



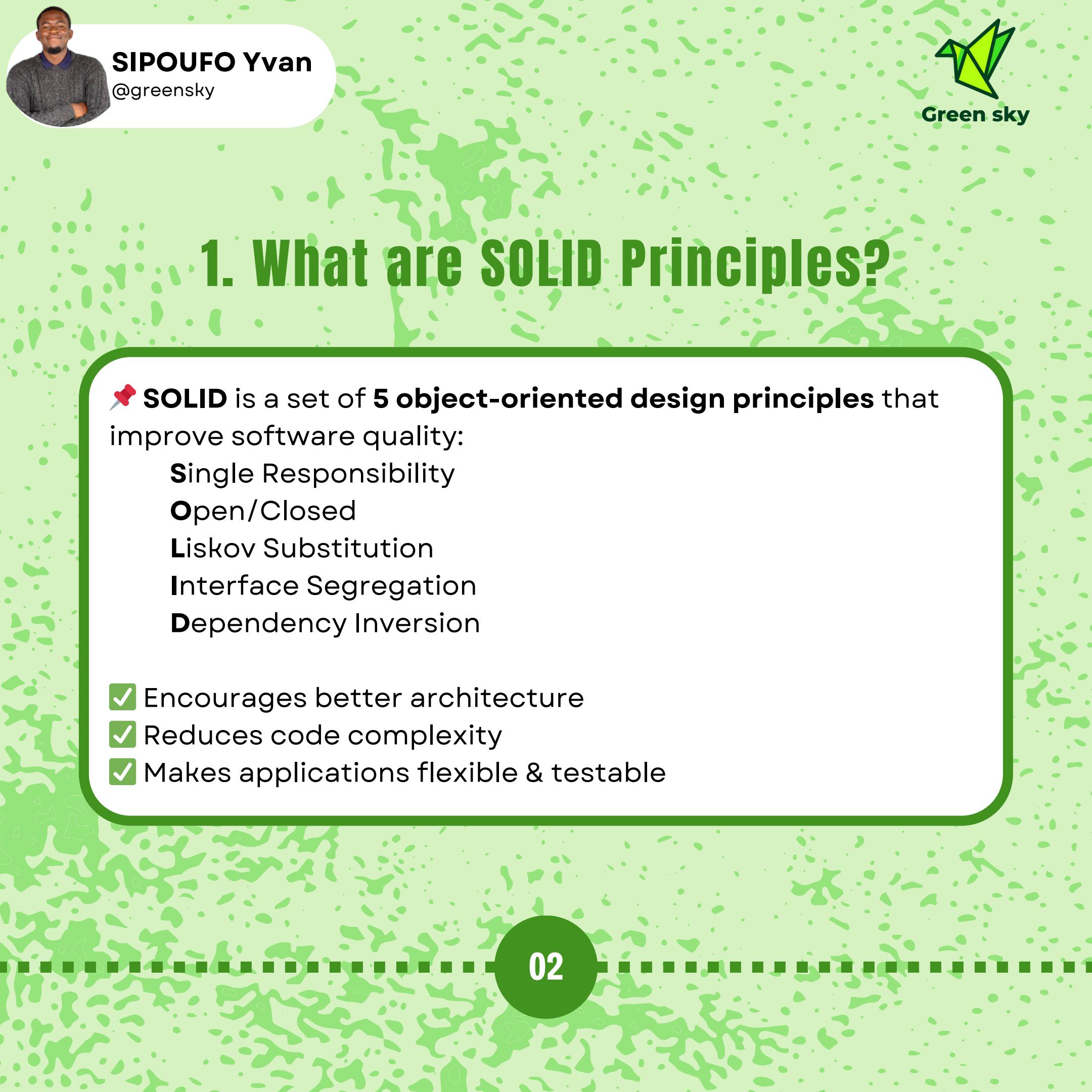
SIPOUFO Yvan
@greensky



MASTER SOLID PRINCIPLES

with Spring Boot





SIPOUFO Yvan

@greensky



1. What are SOLID Principles?

📌 **SOLID** is a set of **5 object-oriented design principles** that improve software quality:

- Single Responsibility
- Open/Closed
- Liskov Substitution
- Interface Segregation
- Dependency Inversion

