

The background features a dark navy blue field. On the left side, there are several overlapping geometric shapes: a bright blue parallelogram, a light green parallelogram, and a dark grey parallelogram, all oriented diagonally. The text 'DAO – Data Access Object' is centered in the right half of the image in a white, sans-serif font.

DAO – Data Access Object

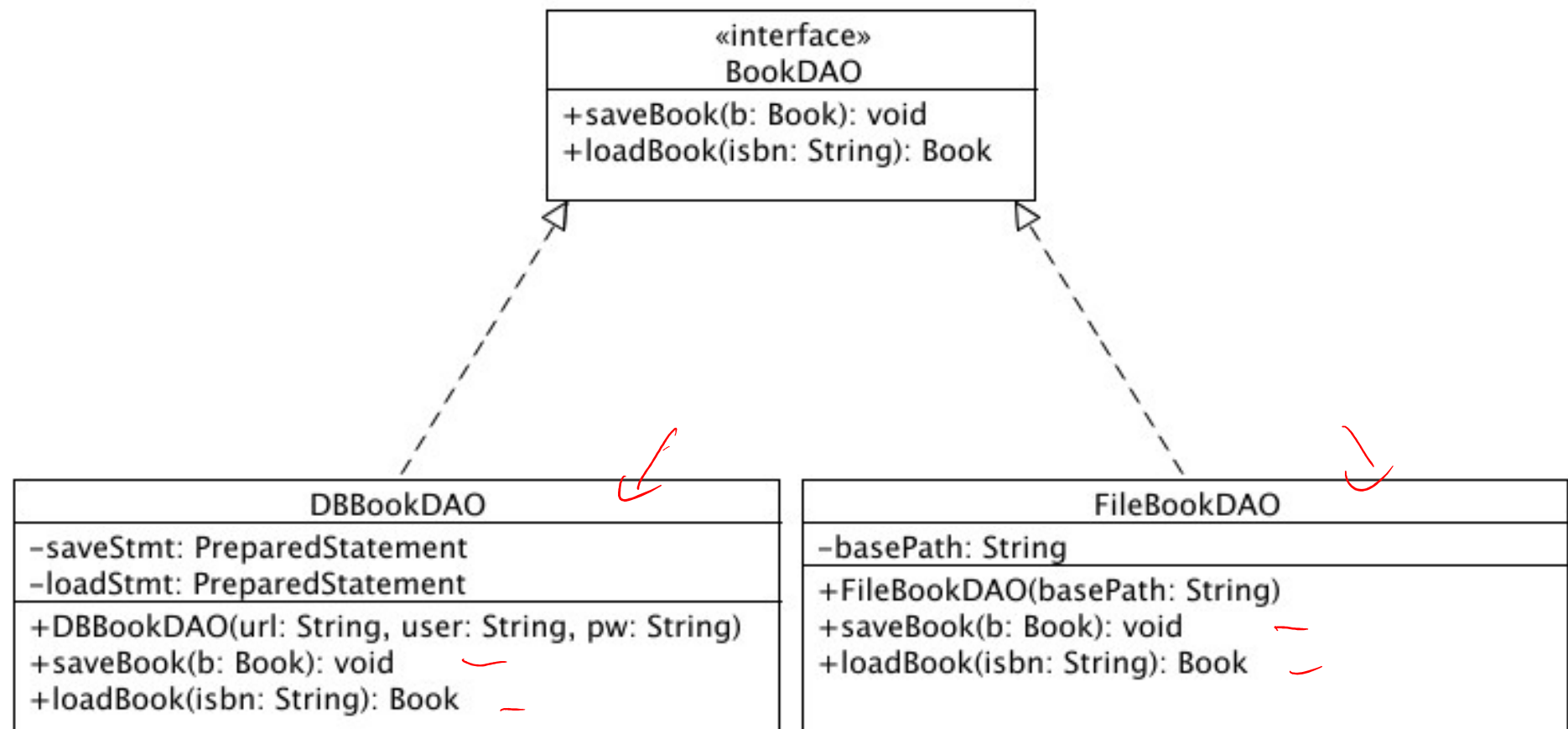


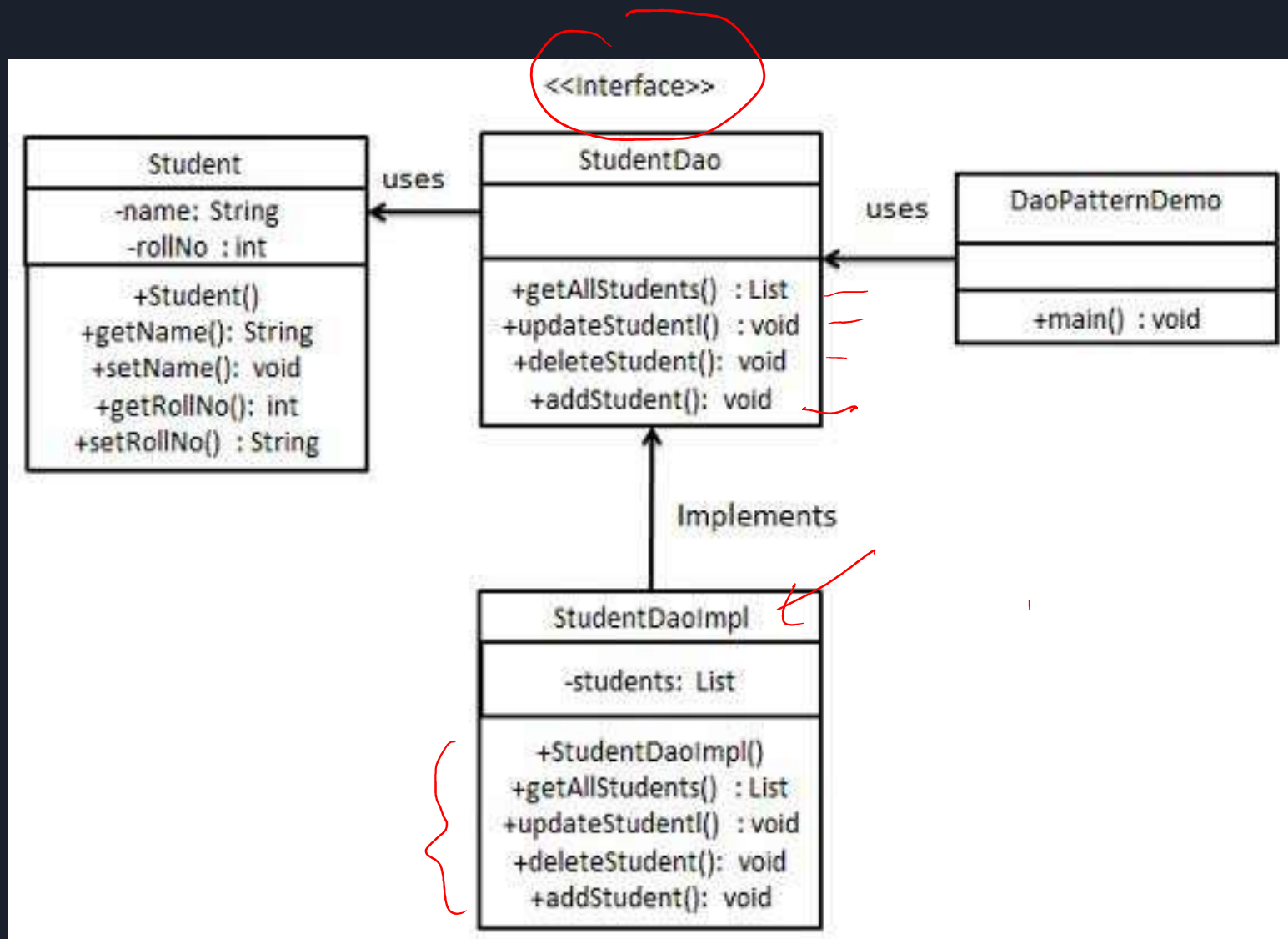
DAO – Data Access Object


- Một design pattern phổ biến
- Tạo ra các đối tượng để tương tác với DB

- save ()
- findById ()
- find All ()
- update ()

- delete ()
- deleteAll ()







```
public interface SinhVienDAO {  
    List<SinhVien> getAll();  
    SinhVien getById(int id);  
    void save(SinhVien sinhVien);  
    void update(SinhVien sinhVien);  
    void delete(int id);  
}
```

Lưu ý

- Chúng ta sẽ cần dùng **EntityManager** , **Datasource** để thực hiện các thao tác với DB
- Được tạo ra tự động dựa trên file cấu hình
- Chúng ta sử dụng @Autowired để inject vào các DAO



