

Database Lab#9 Connecting to postgresql database from java programs

A client program written in java language can connect to a database server by using JDBC (java database connectivity) library. Classes of this library manage and organize the data transfers between the database server and clients.

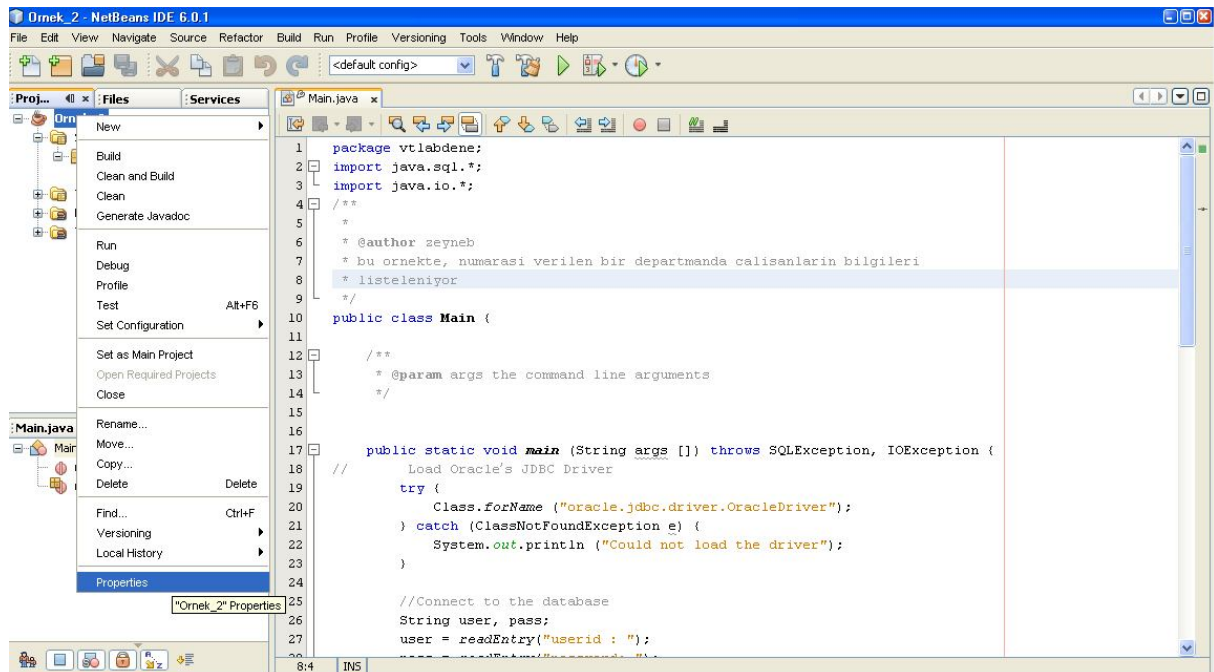
First, a connection between the client and the server should be set up. Then, client requests information from the database server. Server answers to the requests of the client. Several clients can connect to the server at the same time.

In this lab, we examine the steps of the connection process from netbeans projects to the postgresql database server. First, we should get a proper JDBC driver from:

