

21강. 스프링JDBC

- JDBC를 이용한 반복코드 줄이기
- Spring빈을 이용한 코드 간소화
- JDBC를 이용한 리스트 목록 만들기
- insert, update, delete 처리하기

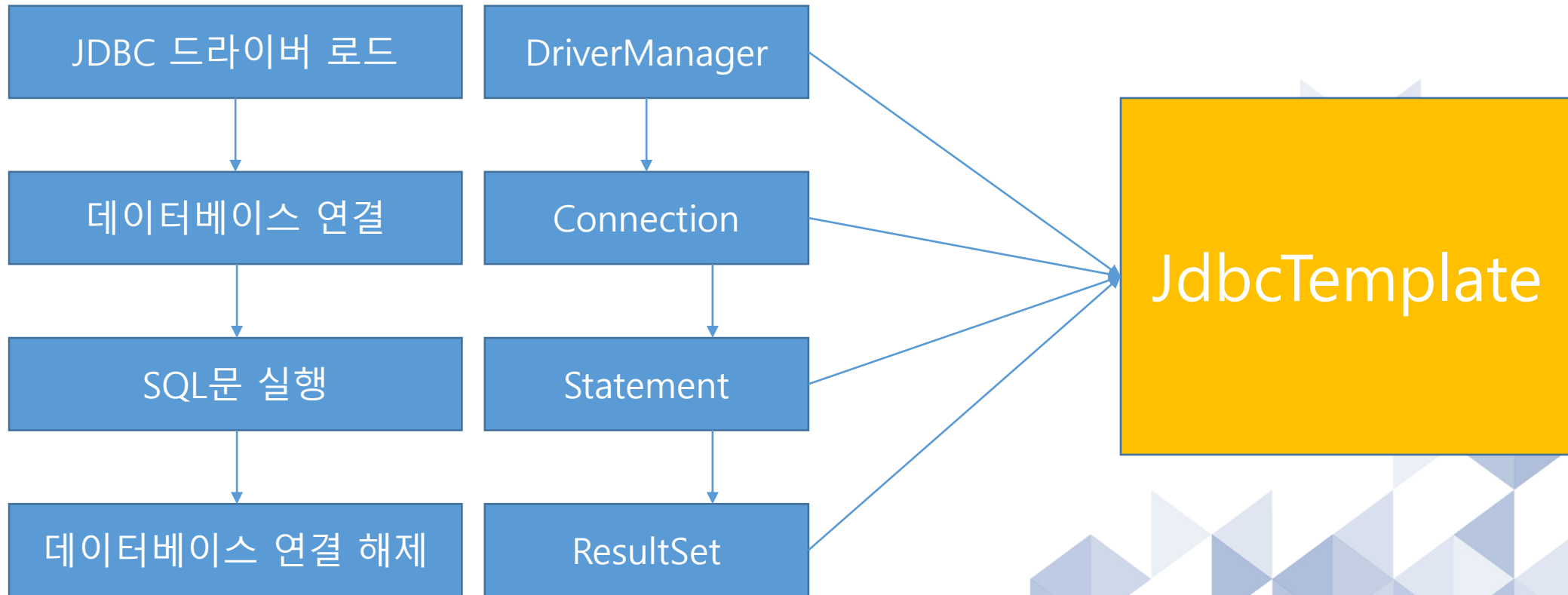
Lecturer Kim Myoung-Ho
Nickname 불스
blogstudy@naver.com

21-1. JDBC를 이용한 반복코드 줄이기

DAO객체를 이용해서 DataBase의 데이터를 이용 합니다.

