

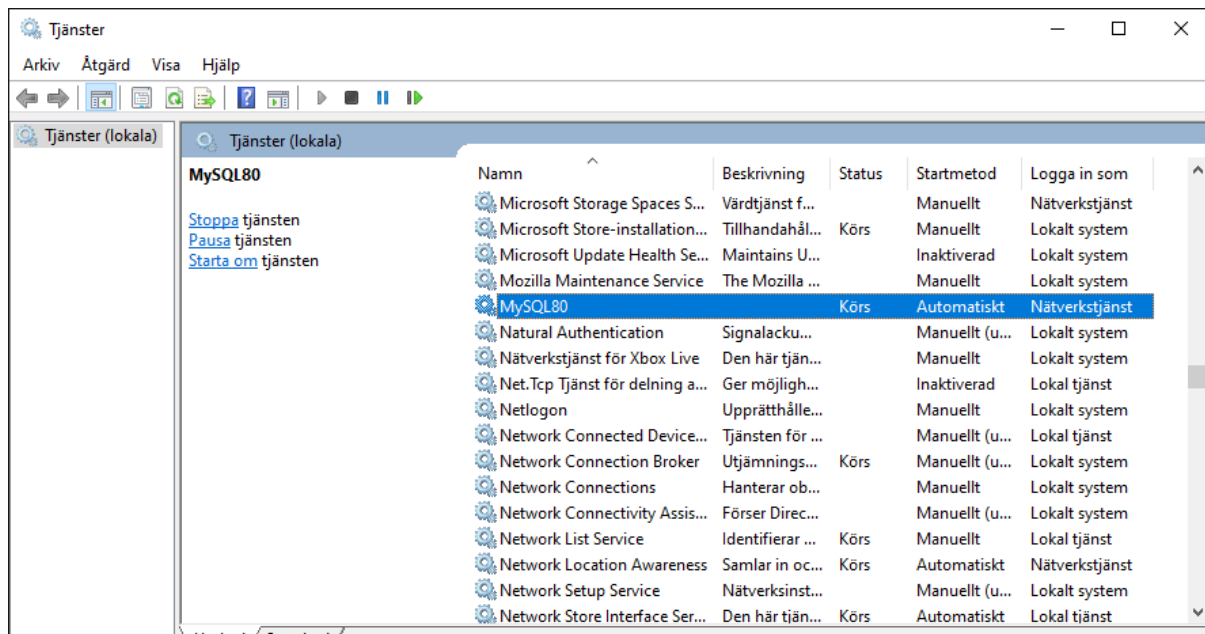
Windows instruktioner för att komma igång med MySQL i Netbeans

Program och filer du behöver ha nerladdat på datorn

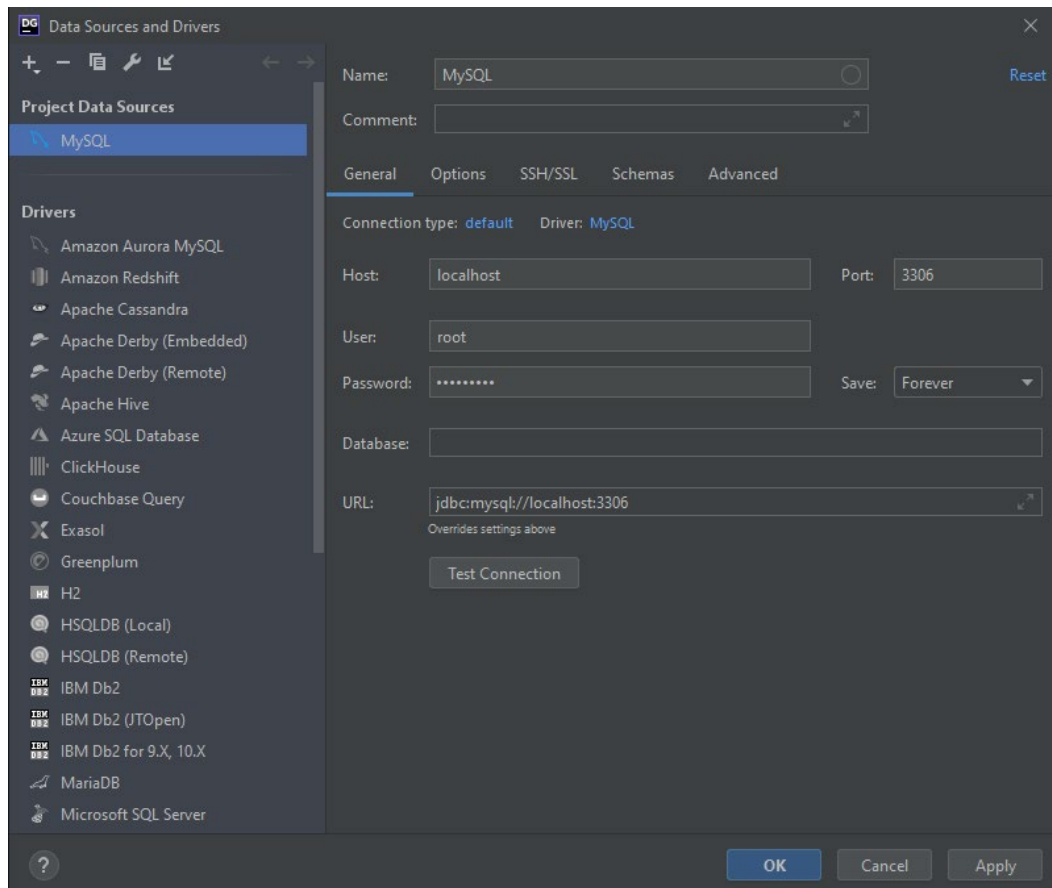
- **SQL Community Server:** <https://dev.mysql.com/downloads/windows/installer/8.0.html>
- **Datagrip:** <https://www.jetbrains.com/datagrip/download/#section=windows>
- **Netbeans IDE:** <https://netbeans.apache.org/download/nb122/nb122.html>
- **Databasfiler.zip** från Blackboard

