

## Instuderingsuppgifter läsvecka 1

1.

Ge exempel på vad som skiljer *programming in the large* från *programming in the small*.

2.

Ur användarnas perspektiv är bl.a. korrekthet och robusthet två aspekter på mjukvarukvalité. Vad menas med att koden är korrekt? Vad menas med att koden är robust?

3.

Vad är de viktigaste aspekterna på mjukvarukvalité från utvecklarnas perspektiv?

4.

Stelhet (*rigidity*) och bräcklighet (*fragility*) är två symtom på unken design (*design smell*). Förklara vad dessa båda begrepp betyder.

5.

Ge några orsaker till att en design blir unken?

6.

Ge några exempel på hur man i koden kan se att en design har eller håller på att ruttna.

7.

Hur kan man motverka att en design ruttar?

8.

Vad innebär refaktorisering av ett programsystem?

9.

Varför är det viktigt att ett utvecklingsteam hålla sig till en gemensam kodstandard?

10.

Vad skall man tänka på när man väljer identifierarnamn i sina program?

11.

Vilka är fördelarna med en modulär design?

12.

Vad innebär begreppen *kohesion (cohesion)* och *koppling (coupling)*? Varför är dessa egenskaper viktiga vid programutveckling?

13.

Vilka premisser måste vara uppfyllda för att en modul skall kunna gå att återanvända?

14.

Vad är skillnaden mellan *implementationsarv* och *specifikationsarv*?

15.

Vad är *multipelt arv*?

16.

Arv kan användas för *specialisering* och *generalisering*. Förklara dessa begrepp.

17.

Varför har man i Java valt att inte stödja multipelt implementationsarv? Visa och förklara något problem som kan uppkomma vid multipelt implementationsarv.

18.

Förklara skillnaden mellan *abstrakta klasser* och *interface*. När är respektive mekanism att föredra?

19.

Vilka relationer finns definierade i UML?

20.

Hur definierar UML skillnaden mellan *aggregation* och *komposition*?

21.

Rita ett UML-diagram för att representera nedanstående klasser och interface. Försök att vara så exakt som möjligt.

```
public interface Order { . . . }  
public interface Iterable { . . . }  
public class StandardOrder implements Order, Iterable {  
    private List<ListItem> items;  
    private OrderRespository repos;  
    public LineItems getItems() { . . . }  
}  
public class BusinessOrder extends StandardOrder { . . . }  
public class OrderRespository { . . . }
```

22.

Vad betyder begreppet *alias*? Vad kan det uppkomma för problem när alias används?

23.

Hur definierar man *nya typer* i Java?

24.

Förklara vad en *typ* är?

25.

Vilka förhållanden finns mellan *supertyp* och *subtyp* avseende värden och operationer?

26.

Java är ett *starkt typat språk*. Vad innebär det att ett språk är starkt typat?

27.

Vad innebär *explicit typomvandling*? Vad finns det för faror med explicit typomvandling?

28.

Förklara vad operationen **instanceof** används till?