



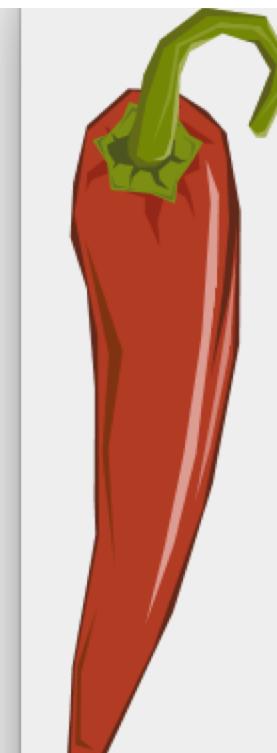
www.poly.edu.vn

LOMBOK GUIDE

LẬP TRÌNH JAVA

- ❑ Lombok là một API hỗ trợ cho các IDE tự sinh mã cho các lớp Java Bean, giúp lập trình viên bớt phần viết những đoạn mã nhảm chán.
- ❑ Các annotation Lombok cung cấp để sinh mã:
 - ❖ **@NoArgsConstructor** sinh mã constructor mặc định không tham số
 - ❖ **@AllArgsConstructor** sinh mã constructor đầy đủ tham số cho các field
 - ❖ **@Data** sinh mã cho các phương thức getters và setters cho các field
 - ❖ **@Getter** sinh mã cho các phương thức getters cho các field
 - ❖ **@Setter** sinh mã cho các phương thức setters cho các field
 - ❖ **@Builder** sinh mã theo pattern design của một builder

- Tải Lombok.jar: <https://projectlombok.org/download>
- Nhấp đúp vào Lombok.jar để chạy và restart eclipse



Javac (and tools that invoke javac such as *ant* and *maven*)
Lombok works 'out of the box' with javac.
Just make sure the lombok.jar is in your classpath when you compile.
Example: `javac -cp lombok.jar MyCode.java`

IDEs
Lombok can update your Eclipse or eclipse-based IDE to fully support all Lombok features.
Select IDE installations below and hit 'Install/Update'.

/Applications/eclipse-luna/Eclipse.app/Contents/MacOS/eclipse 

/Applications/eclipse-mars2/Eclipse.app 

/Applications/eclipse-neon/Eclipse.app 

[Specify location...](#) [Install / Update](#)

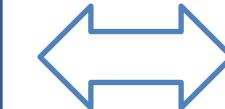
[Show me what this installer will do to my IDE installation.](#)
[Uninstall lombok from selected IDE installations.](#)

<https://projectlombok.org> v1.16.13 [Quit Installer](#)

```
<!--Lombok-->
<dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <version>1.18.34</version>
    <scope>provided</scope>
</dependency>
```

- ☐ Sử dụng **@NoArgsConstructor**, **@AllArgsConstructor** để sinh mã constructors

```
public class Account {  
    String username;  
    String password;  
  
    public Account() {}  
    public Account(String username, String password) {  
        this.username = username;  
        this.password = password;  
    }  
}
```



```
@NoArgsConstructor  
@AllArgsConstructor  
public class Account {  
    String username;  
    String password;  
}
```

SINH MÃ GETTERS & SETTERS

```
public class Account {  
    String username;  
    String password;  
    public String getUsername(){  
        return username;  
    }  
    public void setUsername(String username){  
        this.username = username;  
    }  
    public String getPassword(){  
        return password;  
    }  
    public void setPassword(String password){  
        this.password = password;  
    }  
}
```

- ☐ Sử dụng **@Data** để sinh mã getters và setters



```
@Data  
public class Account {  
    String username;  
    String password;  
}
```

SỬ DỤNG @BUILDER

@Builder

@NoArgsConstructor

@AllArgsConstructor

@Data

```
public class Account {  
    String username;  
    String password;  
}
```

Tạo object với Builder Design Pattern

Tạo object với constructor

```
Account user = Account.builder()  
    .username("teonv")  
    .password("123")  
    .build();
```

```
Account user = new Account();  
user.setUsername("teonv");  
user.setPassword("123");
```

☐ Ưu điểm với @Builder

- ❖ Không cần overloading nhiều constructor
- ❖ Một dòng code



DEMOSTATION



Cảm ơn