STEG 1: Skapa databasen "mibddb" i Datagrip

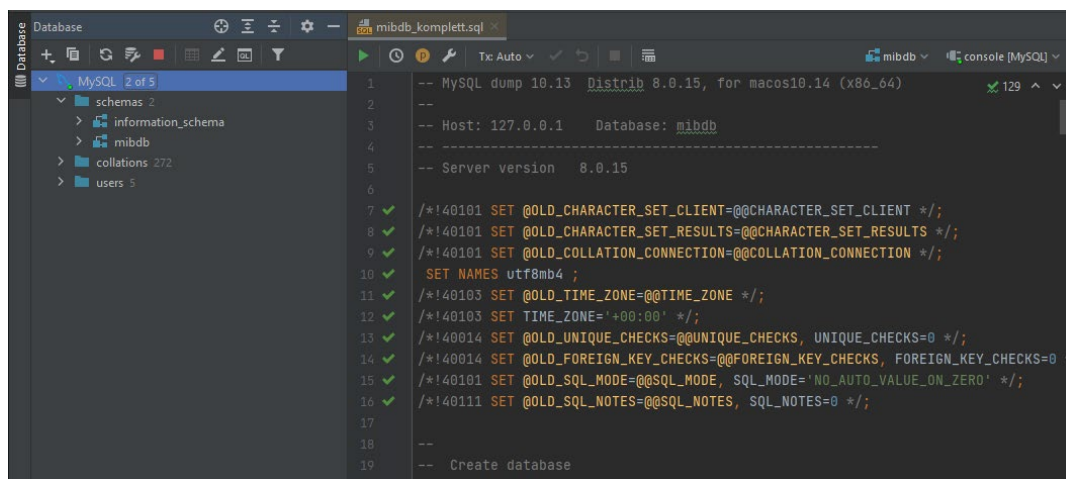
Se till att MySQL servern är igång på din dator. Du kan kolla inställningarna genom att gå in på **Kontrollpanelen** -> **Administrationsverktyg** -> **Tjänster** och sen leta upp **MySQL** i listan. Du kan stoppa eller starta MySQL därifrån.



1. Skapa en ny **Data Source** i Datagrip och välj **MySQL**. Ge den valfritt namn, det kommer inte användas mer än att det är det namn det listas med i Datagrip efteråt. **Host** ska vara "localhost" och **port** ska vara "3306" (porten kan vara något annat också, men ändra inte om inte du måste). **User** är namnet som du antingen tilldelades automatiskt eller fick välja när du installerade MySQL. Valde du inte namn på din MySQL user så är det "root" som gäller. **Password** är det du valde när du installerade MySQL. Lämna **Database** tomt. Klicka på **Test Connection** och lyckas det kan du klicka på **OK**.