- ✓ Encourages better architecture
- ✓ Reduces code complexity
- ✓ Makes applications flexible & testable

02



SIPOUFO Yvan

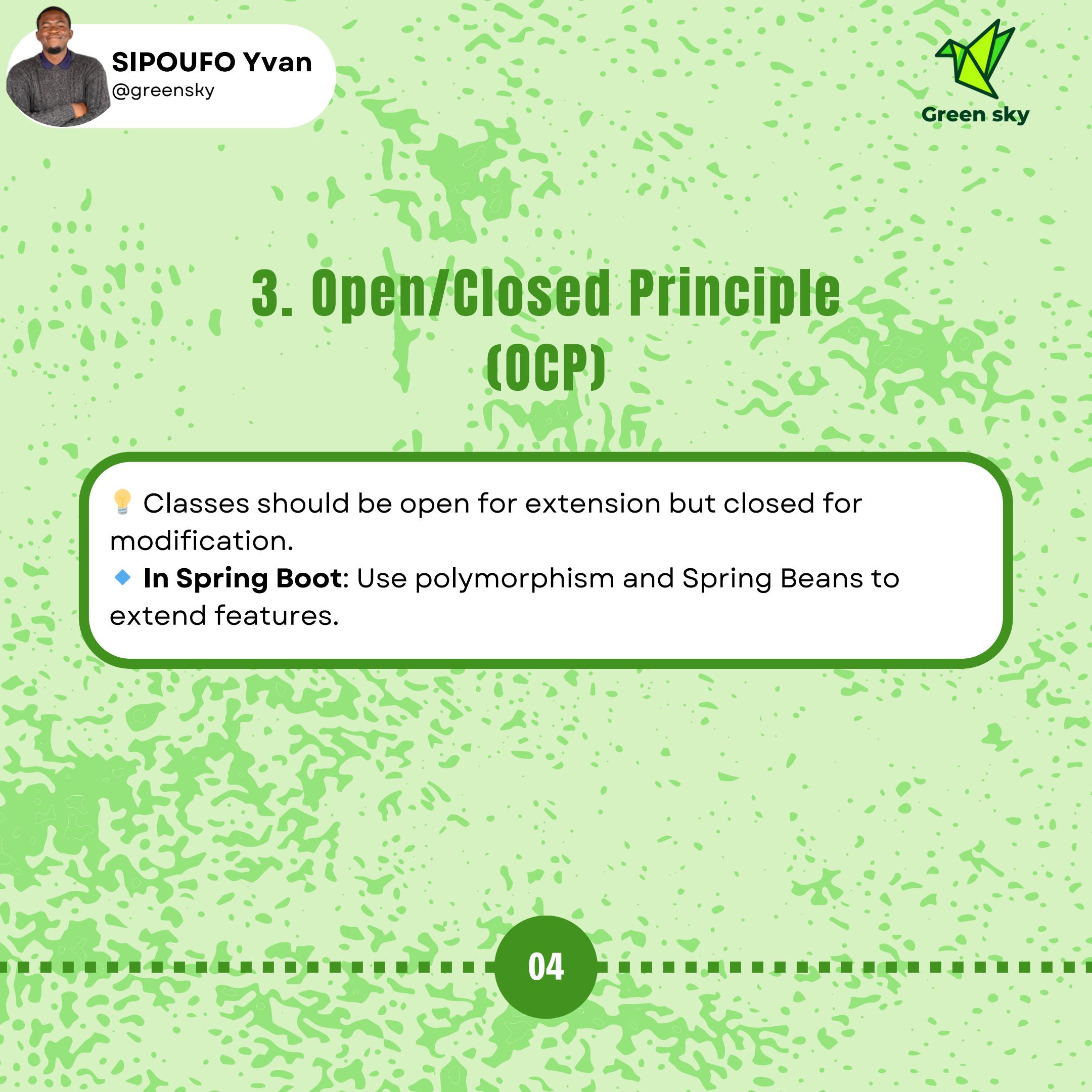
@greensky