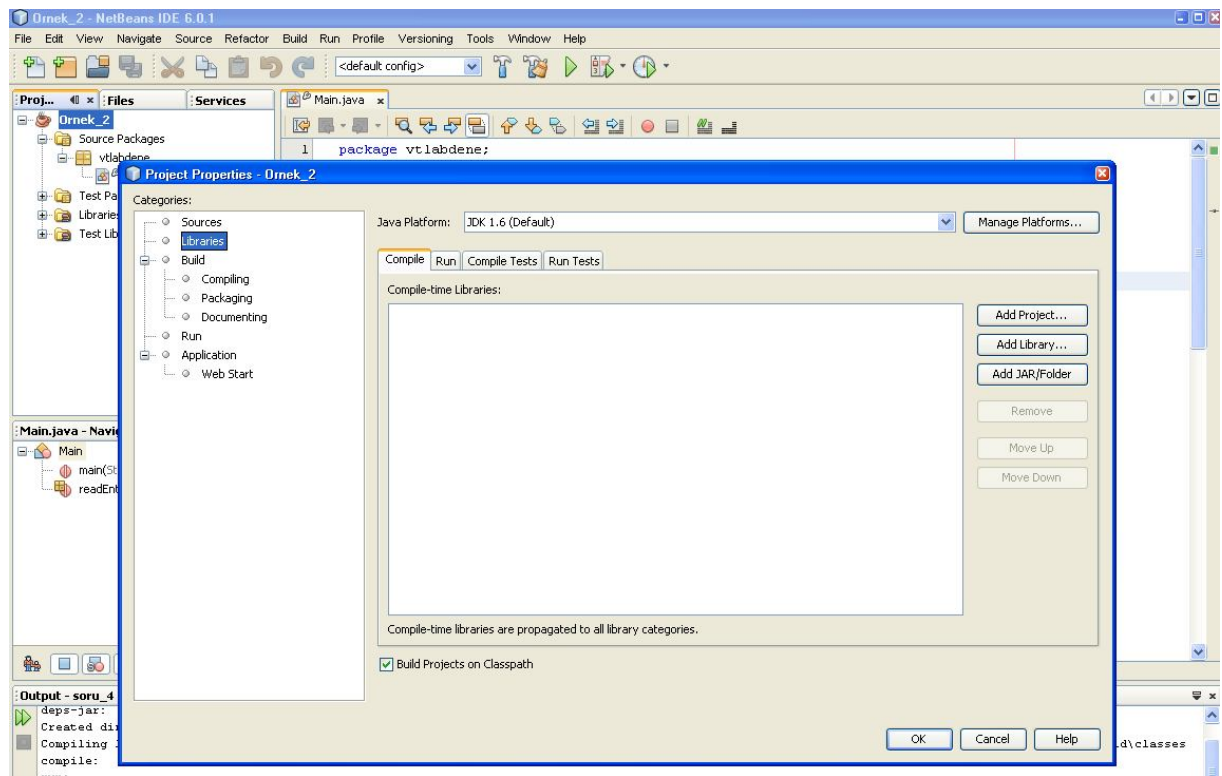
<http://jdbc.postgresql.org/download.html>

After downloading this library, we should indicate its path for our client programs. I renamed the downloaded jar file's name as "jdbc". You can indicate the path as following:

1.Right click the project and click properties:

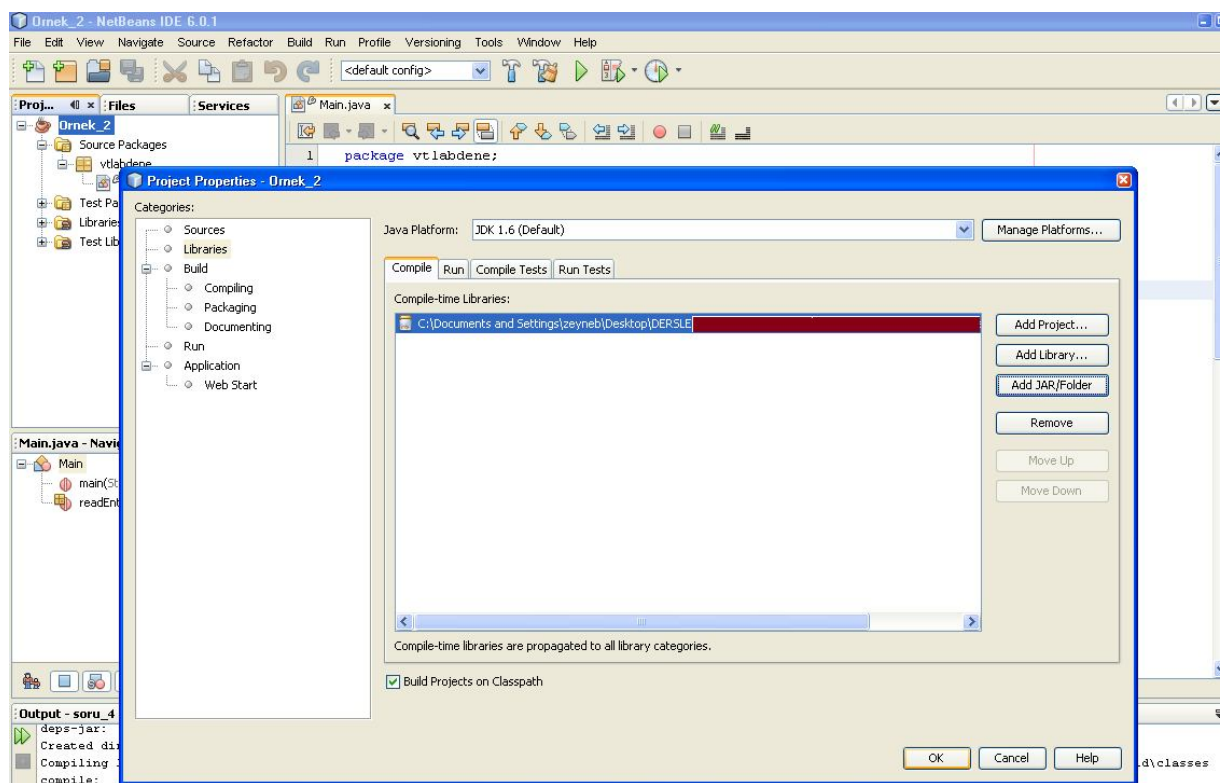


2. Click the "libraries" on the new box:



3. Click “Add JAR/Folder” (third button at the right side)

Add jdbc.jar and write “import java.sql.*;” in the client program. Now you can connect to the postgresql database and write the sql queries.



Examples:

EXAMPLE 1: Find the surname and salary of the employee, whose SSN is given:

20-th and 21-st lines in the main function of Ornek1:

Connection conn =

```
DriverManager.getConnection("jdbc:postgresql://localhost:5432/postgres", user, pass);
```

“jdbc:postgresql”: Defines the protocol, which is used by the driver;
localhost:5432: url address and port number of the server;
postgres: Name of the connected database
user: Username of the database
pass: Password of the user

23-rd and 24-th lines in the main function of Ornek1:

```
String query =  
    "select LNAME,SALARY from EMPLOYEE where SSN = ?";  
PreparedStatement p = conn.prepareStatement (query);
```

The query is prepared to a statement. Question marks (?) get values from the variables after the definition of the queries:

```
String ssn = readEntry("Enter a Social Security Number: ");  
p.clearParameters();  
p.setString(1,ssn); // first question mark gets the value of the variable “ssn”
```

After execution of the query in the statement, conclusion is taken in a resultset object:
ResultSet r = p.executeQuery();

If there are records in the resultset, value of the “r.next()” is true, otherwise its value is false:

```
if (r.next ())
```

Types of the records taken from resultset can be changed into any type we want:

```
String lname = r.getString(1); // get the first column’s value as string  
double salary = r.getDouble(2); // get the second column’s value as double
```

Close the prepared statement of the query (p):

```
p.close();
```

Disconnect from the server:

```
conn.close();
```

EXAMPLE 2: List the name and salary information of the employees, whose department number is given:

Create a “statement” object in this example (see 37-th line of the main code in Ornek2)

```
Statement s = conn.createStatement();
```

The query is executed via this statement object. While there are records in the resultset, information are written on the screen: **(lines between 40 and 44 in main code of Ornek2)**

Close statement object and the connection at the end of the program.

Difference between Statement and preparedStatement objects:

In the PreparedStatement object, query string is sent to the database server before the execution. Syntax check of the query is done by the server. After the question marks (?) in the query get value, query is executed by the server.

However in the statement object, query is already prepared before sending to the server. Syntax control and execution of the query are accomplished at the same time.

Note: Statement objects have two different execution methods: `executeQuery` and `executeUpdate`. The first one takes a sql query as input and execute it. `executeUpdate` takes the commands “create table...”, “insert..”, “update..”, “delete..” (ddl and dml commands) as input and execute those commands.

Note: If a query in a `PreparedStatement` object has more than one question mark (?), we should set the value of question marks by denoting the rank of the question marks in the query and using the appropriate “set...” function. For instance a query has two question marks. Its type is integer, so we should write `p.setInt(2, parameter_2)` to set its value.

Other examples:

Write the following queries in java programs and connect to the PostgreSQL database server to see results.

1. Find the name and salary of the female employees, who are working for the project with the name of “DatabaseSystems”.
2. Find the name and salary of the employees, who are working at the departments that have offices in “Chicago”.
3. Find number of the employees of each department and list the result according to the department name in alphabetical order.
4. Find how many workers are studying for the project “ProductX” and average salary of them.