- Öppna ett nytt query-fönster i Datagrip, lägg in filen **mibdb_komplett.sql** som finns i mappen **Databasfiler** du laddat ner från blackboard och kör frågan via din **Data Source**. Efteråt borde databasen **mibdb** dyka upp under mappen **schemas** i din nyskapade Data Source (se bild nedanför). Efter detta kan du stänga ner Datagrip.

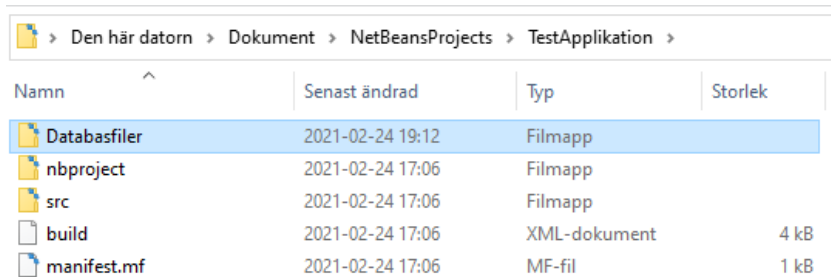


Ps. Om du tittar på skriptet för **mibdb_komplett.sql** ser du att det ganska högst upp skapas en **user** för mibdb-databasen. Dessa uppgifter är det som vi kommer använda sen för att koppla upp till databasen via Netbeans:

```
CREATE USER mibdba@localhost IDENTIFIED BY 'mibkey';
```

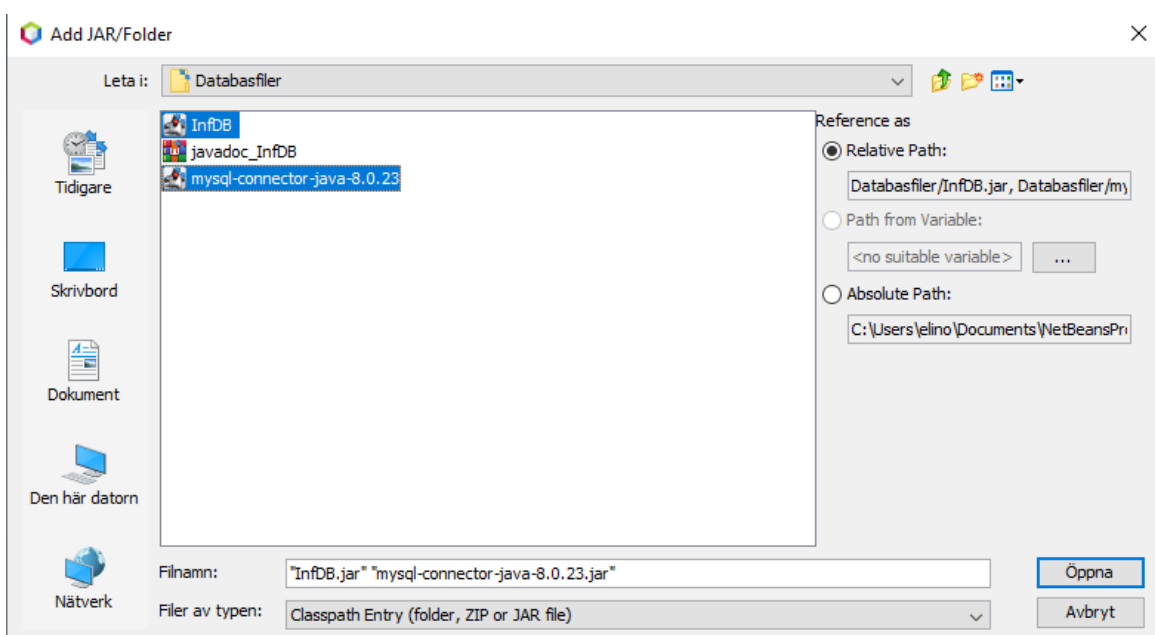
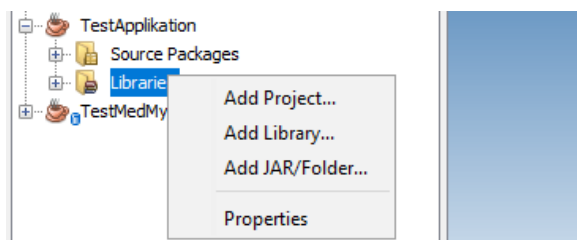
STEG 2: Importera JAR-filerna som behövs för att kunna börja koda mot databasen

- Lägg in mappen **Databasfiler** som du laddat ner från Blackboard i Netbeansprojektmappen som du har på din dator. Detta kommer underlätta delning (ex. via Github) av projektet mellan gruppmedlemmar då alla filer som kommer behöva användas i projektet kommer kunna ha relativa sökvägar. Alla gruppmedlemmar behöver således inte ha alla filer själva på sin dator någonstans med absoluta sökvägar (ex. C:\user\anna...) som andra datorer naturligt inte kommer åt, utan de finns istället tillgängliga i projektmappen.

