

TDDC77 - Objektorienterad programmering

Laboration 1

**Institutionen för datavetenskap
Linköpings universitet**



1.1 Att skriva Java-kod

Starta editorn VSCode när du är redo att börja programmera. I ett skalfönster kan det ske som följande:

```
> code NAMNET_PÅ_DIN_JAVA_KLASS.java&
```

Där det står `NAMNET_PÅ_DIN_JAVA_KLASS` ska du skriva in vad du tänkt döpa din Java klass till (namnet ska vara kopplat till det programmeringsproblem som du ska lösa). Observera att filen måste heta samma som klassen/programmet fast med `".java"` på slutet. `&-tecknet` gör att Atom startas i en separat process i bakgrunden och därför inte låser skalfönstret.

I VSCode kan du få stöd genom så kallad `syntax highlighting`. Det betyder att VSCode känner igen t.ex. alla reserverade ord i Java och automatiskt sätter olika färger på olika delar av programkoden. På det viset blir det mycket lättare att läsa sina Javaprogram. Nu är det bara att skriva in Javakod i VSCode. När du vill prova att kompilera och köra ditt program så sparar du först koden i VSCode och skriver sen följande i skalfönstret:

```
> javac NAMNET_PÅ_DIN_JAVA_KLASS.java
> java NAMNET_PÅ_DIN_JAVA_KLASS
```

Om något inte är korrekt i din kod så kommer kompilatorn, `javac`, att ge dig felmeddelande. Då kommer heller ingen körbar fil att skapas vilket betyder att du inte kan starta programmet med kommandot `java NAMNET_PÅ_DIN_JAVA_KLASS`. Istället får du läsa felmeddelandet, återgå till VSCode och försöka rätta till felen som kompilatorn rapporterade.

1.2 Räkna växelpengar*

Låt oss nu lösa en lite mer komplicerad programmeringsuppgift. Du ska skriva ett program som räknar ut växel för svensk valuta. Uppgiften är uppdelad i små steg som följande:

1. Skriv ett program som räknar ut växel på en tiokrona i femkronor och enkronor. Användaren anger ett pris mellan en och tio kronor och ditt program svarar med vilken växel hen får tillbaka.
2. Ändra till växel på en femhundralapp i hundralappar, femtiolappar, tjugor, tiokronor, femkronor och enkronor. Användaren får nu ange ett pris mellan en och 500 kronor.
3. Lägg till att användaren får ange både priset och betalningssumman. Utöka programmet så att det hanterar femtioöringar. En programkörning efter steg 3 skulle kunna se så här ut:

```
[john@emil23 java]$ java Pengar
Ange pris (kr): 124.50
Du betalar (kr): 500
Du får tillbaka 3 hundralappar, 1 femtiolappar, 1 tjugor, 0 tior, 1
femmor, 0 enkronor och 1 femtioöringar.
[john@emil23 java]$
```

4. Studera programmet. Flera steg är troligen väldigt snarlika. Samma operationer utförs med olika data. □Ändra din lösning så att du använder arrayer för att spara data i varje steg och en loop för att gå igenom alla steg. När du är klara med programmet tänk sedan igenom och testa följande:
 - Vad händer om man betalar för lite?
 - Vad händer om man matar in ett negativt pris eller en negativ betalningssumma?
 - Vad händer om man matar in bokstäver istället för siffror, t.ex. priset "tio"?
 - Får man någon växel om man matar in priset 5.60 kr och betalar 6 kr?
 - Om man ska få 60 kr i växel vill man kanske ha det i form av tre tjugor. Får man det via ert program? Om inte, hur skulle det lösas?

Nu har ni genomfört ert första test av ett program! Det är alltid bra att utsätta sina lösningar för olika test och specialfall. Man upptäcker alltid saker man glömt eller inte förutsett. Redovisa programmet för din handledare.