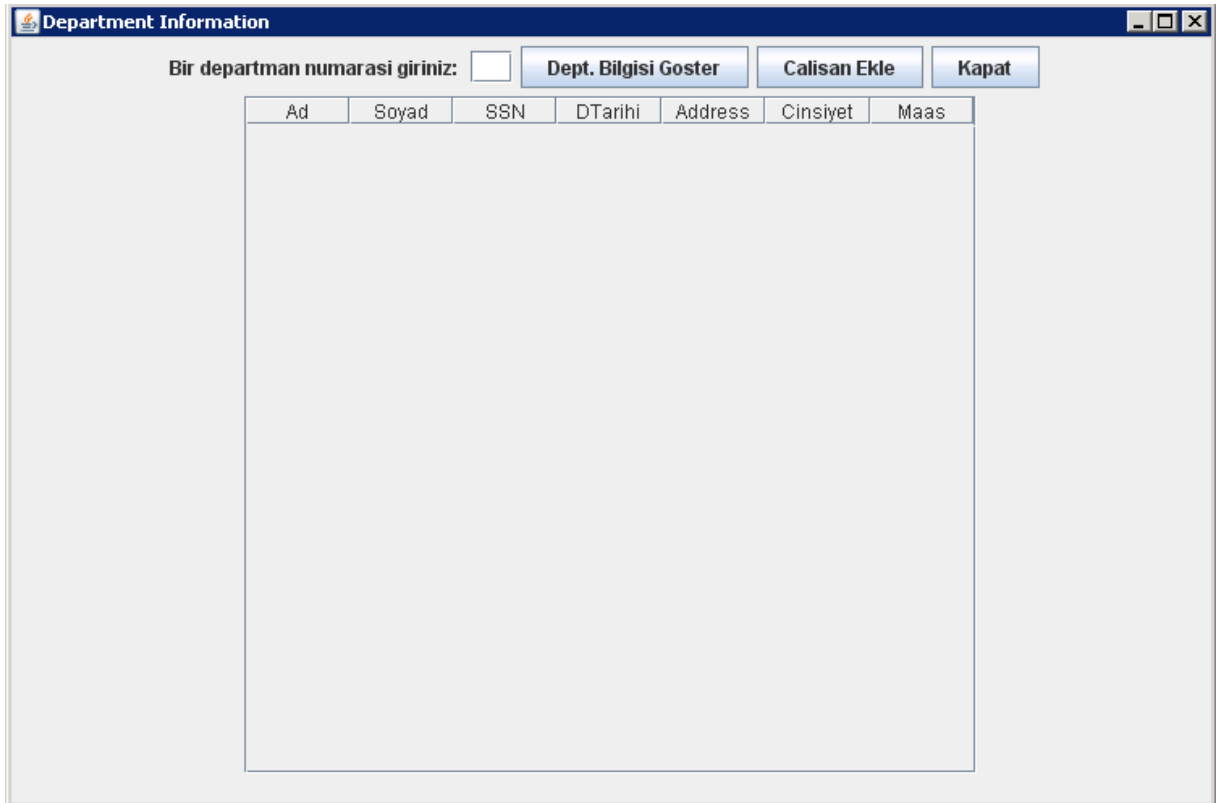
If we write the queries by not using java programs, queries will be as:

1. `select fname , lname from employee e, works_on w, project p where w.essn=e.ssn and w.pno=p.pnumber and e.sex= 'F' and pname='DatabaseSystems'`
2. `select fname, lname, salary from employee e, department d, dept_locations dl where e.dno=d.dnumber and d.dnumber = dl.dnumber and dlocation = 'Chicago'`
3. `select dname, count(*) from department d, employee e where d.dnumber= e.dno group by dname order by dname desc`
4. `select count(*), avg(salary) from employee e, project p, works_on w where e.ssn=w.essn and p.pnumber=w.pno and pname = 'ProductX'`

Constructing simple interfaces by using frames in java platform:

The class including the main function connects to the database server. Main function sends a connection object as parameter to a JFrame object.