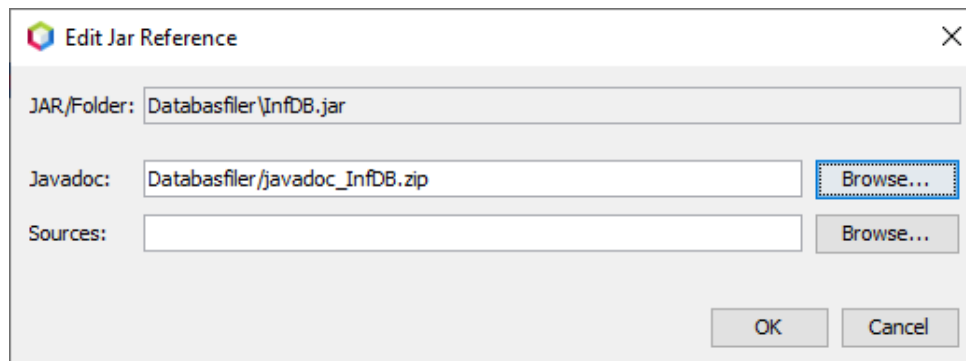
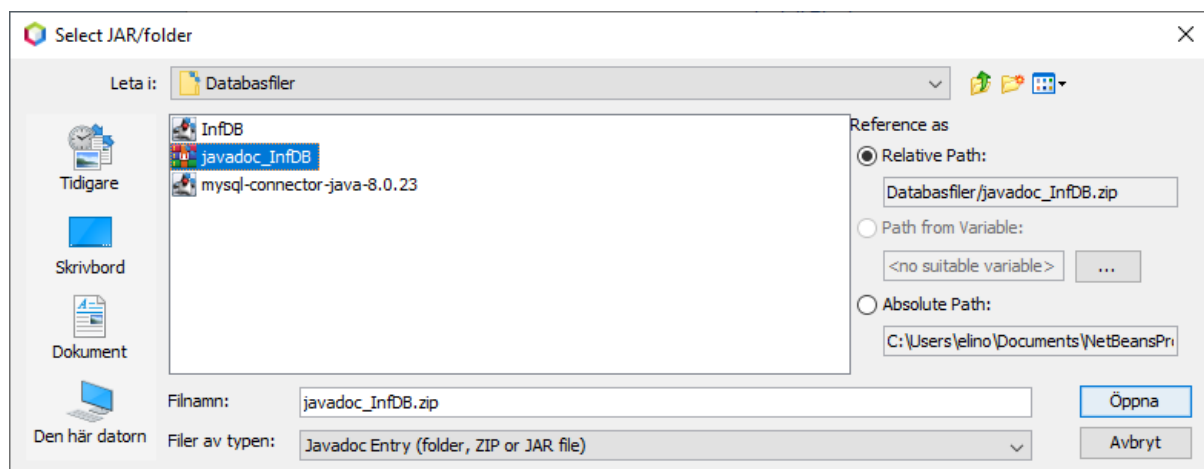
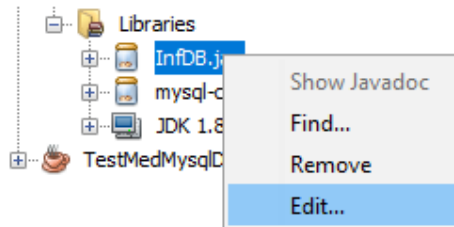


» Den här datorn » Dokument » NetBeansProjects » TestApplikation »			
Namn	Senast ändrad	Typ	Storlek
Databasfiler	2021-02-24 19:12	Filmapp	
nbproject	2021-02-24 17:06	Filmapp	
src	2021-02-24 17:06	Filmapp	
build	2021-02-24 17:06	XML-dokument	4 kB
manifest.mf	2021-02-24 17:06	MF-fil	1 kB

- Öppna Netbeans och högerklicka på mappen **Libraries** som finns i den vertikala projektmenyn för ditt projekt, och välj **ADD JAR/Folder**. Markera **InfDB.jar** och **mysql-connector-java-8.0.23.jar** som då rimligtvis ligger din projektmapp under mappen **Databasfiler**, och kom ihåg att välja **Relative Path**. Klicka på **Öppna**.

