**Лабораторная работа №10**

**Система управления базами данных (СУБД)** – совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования базы данных (БД) многими пользователями.

Для работы с СУБД в Java была предложена и реализована технология Java DataBase Connectivity (JDBC).

**JDBC**  – это платформенно-независимый промышленный стандарт взаимодействия Java-приложений с различными СУБД, реализованный в виде пакета java.sql, входящего в состав Java SE, и его расширения javax.sql, входящее в Java EE.

JDBC является интерфейсом, основанным на спецификациях SAG CLI (SQL Access Group Call Level Interface – интерфейс уровня вызова группы доступа SQL). JDBC определяет, как приложение открывает соединение, ведет обмен информацией с базой данных, выполняет команды SQL и извлекает результаты. Многие концепции JDBC API заимствованы из других источников, в частности, из Microsoft ODBC.

СУБД MySQL совместима c JDBC и будет рассматриваться далее для создания баз данных. Последняя версия CУБД может быть загружена с сайта http://dev.mysql.com, для ее корректной установки необходимо следовать инструкциям (Приложение).

Алгоритм подключения Java-программы к реляционной СУБД с помощью JDBC выглядит следующим образом.

*1. Импортирование пакета* java.sql.\*, который содержит классы для работы с базами данных.

*2. Загрузка и регистрация драйвера*. Например, для СУБД MySQL:

|  |
| --- |
| Class.forName("com.mysql.jdbc.Driver"); |

После загрузки драйвера в память становится возможным соединение с системой управления базами данных.

*3. Установление соединения с сервером базы данных.* Для получения соединения вызывается метод DriverManager.getConnection(), в качестве параметров которому передаются JDBC URL базы данных, логин пользователя и пароль доступа. Метод возвращает объект Connection.

|  |
| --- |
| cn = DriverManager.getConnection(url, name, password); |

Стандартный синтаксис JDBC URL выглядит следующим образом:

|  |
| --- |
| jdbc:<subprotocol>:<subname> |

где <subprotocol> (подпротокол) – имя драйвера или механизма соединения с БД, <subname> – идентификатор базы данных.

*4. Создание запроса (инструкции).*

При создании объекта для передачи запросов применяется метод createStatement() класса Connection.

|  |
| --- |
| Statement st = cn.createStatement(); |

С помощью объекта Statement осуществляется передача SQL-команд в базу данных.

*5. Выполнение запроса* производится с использованием метода executeQuery(), результат которого помещается в объект ResultSet:

|  |
| --- |
| ResultSet rs = st.executeQuery(query); |

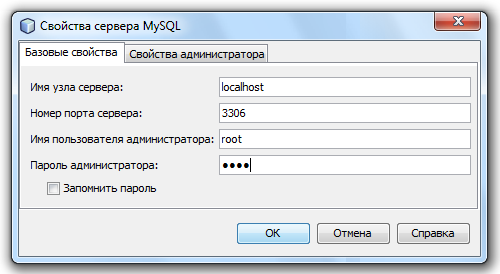
*6. Чтение полученных результатов* производится методами интерфейса ResultSet. Набор данных результата (result set) является таблицей с заголовками колонок и соответствующих значений, возвращенных запросом.

*7. Закрытие соединения* осуществляется с использованием метода close() класса Connection.