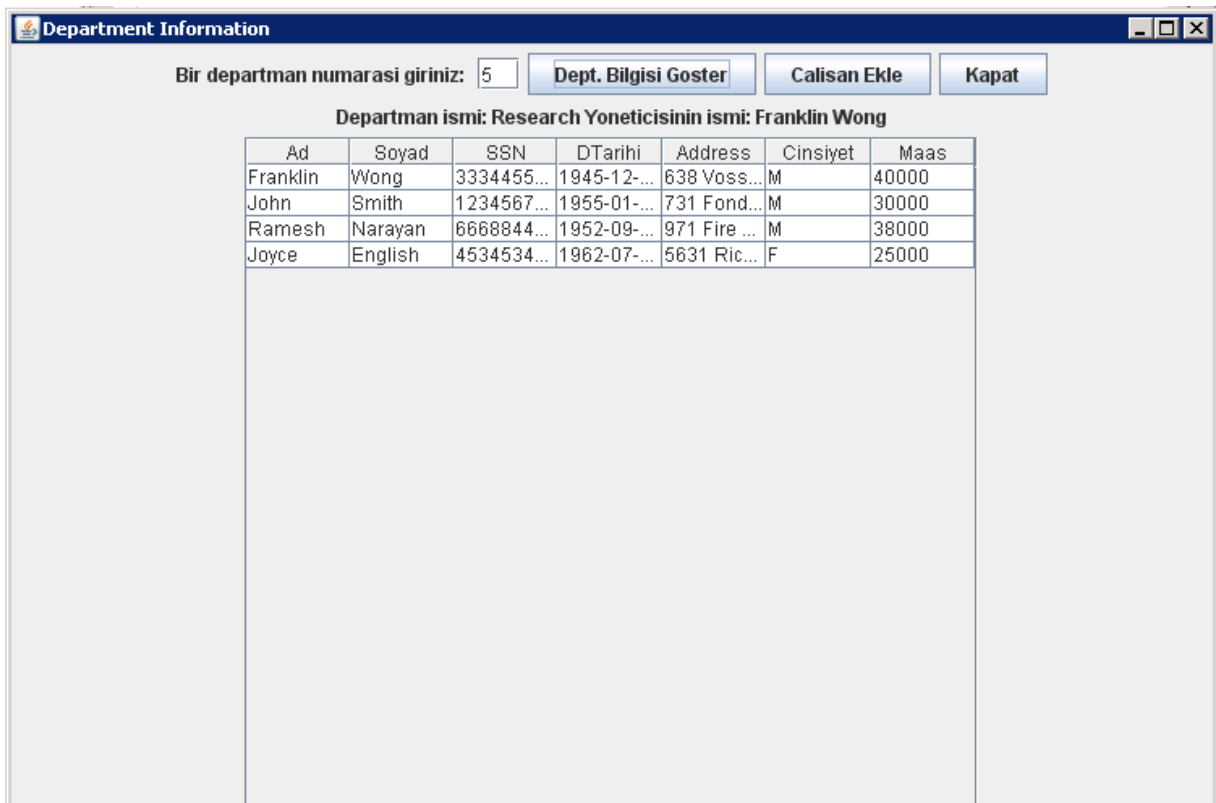
The JFrame object defines the size of the interface and creates a JPanel object. Connection parameter is given to JPanel object too.



Jpanel involves the components of the interface: Labels, textbox, buttons, Jtable (the data grid which shows the information of the employees).

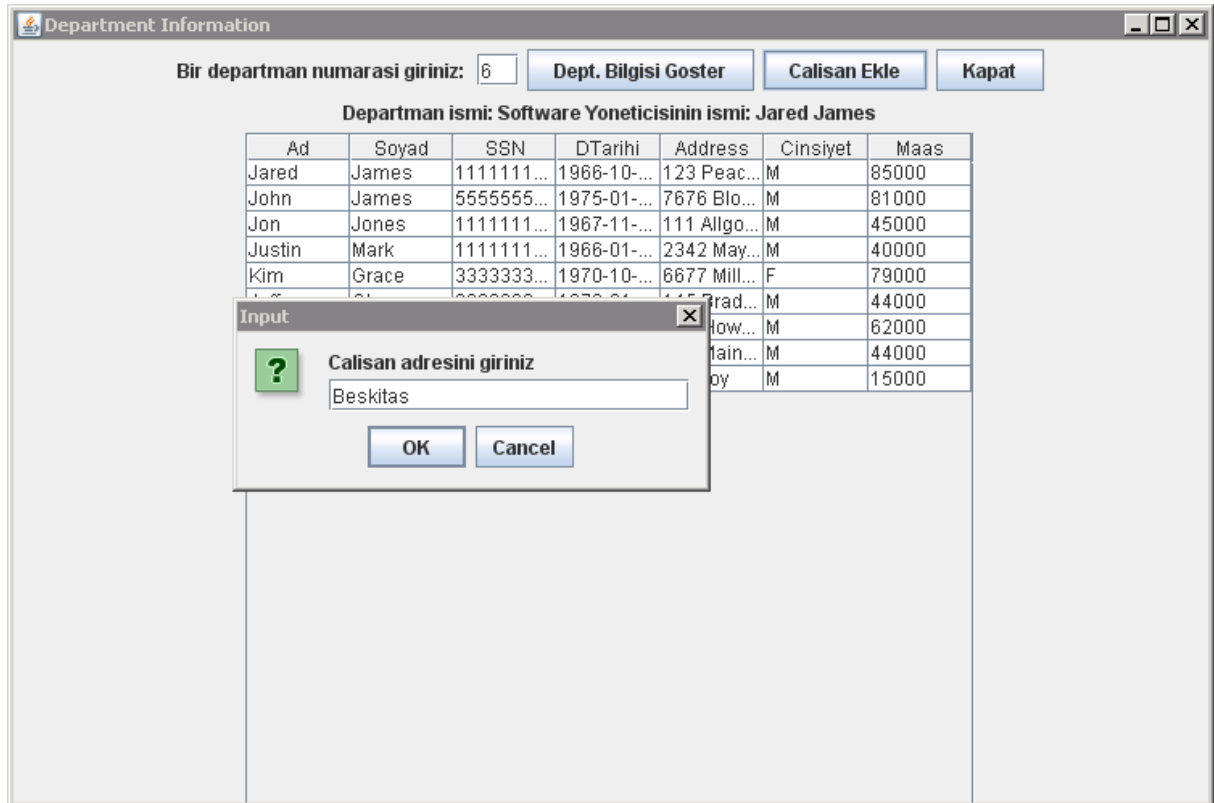
To see the department name and the manager name of a given department number, enter the department number to the text box. The Label before the textbox with the caption “Bir departman numarası giriniz” means “Enter a department number”. Push the button “Dept. Bilgisi goster” which means “Show the information of the department”.