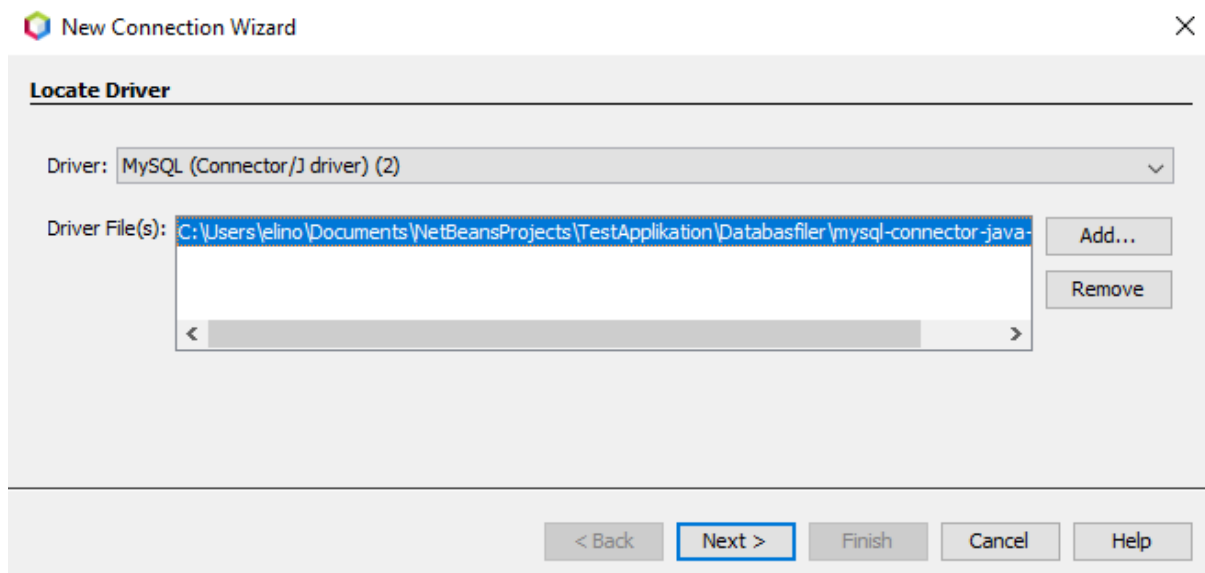
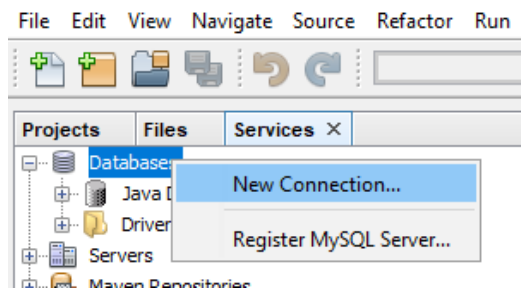


5. Tänkte även rekommendera att importera dokumentationen för InfDB-klassen. Hoppa över till nästa steg ifall du känner att du inte behöver ha detta. Högerklicka på **InfDB.jar** i sidomenyn och klicka på **Edit**. Klicka på **Browse** och leta upp **javadoc_InfDB.zip** som ligger i mappen **Databasfiler** i projektmappen och kom ihåg att välja **Relative Path**. Klicka sen på **Öppna** och sen **OK** och dokumentationen är inlagd för ditt projekt.



STEG 3: Skapa databasuppkoppling mellan ditt projekt och MySQL-databasen

6. Klicka på **Services** uppe i projektmenyn, högerklicka på **Databases** och välj **New Connection**. Först behöver vi lägga till en MySQL driver, så klicka på **New Driver** i dropdown-menyn. Klicka sen på **Add** och välj **mysql-connector-java-8.0.23.jar** som du hittar i mappen **Databasfiler** och klicka på **OK**. Klicka sen på **Next**.



