

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Лабораторна робота №3

Тема: «Патер проектування Будівельник»

Дисципліна: «Ефективність та якість архітектурних рішень інформаційних систем»

Виконав: студент групи ІКМ-М224а  
Нестеренко Владислав Валентинович

Перевірів: Асистент кафедри  
Хорошун Андрій Сергійович

Харків - 2024р.

**Мета:** Здобути навички з реалізацією патерна проектування Будівельник.

### Завдання:

1. Ознайомитися з патерном Будівельник.
2. Виконати всі пункти лабораторної роботи
3. Надати звіт про виконану роботу звіт про виконану роботу.

### Хід роботи:

Уявимо що ми розробляємо систему яка буде взаємодіяти з декількома СУБД. Але клієнт не хоче використовувати готові рішення і просить нас написати свій власний QueryBuilder.

1. Система має підтримувати дві СУБД :
  - PostgreSQL
  - MySQL
2. Мають бути реалізовані методи:
  - select
  - where
  - limit
  - getSQL
3. В обох будівельників запитів має бути спільний інтерфейс.

Необхідно створити структуру класів та методів яка буде демонструвати реалізацію патерну Будівельник і буде вирішувати описане завдання. Навести приклад клієнтського коду з використанням звернення до обох СУБД. У рамках виконання лабораторної роботи не потрібно описувати деталі реалізації самих методів! Достатньо вказати сам метод та параметри який він приймає та повертає.

### Виконання:

Будівельник — це породжувальний патерн проектування, що дає змогу створювати складні об'єкти крок за кроком. Будівельник дає можливість використовувати один і той самий код будівництва для отримання різних відображень об'єктів.

Реалізація на Java:

Інтерфейс QueryBuilder - визначає методи для побудови SQL-запитів.

```
// Інтерфейс для всіх будівельників запитів
public interface QueryBuilder {
    QueryBuilder select(String columns);
    QueryBuilder where(String condition);
    QueryBuilder limit(int limit);
}
```

```
String getSQL();  
}
```

Клас PostgreSQLQueryBuilder -реалізує інтерфейс QueryBuilder для PostgreSQL.

```
// Реалізація будівельника для PostgreSQL  
public class PostgreSQLQueryBuilder implements QueryBuilder {  
    private StringBuilder query;  
  
    public PostgreSQLQueryBuilder() {  
        this.query = new StringBuilder();  
    }  
  
    @Override  
    public QueryBuilder select(String columns) {  
        query.append("SELECT ").append(columns).append(" ");  
        return this;  
    }  
  
    @Override  
    public QueryBuilder where(String condition) {  
        query.append("WHERE ").append(condition).append(" ");  
        return this;  
    }  
  
    @Override  
    public QueryBuilder limit(int limit) {  
        query.append("LIMIT ").append(limit).append(" ");  
        return this;  
    }  
  
    @Override  
    public String getSQL() {  
        return query.toString().trim();  
    }  
}
```

Клас MySQLQueryBuilder - реалізує інтерфейс QueryBuilder для MySQL.

```
// Реалізація будівельника для MySQL  
public class MySQLQueryBuilder implements QueryBuilder {  
    private StringBuilder query;  
  
    public MySQLQueryBuilder() {  
        this.query = new StringBuilder();  
    }  
  
    @Override  
    public QueryBuilder select(String columns) {  
        query.append("SELECT ").append(columns).append(" ");  
        return this;  
    }  
  
    @Override  
    public QueryBuilder where(String condition) {  
        query.append("WHERE ").append(condition).append(" ");  
        return this;  
    }  
  
    @Override  
    public QueryBuilder limit(int limit) {  
        query.append("LIMIT ").append(limit).append(" ");  
        return this;  
    }  
}
```

```

@Override
public String getSQL() {
    return query.toString().trim();
}
}

```

Клас QueryDirector - використовує будівельника для побудови конкретного запиту

```

// Клас-директор, який керує процесом побудови запиту
public class QueryDirector {
    private QueryBuilder builder;

    public QueryDirector(QueryBuilder builder) {
        this.builder = builder;
    }

    // Метод для побудови простого запиту
    public String buildSimpleQuery() {
        return builder
            .select("*")
            .where("age > 18")
            .limit(10)
            .getSQL();
    }
}

```

Клієнтський код: демонструє створення запитів для обох СУБД, (QueryDirector) і будівельники (PostgreSQLQueryBuilder та MySQLQueryBuilder):

```

//Клієнтський код
public class Main {
    public static void main(String[] args) {
        // Створення будівельника для PostgreSQL
        QueryBuilder postgresBuilder = new PostgreSQLQueryBuilder();
        // Створення директора з використанням будівельника для PostgreSQL
        QueryDirector postgresDirector = new QueryDirector(postgresBuilder);
        // Побудова запиту для PostgreSQL
        String postgresQuery = postgresDirector.buildSimpleQuery();
        System.out.println("PostgreSQL Query: " + postgresQuery);

        // Створення будівельника для MySQL
        QueryBuilder mysqlBuilder = new MySQLQueryBuilder();
        // Створення директора з використанням будівельника для MySQL
        QueryDirector mysqlDirector = new QueryDirector(mysqlBuilder);
        // Побудова запиту для MySQL
        String mysqlQuery = mysqlDirector.buildSimpleQuery();
        System.out.println("MySQL Query: " + mysqlQuery);
    }
}

```

## Висновки

Під час виконання даної лабораторної роботи я здобув навички з реалізації патерна проектування Будівельник. А також написав свій власний QueryBuilder з дотриманням умов поставленого завдання.