Enter the value of the department number as 5, then the interface will be:

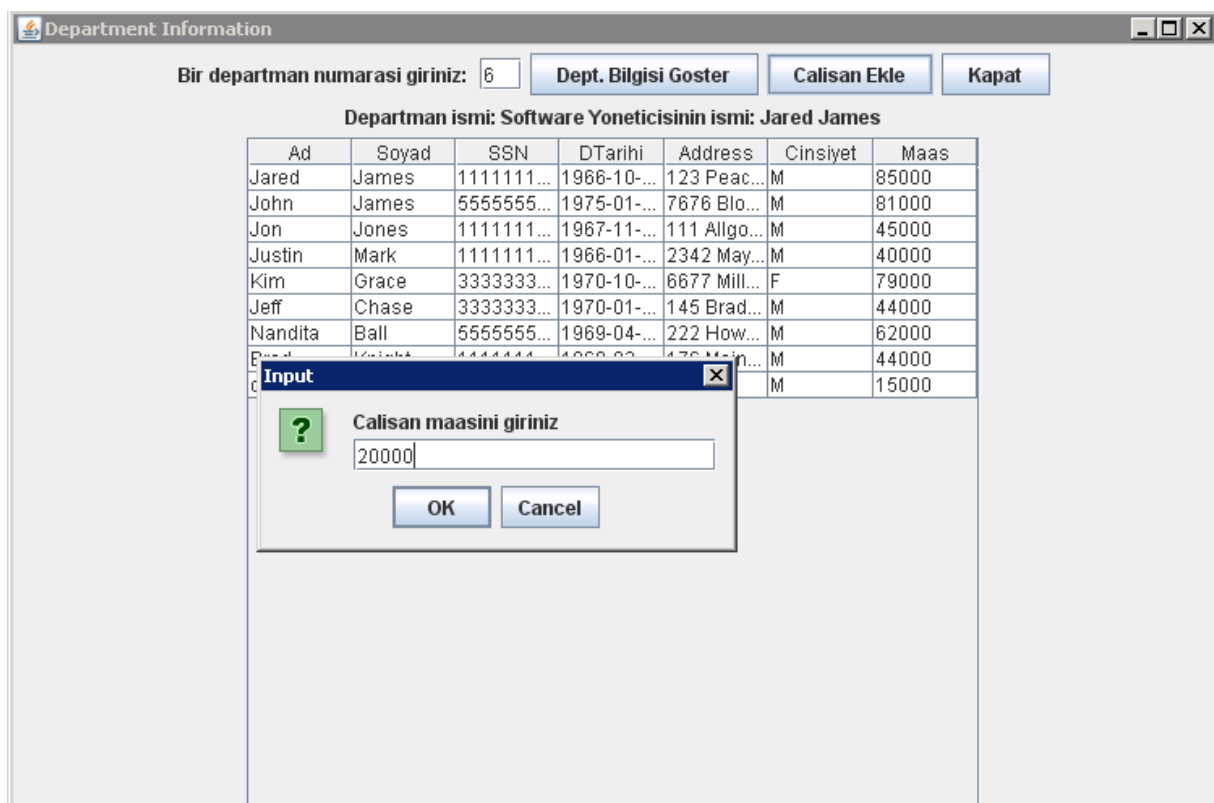


The label “Department ismi” means “Name of the dept.,” the label “Yoneticisinin ismi” means “Name of the department’s manager”. The data grid shows the information of the employees who are working at the 5-th department. (meanings of the captions: Ad=Name, Soyad=Lastname, DTarihi=Bdate, Cinsiyet=Sex, Maas=Salary)