7. Nu är det dags att fylla i informationen som behövs för att Netbeans ska kunna koppla upp sig mot databasen vi skapade i Datagrip. **Database** är namnet på databasen som du kan se i sidomenyn på Datagrip, alltså *"mibdb"*. **User Name** och **Password** är den **user** som skriptet skapade, alltså *"mibdba"* och *"mibkey"*. Se även till att **JDBC URL** är:

`jdbc:mysql://localhost:3306/mibdb`

Klicka på **Test Connection** och se ifall uppkopplingen lyckades. Gjorde den det så är det bara **Next** tills du kan skriva in **Input connection name**. Där får du möjlighet att ge ett namn till databasen (endast för att särskilja olika databaser i Netbeans sidomeny ifall du har flera, ingen mer praktisk funktionalitet), och klicka sen på **Finish**.

New Connection Wizard

Customize Connection

Driver Name: MySQL (Connector/J driver) on MySQL (Connector/J driver) (1)

Host: localhost Port: 3306

Database: mibdb

User Name: mibdba

Password: ☒ Remember password

Connection Properties Test Connection

JDBC URL: jdbc:mysql://localhost:3306/mibdb

< Back Next > Finish Cancel Help

New Connection Wizard

Choose name for connection

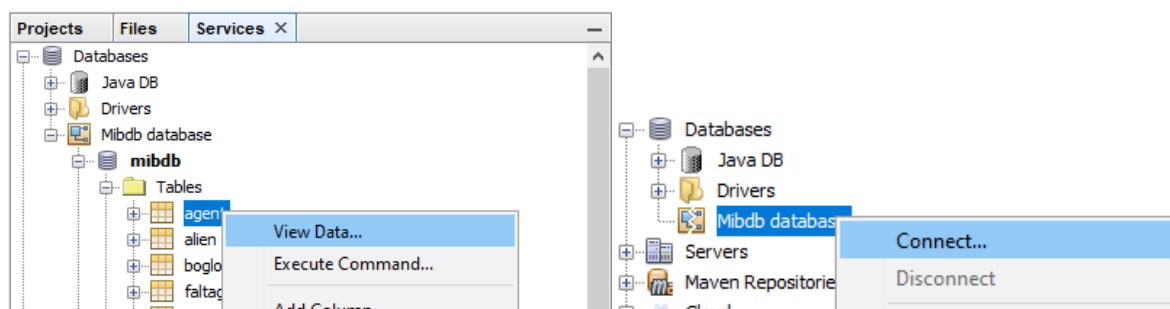
Override the default name for the connection. The name should be descriptive about the connection you are creating.

Input connection name:

Mibdb database

< Back Next > Finish Cancel Help

Efter det kommer databasen dyka upp under **Databases**, och du borde kunna gå in och titta på tabellerna och köra SQL-frågor mot den. Har databasen en trasig ikon och inte har något innehåll när man försöker öppna upp den (kanske händer under projektets gång) är det bara att högerklicka och försöka återkoppla.



STEG 4: Programmera startklassen för att börja arbeta mot databasen

8. Nu har vi en fungerande databas redo att användas i Netbeansprojektet, men för att faktiskt kunna nå den med kod behöver vi skapa en instans av InfDB-klassen. Det är den som kopplar upp sig till databasen och har metoder som du kan använda för att hämta data från databasen. Vi har i **steg 2** gått igenom importen av själva filerna in i projektet, och nu kan du lätt importera referenser till InfDB-klasserna högst upp i startklassen, så att det går att komma åt klassernas metoder du behöver för att skapa en instans av klassen, och senare för att kunna ställa SQL-frågor.

InfDB-klassen har en konstruktor som tar emot 4 parametrar:

- **Databasnamn** ("mibdb")
- **Port** till din MySQL server ("3306")
- **Användarnamn** för databasen ("mibdba")
- **Lösenord** för databasen ("mibkey")

När du har skapat en instans av InfDB (se exempel nedanför) kan du prova köra programmet och se ifall det funkar.

```
package testapplikation;

import java.util.logging.Level;
import java.util.logging.Logger;
import oru.inf.InfDB;
import oru.inf.InfException;

public class TestApplikation {

    private static InfDB idb;

    public static void main(String[] args) throws InfException {

        try {
            idb = new InfDB("mibdb", "3306", "mibdba", "mibkey");
        } catch (InfException ex) {
            Logger.getLogger(TestApplikation.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

Om allt gått rätt till så borde det bara vara att börja programmera!