {
        skrivUt();
    }

    public void skrivUt() {
        System.out.println("TempTabell");
        System.out.println("-----\n");
        System.out.println("C\tF");

        for(int i = 10; i<=20; i++) {
            System.out.print(i + "\t");
            System.out.println(Math.round(i * 9.0/5 + 32));
        }

    }

    public static void main(String[] args) {
        new Fraga10();
    }
}
```

(8 poäng)