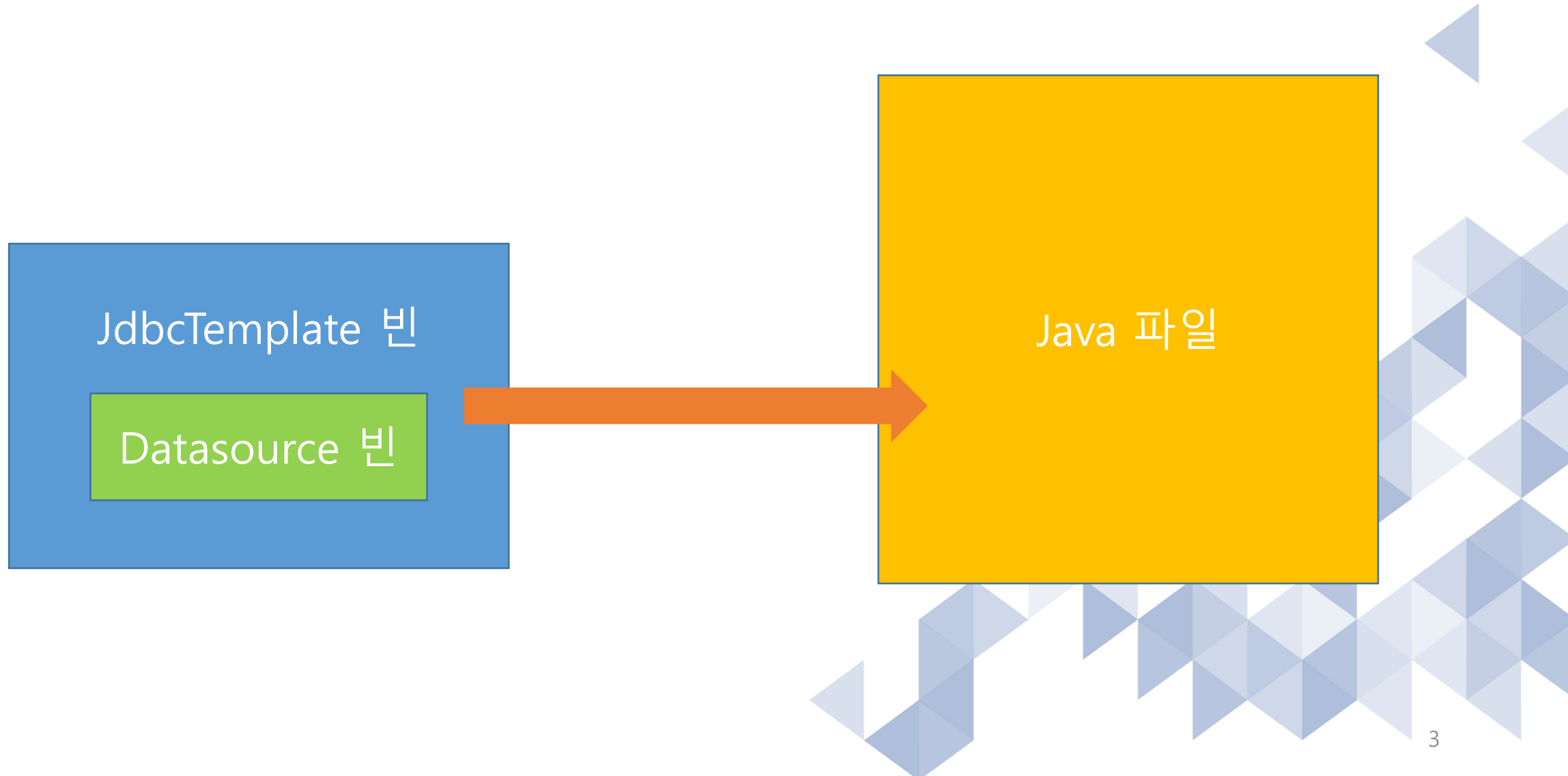
이때 매번 같은 동작을 반복하는 부분이 있습니다.(드라이버 로드, 커넥션 생성 및 DB연결, SQL실행, 자원해제)

이런 반복적인 작업을 스프링에서는 간단하게 처리 할 수 있습니다.



21-2. Spring빈을 이용한 코드 간소화

Spring빈을 이용하여 DataBase관련 클래스들을 생성하고 조립해 봅니다.



21-3. JDBC를 이용한 리스트 목록 만들기

JdbcTemplate의 메소드를 이용하여 코드를 간단하게 변경 할 수 있습니다.

```
public ArrayList<BDto> list() {  
  
    ArrayList<BDto> dtos = new ArrayList<BDto>();  
    Connection connection = null;  
    PreparedStatement preparedStatement = null;  
    ResultSet resultSet = null;  
  
    try {  
        connection = dataSource.getConnection();  
  
        String query = "select bld, bName, bTitle, bContent  
preparedStatement = connection.prepareStatement  
resultSet = preparedStatement.executeQuery();  
  
        while (resultSet.next()) {  
            int bld = resultSet.getInt("bld");  
            String bName = resultSet.getString("bName");  
            String bTitle = resultSet.getString("bTitle");  
            String bContent = resultSet.getString("bContent");  
            Timestamp bDate = resultSet.getTimestamp("bDate");  
            int bHit = resultSet.getInt("bHit");  
            int bGroup = resultSet.getInt("bGroup");  
            int bStep = resultSet.getInt("bStep");  
            int bIndent = resultSet.getInt("bIndent");  
  
            BDto dto = new BDto(bld, bName, bTitle, bContent, bDate, bHit, bGroup, bStep, bIndent);  
            dtos.add(dto);  
        }  
    } catch (Exception e) {  
        // TODO: handle exception  
        e.printStackTrace();  
    } finally {  
        try {  
            if(resultSet != null) resultSet.close();  
            if(preparedStatement != null) preparedStatement.close();  
            if(connection != null) connection.close();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```



```
public ArrayList<BDto> list() {  
  
    String query = "select bld, bName, bTitle, bContent, bDate, bHit, bGroup, bStep, bIndent  
return (ArrayList<BDto>) jdbcTemplate.query(query, new BeanPropertyRowMapper<BDto>(BDto.class));  
}
```

21-4. insert, update, delete 처리하기

JdbcTemplate을 이용하여 insert, update, delete 사용하는 메소드를 살펴 봅니다.