2. Single Responsibility Principle (SRP)

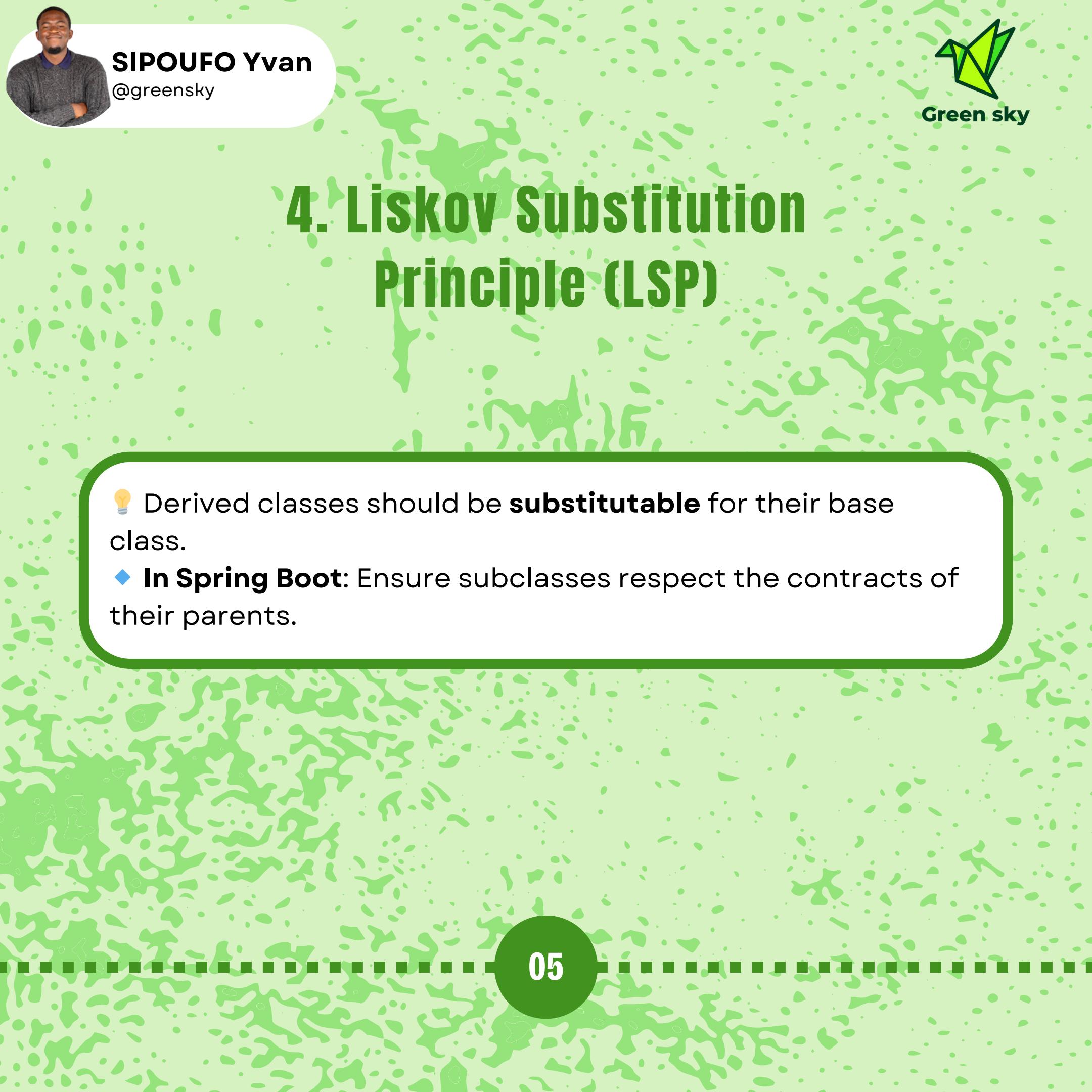
- 💡 A class should have only one reason to change.
- ◆ In **Spring Boot**: Split services, controllers, and repositories clearly.