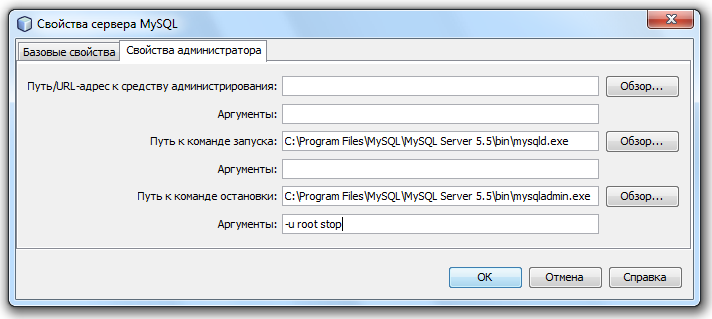
***Лабораторная работа***

**Цель работы:**создание веб-проекта с использованием JSP и СУБД MySQL.

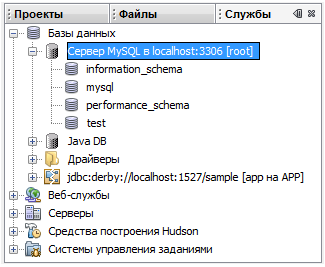
1. Создайте новый проект **DB**.
2. Настройте свойства сервера базы данных MySQL:
   1. перейдите на вкладку **Службы**;
   2. разверните узел **Базы данных**;
   3. нажмите на узел **Сервер MySQL** правой кнопкой мыши (если его нет, необходимо зарегистрировать его при помощи пункта контекстного меню **Зарегистрировать сервер MySQL**);
   4. выберите пункт контекстного меню **Свойства**;
   5. в окне **Свойства сервера MySQL** введите пароль администратора, который был определен при установке сервера базы данных:



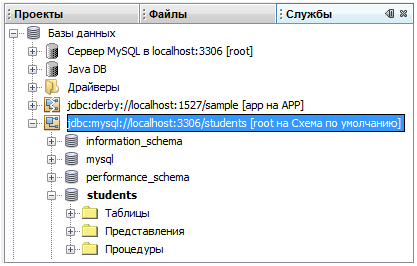
* 1. перейдите на вкладку **Свойства администратора**;
  2. в поле **Путь к команде запуска** с помощью кнопки **Обзор** определите путь к файлу **mysqld.exe** папки **bin** каталога установки MySQL;
  3. в поле **Путь к команде остановки** определите путь к файлу **mysqladmin.exe** папки **bin** каталога установки MySQL, в поле **Аргументы** введите **-u rootstop** для получения прав пользователя **root** на остановку сервера.



1. Запустите сервер базы данных MySQL на вашем компьютере.
2. Выберите пункт контекстного меню **Подключение** (перед попыткой подключения к серверу базы данных MySQL убедитесь в том, что он запущен на компьютере). В случае успешного подключения узел сервера MySQL в окне **Службы** выглядит следующим образом:



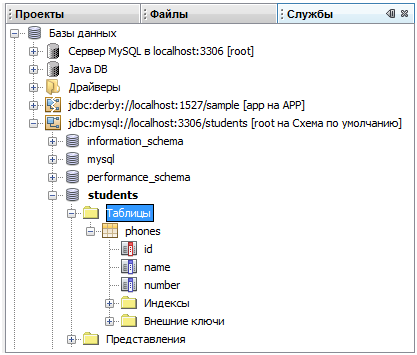
1. Создайте базу данных **students** (в среде NetBeans IDE существует встроенный редактор SQL, который является распространенным способом взаимодействия с базами данных):
   1. щелкните на узле **Сервер MySQL** правой кнопкой мыши;
   2. выберите пункт контекстного меню **Создать базу данных**;
   3. введите название базы данных **students**;
   4. в узле **Сервер MySQL** будет выведена новая база данных.
2. Создайте таблицу **phones**:
   1. в проводнике баз данных разверните узел подключения значок узла подключения **students**;



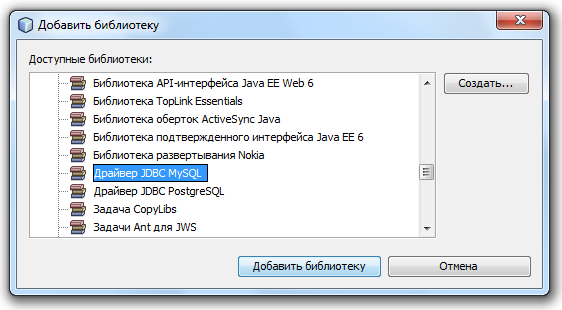
* 1. нажмите правой кнопкой мыши на пункт **Таблицы**;
  2. выберите пункт контекстного меню **Выполнить команду**, в редакторе SQL в главном окне отобразится пустой холст;
  3. введите следующий запрос в редакторе SQL:

|  |
| --- |
| CREATE TABLE phones  (  id INT (3) NOT NULL AUTO\_INCREMENT,  name VARCHAR (50) NOT NULL,  number BIGINT (10) NOT NULL,  PRIMARY KEY (id)  ) |