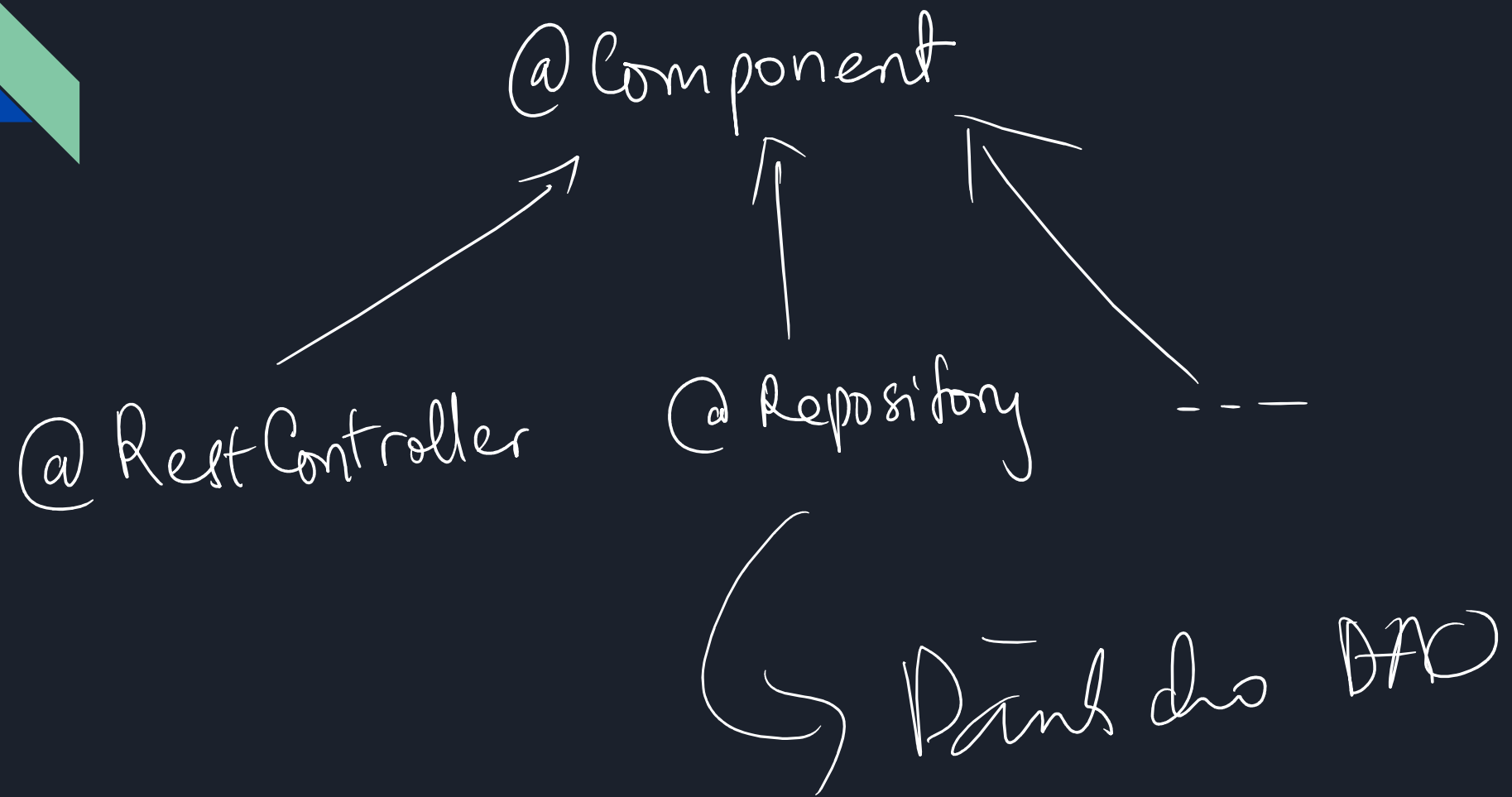
@Component


@Repository

```
public class SinhVienDAOImpl implements SinhVienDAO {  
    private final EntityManager entityManager;
```

→ @Autowired

```
public SinhVienDAOImpl(EntityManager entityManager) {  
    this.entityManager = entityManager;  
}
```





```
@Override  
public void save(SinhVien sinhVien) {  
    entityManager.persist(sinhVien);  
}
```

→ *lưu save*

Transaction

- Trong Hibernate, một transaction (giao dịch) là một khối mã hoặc một tác vụ logic được thực hiện trên cơ sở dữ liệu. Giao dịch được sử dụng để đảm bảo tính nhất quán và độ tin cậy của dữ liệu trong hệ thống.
- Khi bạn bắt đầu một giao dịch, Hibernate sẽ theo dõi tất cả các thay đổi dữ liệu được thực hiện trong phiên làm việc hiện tại. Các thay đổi này bao gồm việc thêm, cập nhật hoặc xóa các đối tượng được quản lý bởi Hibernate.





Transaction

- Giao dịch có hai trạng thái chính: chưa bắt đầu (not started) và đã kết thúc (ended). Trong quá trình giao dịch, bạn có thể thực hiện các hoạt động như lấy dữ liệu, thay đổi dữ liệu, và lưu dữ liệu vào cơ sở dữ liệu.
- Nếu giao dịch thành công, tất cả các thay đổi sẽ được xác nhận và cập nhật vào cơ sở dữ liệu. Ngược lại, nếu giao dịch gặp lỗi, tất cả các thay đổi sẽ bị hủy và không được áp dụng vào cơ sở dữ liệu.



```
public void saveStudent(Student student) {  
    Transaction transaction = null;  
    try (Session session = HibernateUtil.getSessionFactory().openSession()) {  
        // start a transaction  
        transaction = session.beginTransaction();  
        // save the student object  
        session.save(student);  
        // commit transaction  
        transaction.commit();  
    } catch (Exception e) {  
        if (transaction != null) {  
            transaction.rollback();  
        }  
        e.printStackTrace();  
    }  
}
```

start a transaction

commit a trasaction

rollback trasaction

@Transactional

- Đánh dấu @Transactional

```
@Repository  
@Transactional  
public class SinhVienDAOImpl implements SinhVienDAO {  
    private final EntityManager entityManager;
```

```
    @Override  
    @Transactional(readonly = true)  
    public SinhVien getById(int id) {  
        return entityManager.find(SinhVien.class, id);  
    }
```