03



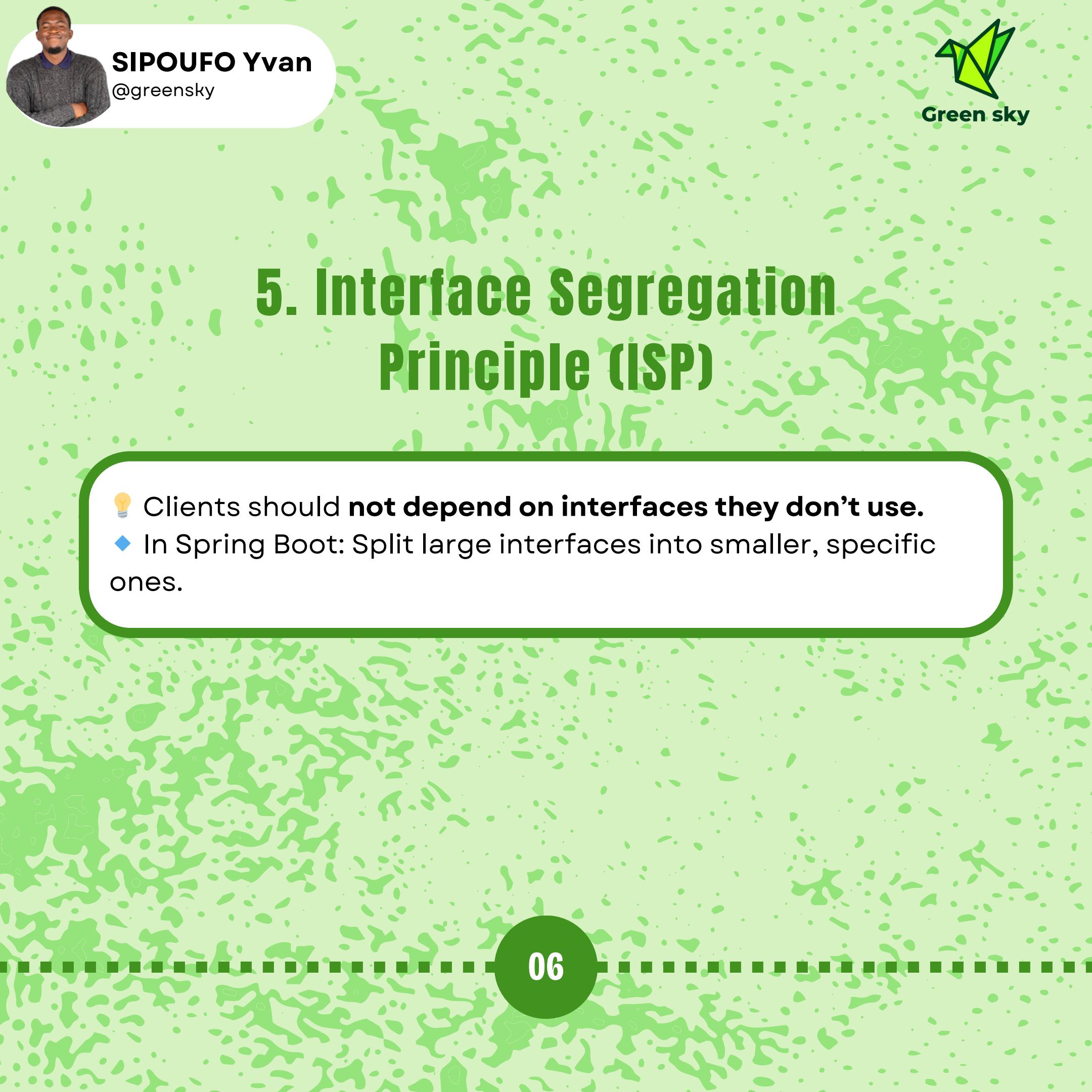
3. Open/Closed Principle (OCP)

- 💡 Classes should be open for extension but closed for modification.
 - ◆ **In Spring Boot:** Use polymorphism and Spring Beans to extend features.