* 1. таблица **phones** содержит 3 поля: ключевое (**id**), имя (**name**), номер телефона (**number**);
  2. для выполнения запроса нажмите на кнопку **Выполнить SQL** верхней части окна или при помощи комбинации клавиш Ctrl+Shift+E;
  3. после выполнения запроса в окне **Вывод** будет создана обратная связь механизма SQL, указывающая на успешность выполнения или ошибку;
  4. для проверки изменений щелкните правой кнопкой мыши на узел **Таблицы** в проводнике баз данных и выберите команду **Обновить**.



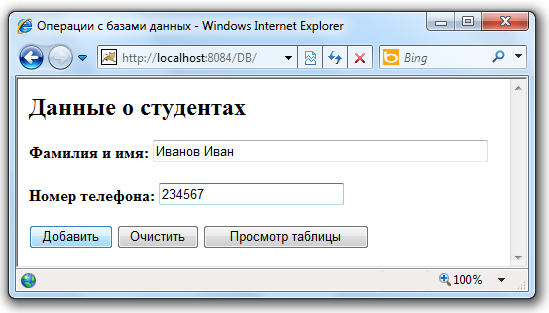
1. Подключите **JDBC-драйвер** для подсоединения к **MySQL**:
2. нажмите на название проекта в окне **Проекты** правой кнопкой мыши;
3. выберите пункт контекстного меню **Свойства**;
4. в списке **Категории** выберите **Библиотеки**;
5. нажмите на кнопку **Добавить библиотеку**;
6. в списке **Доступные библиотеки** выберите **Драйвер JDBCMySQL**, добавьте его.

****

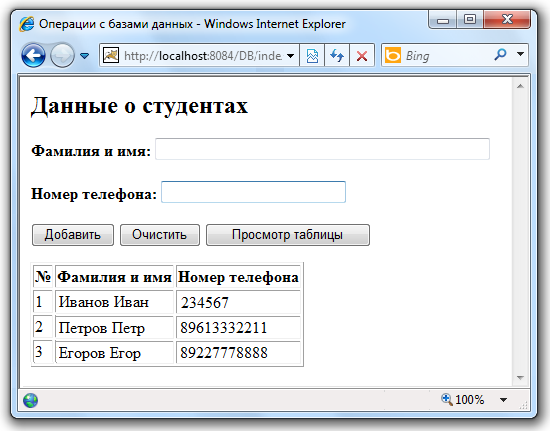
1. Сформируйте код страницы **index.jsp**:

|  |
| --- |
| <%@page language="java" import="java.sql.\*, java.io.\*"  contentType="text/html" pageEncoding="UTF-8"%>  <html>  <head>  <meta http-equiv="Content-Type"  content="text/html; charset=UTF-8">  <title> Операции с базами данных </title>  </head>  <body>  <%!  //объявление переменных  String sName, sNumber, user, pass, url,sql;  boolean flagName, flagNumber;  Connection conn;  Statement stmt;  ResultSet rs;  ResultSetMetaData rsm;  int colCount;  long iNumber;  %>  <%  sName = null;//имя  sNumber = null; //номер  flagName = flagNumber = false;  conn = null; //для установки соединения с БД  stmt = null; //для отправки инструкций SQL в базу данных  rs = null;/\* содержит возвращенные значения (таблицу данных) после выполнения SQL-команды \*/  rsm = null; /\*содержит количество, типы и свойства столбцов из объекта ResultSet \*/  url = null; // адрес для соединения с БД  user = "root"; // имя пользователя БД  pass = "root"; // пароль пользователя БД  colCount = 0; // количество столбцов в таблице БД  iNumber = 0; // номер  %>  <h2> Данные о студентах </h2>  <form action="index.jsp" method="get">  <p> <b> Фамилия и имя: </b>  <input type="text" name="name" size="50">  <p> <b> Номер телефона: </b>  <input type="text" name="number" size="25">  <p>  <input type="submit" value="Добавить" name="Add">  <input type="reset" value="Очистить">  <input type="submit" value="Просмотр таблицы"  name="Table">  </form>  <%  response.setContentType("text/html; charset=UTF-8");  request.setCharacterEncoding("UTF-8");  if(request.getParameter("Add") != null){  //проверка введенных данных  try{  sName=request.getParameter("name").toString().trim();  sNumber=request.getParameter("number").toString().trim();  iNumber = Long.parseLong(sNumber);  if (sName != "")  flagName = true;  else  out.println("<br> Введите имя");  if(sNumber != "")  flagNumber = true;  else  out.println("<br> Введите номер телефона");  }  catch(Exception e){  out.println(e.toString());  }  %>  <br>  <%  //Вставка в БД корректных данных  if (flagName && flagNumber){  try {  Class.forName("com.mysql.jdbc.Driver");  url = "jdbc:mysql://localhost/students";  conn =DriverManager.getConnection(url,user, pass);  stmt = conn.createStatement();  sql="insert into phones (name, number) values ('"+sName+"', "+iNumber+")";  stmt.execute(sql);  stmt.close();  conn.close();  }  catch(Exception ex){  out.println("<br> Ошибка при работе с БД <br>" + ex.toString());  }  out.println("<br> Данные успешно добавлены");  }  }  %>  <%  if(request.getParameter("Table") != null){  try{  Class.forName("com.mysql.jdbc.Driver");  url = "jdbc:mysql://localhost/students";  conn=DriverManager.getConnection(url, user, pass);  stmt = conn.createStatement();  rs = stmt.executeQuery("select \* from phones");  rsm = rs.getMetaData();  colCount = rsm.getColumnCount();  // Выводим данные таблицы phones из БД  out.println("<table border='1'>");  out.println("<th> № </th>");  out.println("<th> Фамилия и имя </th>");  out.println("<th> Номер телефона </th>");  while(rs.next()){  out.println("<tr>");  for(int i=1; i<=colCount; i++){  out.println("<td>"+rs.getString(i)+"</td>");  }  out.println("</tr>");  }  out.println("</table>");  stmt.close();  conn.close();  }  catch(Exception ex){  out.println("Ошибка при работе с БД <br> " + x.toString());  }  }  %>  </body>  </html> |

1. Запустите проект на выполнение и просмотрите результат работы в браузере:
   1. введите данные о нескольких студентах:



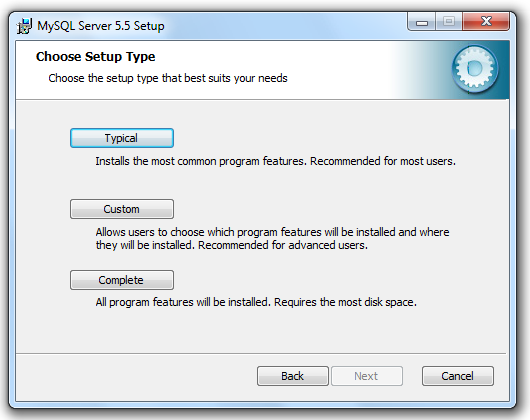
* 1. просмотрите данные таблицы:



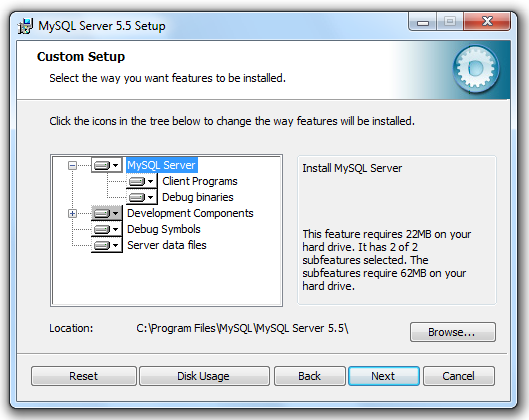
# 

# Приложение. Установка и настройка сервера MySQL

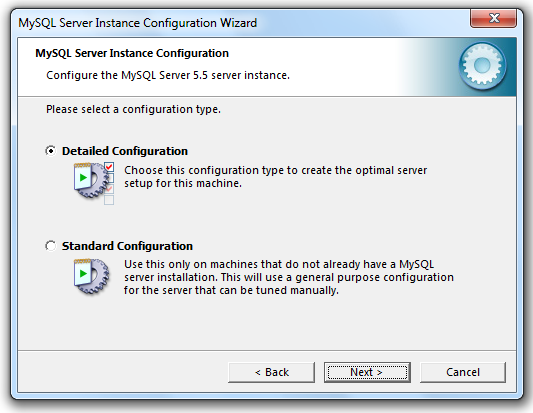
1. Скачайте дистрибутив сервера базы данных MySQL с официального сайта <http://dev.mysql.com/downloads/mysql/>
2. Запустите дистрибутив на выполнение.
3. Выберите режим установки **Custom,** нажмите на кнопку **Next**.



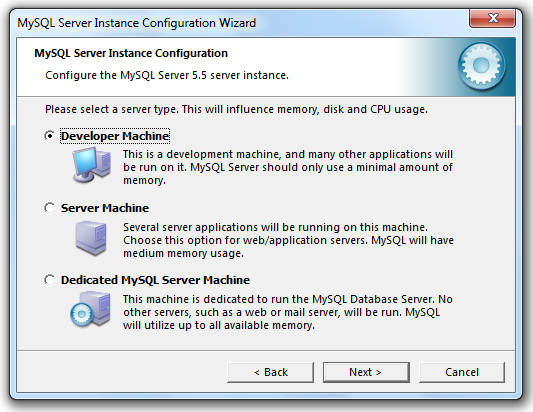
1. Ознакомьтесь с перечнем устанавливаемого программного обеспечения, нажмите на кнопку **Next**.



1. Продолжите установку, после ее завершения **Setup Wizard** предложит произвести конфигурацию БД и зарегистрироваться на сайте mysql.com, что позволит получать уведомления об обновлении сервера, подписаться на рассылку **MySQL Newsletter** и др.
2. В списке вариантов конфигурирования выберите **детальный (Detailed Configuration)**, нажмите на кнопку **Next**.



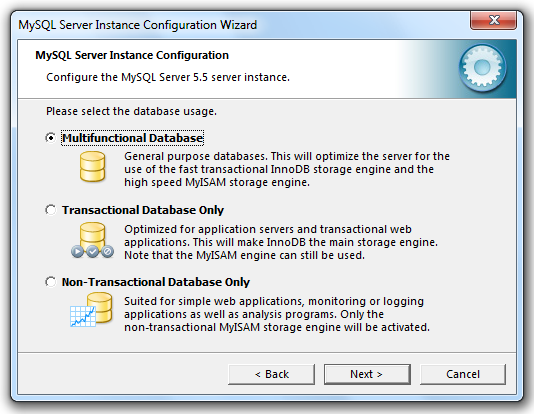
1. Выберите режим назначения сервера **Developer Machine**, который используется для отладки и тестирования, предоставляет минимальные требования к ресурсам.



**Server Machine**  подходит для рабочей базы данных, которая запускается на веб-сервере, где требуются повышенная производительность и устойчивость, соответственно и аппаратные ресурсы в таком варианте более значительные.

**Dedicated MySQL Server Machine** – все ресурсы сервера отдаются в монопольное использование MySQL и такая инсталляция будет обладать наибольшим быстродействием.

1. Выберите тип базы данных **Multifunctional Database**, который является наиболее универсальным и поддерживает как обычные таблицы (нескольких форматов), так и таблицы, в которых есть механизм транзакций.