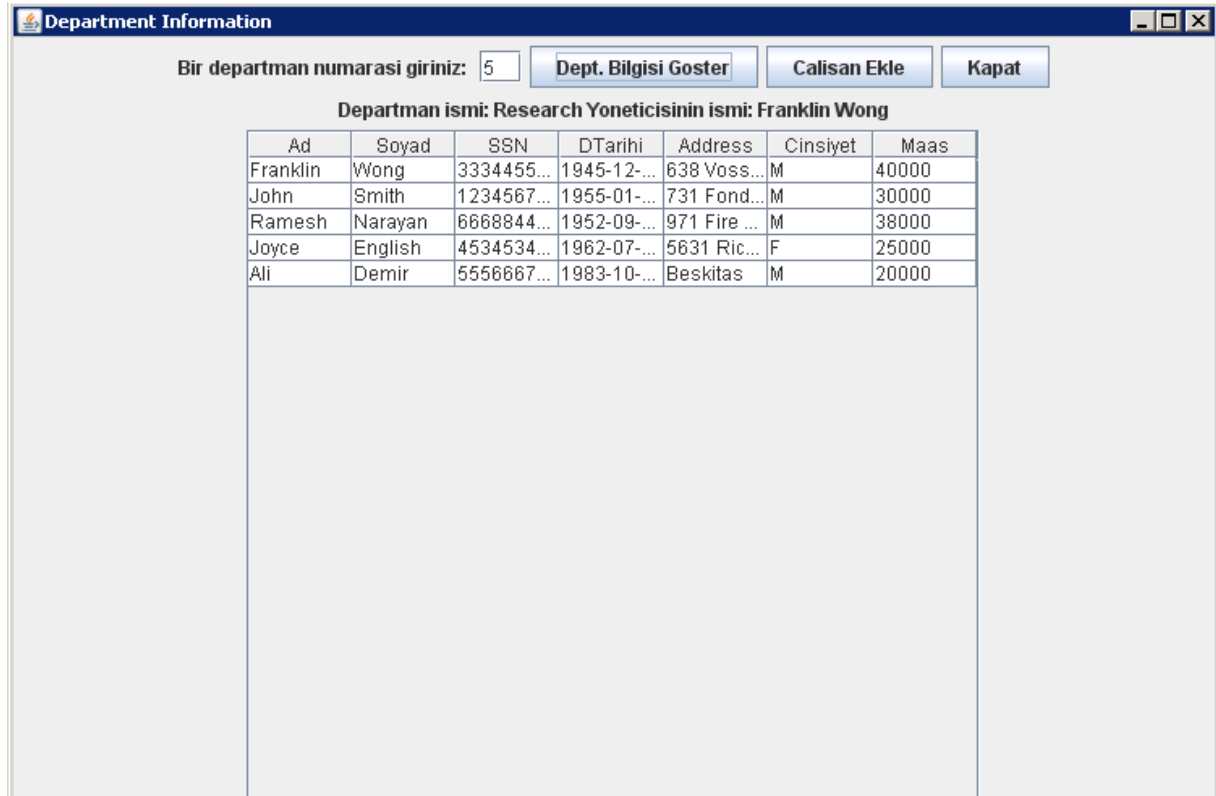
To insert a new employee push the second button with the name “Calisan Ekle” (means “insert employee”). A few inputMsgBoxes will be seen and they ask for the information of new employee from the user. (respectively informations of *fname*, *lname*, *ssn*, *bdate*, *address*, *sex*, *salary*, *dno*). For example the inputbox asking for the address of employee:



The inputbox asking for salary information of the new employee:



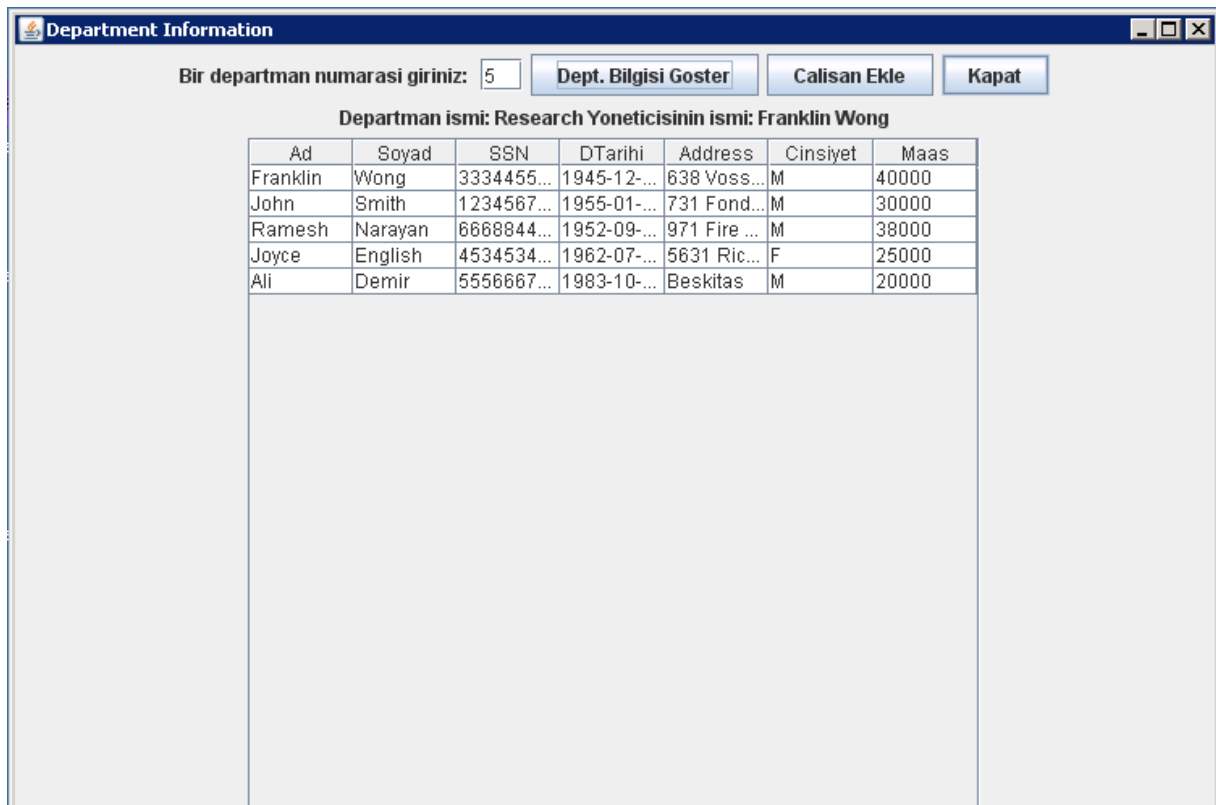
For instance dno is entered as 5 for the new employee. To see whether the new employee is added to the table, enter 5 to the textbox, which asks the dnumber information and push the first button (“dept bilgisi goster” which means “show the information of the dept”):



Ad	Soyad	SSN	DTarihi	Address	Cinsiyet	Maas
Franklin	Wong	3334455...	1945-12-...	638 Voss...	M	40000
John	Smith	1234567...	1955-01-...	731 Fond...	M	30000
Ramesh	Narayan	6668844...	1952-09-...	971 Fire ...	M	38000
Joyce	English	4534534...	1962-07-...	5631 Ric...	F	25000
Ali	Demir	5556667...	1983-10-...	Beskitas	M	20000

The new employee with the name of “Ali Demir” is listed at the bottom of the employee list of the 5-th department.

The button “kapat” means “close”. It first disconnects from the database server and then closes the interface:



Ad	Soyad	SSN	DTarihi	Address	Cinsiyet	Maas
Franklin	Wong	3334455...	1945-12-...	638 Voss...	M	40000
John	Smith	1234567...	1955-01-...	731 Fond...	M	30000
Ramesh	Narayan	6668844...	1952-09-...	971 Fire ...	M	38000
Joyce	English	4534534...	1962-07-...	5631 Ric...	F	25000
Ali	Demir	5556667...	1983-10-...	Beskitas	M	20000