

# Hur man börjar programmera i Java på Ubuntu

Av:Lukas Klingsbo Itnv-06

Det utvecklingssystemet vi ska använda heter Java 2 SDK. Det kan man ladda hem på <http://java.sun.se/>. När du har laddat hem och installerat Java 2 SDK så har du fått en kompilator och en interpretator installerade.

En Java kompilator är ett program som gör om källkoden du har skrivit till Java bytekod. Java bytekoden läses sedan av interpretatorn när du exekverar programmet och så översätter sedan interpretatorn Java bytekoden till maskinkod som din plattform kan läsa.

Du kan skriva din kod i en valfri textredigerare. Jag ska använda gEdit som finns installerat i Ubuntu som standard. gEdit har någonting som kallas syntax highlighting, då blir olika ord i koden färg så det blir lättare att läsa den. Du öppnar gEdit genom att antingen trycka alt+F2 och skriva gedit eller klicka Program > Tillbehör > Textredigerare. En annan bra textredigerare för programmerare som kör Linux heter MonoDevelop (den kan du hämta genom att skriva sudo apt-get install monodevelop när du har tryckt alt+F2).

När du har öppnat en textredigerare börjar du med att skriva

```
public class Namn{
```

på första raden.

Efter du har gjort det ska du spara filen, i nästan alla textredigerare kan du trycka ctrl+s för att spara. Du kan också gå in på Arkiv och sen trycka spara. Gör en mapp på skrivbordet som heter MyJava och spara din fil som **Namn.java**. Vid Namn kan du ha vilket namn du vill bara det är samma namn som står efter public class. När du sparar din kod ska du alltid spara den med filändelsen **.java**.

Efter du har sparat din fil i MyJava mappen på skrivbordet kan du fortsätta skriva koden. Efter du har skrivit public class Namn{ ska du skriva:

```
public static void main(String[] args){
```

på nästa rad. Vid { börjar programmet.

På nästa rad ska du skriva:

```
System.out.print("Mitt första program");
```

System.out.print säger till ditt system att skriva ut det inom citationstecknena på skärmen.

Nu ska du skriva en } på nästa rad. och sen hoppa ner en rad till och skriv en till }.

Nu ska din kod se ut så här:

```
public class Namn{
```

```
public static void main(String[] args){
```

```
System.out.print("Mitt första program");
```

```
}
```

```
}
```

För att göra det enklare att läsa din kod ska vi använda tab-knappen och göra lika långa indrag vid den koden som hör ihop.

Gör så att ditt program ser ut så här:

```
public class Namn{
    public static void main(String[] args){
        System.out.print("Mitt första program");
    }
}
```

Kompilatorn bryr sig inte om när du tabbar så som vi just gjorde.

```
public class Namn{
    public static void main(String[] args){
        System.out.print("Mitt första program");
    }
}
```

Nu kan man lättare se vilka { } som hör ihop med vilka. De som jag har färgat röda hör ihop med varandra och de gröna hör också ihop. Om du följer en av de två första raderna nedåt så ser du vilken } raden hör ihop med.

Nu ska vi kompilera din kod. Öppna en kommandotolk, det kan du göra genom att trycka alt+F2 och skriva gnome-terminal eller gå in på Program > Tillbehör > Terminal. När du har öppnat terminalen kan du börja med att skriva **pwd** (print working directory), då får du se var någonstans i Systemet du befinner dig. Vi ska ta oss till din mapp på skrivbordet som heter MyJava. Vanligvis så startas terminalen upp i din hemkatalog och för att se vilka filer som finns i den så skriver du **ls** (list). Nu ser du en lista med mappar och filer som finns i din hemkatalog. Alla mappar som visas är i blå färg, vi ska gå in i mappen Desktop som visar filerna du har på skrivbordet. För att gå in i mappen skriver du **cd Desktop**, cd står för change directory. När du är på skrivbordet står det ~/Desktop\$ i början på raden i terminalen. Nu ska vi gå in i din mapp MyJava som du gjorde på skrivbordet, för att göra det skriver du **cd MyJava**. Nu när du är inne i mappen med din Java fil kan du kolla för säkerhetsskull att den är där och då gör du samma sak som du gjorde förrut när du skrev **ls**. När du har kollat att din källkodsfil är där så ska vi kompilera den, dvs göra om den till kod som java interpretatorn (Java Virtual Machine) kan läsa och göra om till maskinkod. För att kompilera din fil skriver du **javac Namn.java**. Ifall du får kompileringsfel har du antagligen skrivit någonting fel i koden t.ex. om du får det här felet:

```
Namn.java:3: ';' expected
        System.out.print("Mitt första program")
                                ^
```

## 1 error

så har du glömt att sätta ditt ett semikolon i slutet av raden. Felmeddelandet man får när man har kompileringsfel är oftast ganska lätt att förstå men de kan vara krångliga att förstå också ibland. När du ska rätta dina kompileringsfel, börja alltid vid det första felmeddelandet för de andra felmeddelandena kan bli för det första felet. Om du får ett logiskt fel kan du t.ex. ha försökt att dividera med 0. Om du får ett exekveringsfel är det där emot jobbigare att lösa, då blir det fel när du kör ditt program och du har antagligen en bugg i din kod.

När du har kompilerat ditt program med **javac Namn.java** och inte får några felmeddelanden så kan du köra programmet och lägg märke till att när ditt program har kompilerats så blir det en **Namn.class** fil i mappen där du hade källkodsfilen. För att köra programmet skriver du **java Namn** i terminalen när du fortfarande är inne i din MyJava mapp. När du skriver **java Namn** körs din kod i interpretatorn (Java Virtual Machine) och översätts till maskinkod av den.

Ha så kul med programmeringen.