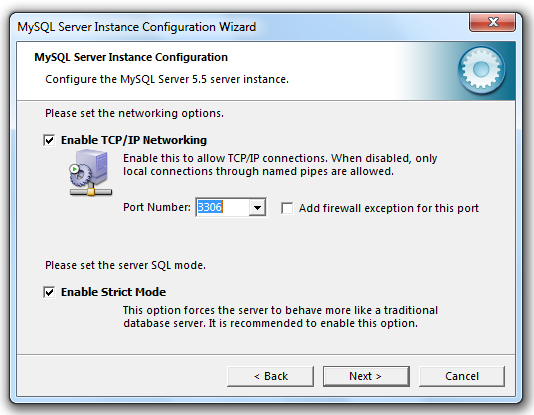
1. Для транзакционных таблиц формата **InnoDB** необходимо указать дополнительно логический диск и путь до каталога, где сервер будет хранить временные файлы транзакций.
2. Произведите настройку количества одновременных подключений. Выберите вариант Decision Support, которыйхарактеризуется небольшим количеством одновременных подключений, но в то же время запросы каждого клиента могут быть более сложными, включающими в себя группирующие функции, выборки одновременно с нескольких таблиц и сложной сортировкой.

Предел подключений – 20.

**Online Transaction Processing** характеризуется меньшей сложностью запросов, но более жесткими требованиями к производительности и количеству одновременно работающих пользователей.

**Manual Setting** позволяет самостоятельно установить число подключений, исходя из собственного анализа нагрузки при реальной работе.

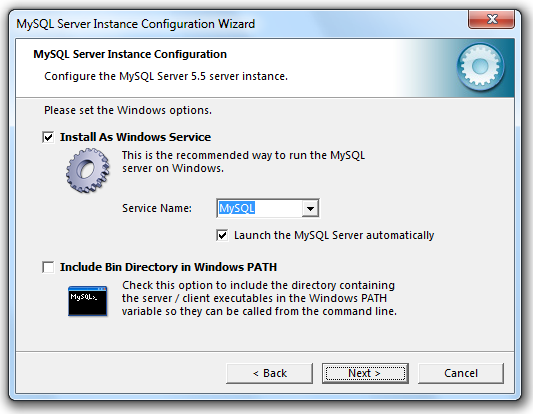
1. Выберите тип подключения к серверу по сети через TCP/IP, для этого MySQL использует по умолчанию порт 3306 **(**при необходимости можно его изменить).



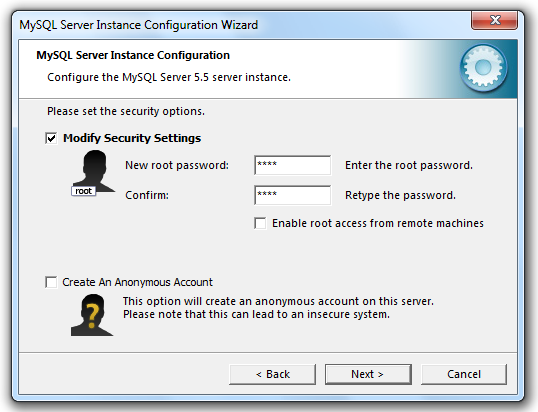
1. В списке установок кодировки выберите **Manual Selected Default Character Set/ Collation**, определите кодировку **UTF-8** для хранения текстовых данных.
2. Установите опции Windows.

**Install As Windows Service** позволяет установить сервис с именем, которое следует выбрать в раскрывающемся списке **Service Name**. Это дает возможность избежать конфликтов при запуске, если в системе уже имеются установленные сервера MySQL ранних версий.

**Include Bin Directory in Windows PATH** позволяет присвоить путь к каталогу с сервером системной переменной PATH.



1. В окне установки опций защиты выберите Modify Security Settings, при котором создается пользователь root. Введите пароль и его подтверждение, нажмите на кнопку Next.



1. В следующем окне нажмите на кнопку **Execute** для настройки сервера согласно данным указаниям.
2. Нажмите на кнопку **Finish** для завершения установки, настройки и конфигурирования MySQL.