



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

# Nhập môn Công nghệ phần mềm Introduction to Software Engineering (IT3180)

## **Lab Guide 3**

## Application Resources

Thông tin GV

# Nội dung

- Giới thiệu một vài loại tài nguyên khác nhau trong ứng dụng web
- Giới thiệu một số phương thức khác nhau thông qua đó các tài nguyên này được sử dụng trong ứng dụng
- Giới thiệu về Java Naming Directory Interface (JNDI)

# Tài nguyên - Resource

- Resources hay tài nguyên là những thành phần của một ứng dụng có thể được truy nhập đến trong chính mã nguồn của ứng dụng
- Một số ví dụ về tài nguyên: các thực thể quan hệ dữ liệu, dịch vụ mail v.v
- Các tài nguyên này có thể đặt ở các vị trí khác nhau (tập trung hoặc phân tán) tùy quy mô và mục đích sử dụng
- JNDI – Java Naming and Directory Interface được sử dụng để truy cập đến các thành phần tài nguyên khác nhau của ứng dụng

# Java Naming and Directory Interface (JNDI)

- Cho phép gắn mỗi tài nguyên ứng dụng với một cái tên cụ thể
- Gắn tên của một đối tượng với một đối tượng hoặc tham chiếu của đối tượng đó trong ứng dụng
- Tổ chức không gian tên (naming space) của ứng dụng theo cấu trúc phân cấp
- Định nghĩa ngữ cảnh cụ thể cho phép xác định tìm kiếm các đối tượng tài nguyên theo tên

# Database as Resource in Application

- Database là loại tài nguyên được sử dụng rộng rãi và thường xuyên trong các ứng dụng
- Mô hình dữ liệu quan hệ (data được biểu diễn trong các bảng) là mô hình dữ liệu được sử dụng rộng rãi
- Java cung cấp các API ví dụ Java Data Object (JDO) và Java Database Connectivity (JDBC) để truy cập và thao tác với CSDL

# JDBC – Java Database Connectivity

- JDBC truy cập tới CSDL thông qua các đối tượng DataSource
  - Mỗi đối tượng DataSource có các thuộc tính như: location, user credentials, protocol v.v.
- Đối tượng DataSource để truy cập được tới CSDL thì cần phải được đăng ký naming service với JNDI
- Naming Service cho phép ánh xạ một tên với một data source và cung cấp các tham chiếu để truy cập đến đối tượng dữ liệu

# Ví dụ về Binding a DataSource object với Database

- Mỗi thể hiện của đối tượng DataSource được tạo tương ứng với driver của database

```
com.microsoft.sqlserver.BasicDataSource ds =  
new com.microsoft.sqlserver.BasicDataSource();
```

- Đối tượng ds sau khi được khai báo sẽ được thiết lập các thuộc tính cần thiết để có thể thực hiện kết nối tới CSDL

```
ds.setServerName("Europa");  
ds.setDatabaseName("Student");  
ds.setDescription("Student Information");
```

# Ví dụ về Binding a DataSource object với Database

- Đối tượng ds sẽ được gán một JNDI tên:

```
Context c = new InitialContext();
```

```
c.bind("jndiTestDB");
```

- CSDL được truy cập thông qua các đối tượng `Connection`
- `PooledConnection` lưu trữ một tập hợp các đối tượng `Connection`
- Thông thường đối tượng `Connection` sẽ bị huỷ khi kết nối tới CSDL đóng lại
- Trong trường hợp của `PooledConnection`, đối tượng `Connection` được lấy từ pool và trả lại pool mỗi khi đóng kết nối



# Try Lab 03

- Tạo các JNDI name cho một kết nối tới CSDL (SQLServer hoặc MySQL)
- Tạo connection pool
- Truy cập và hiển thị dữ liệu trên trang web