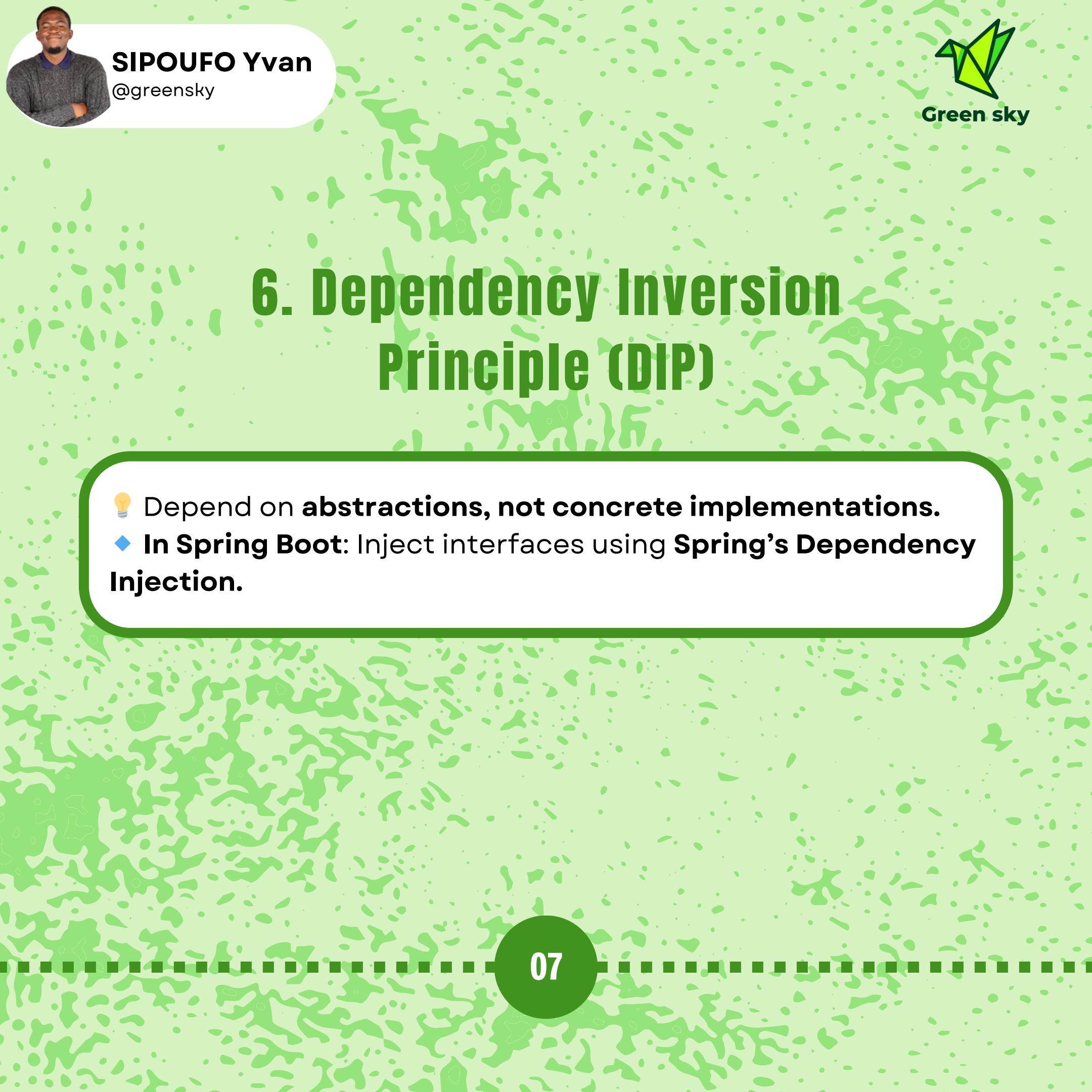
4. Liskov Substitution Principle (LSP)

- 💡 Derived classes should be **substitutable** for their base class.
 - ◆ **In Spring Boot:** Ensure subclasses respect the contracts of their parents.



5. Interface Segregation Principle (ISP)

- 💡 Clients should **not depend on interfaces they don't use.**
 - ◆ In Spring Boot: Split large interfaces into smaller, specific ones.



6. Dependency Inversion Principle (DIP)

- 💡 Depend on **abstractions**, not concrete implementations.
- ◆ In **Spring Boot**: Inject interfaces using **Spring's Dependency Injection**.



SIPOUFO Yvan

@greensky



7. Conclusion

- ◆ **SOLID + Spring Boot** = Clean & maintainable architecture
- ◆ Better testability, scalability & flexibility
- ◆ Essential for enterprise-grade Java applications
- 🚀 Start refactoring your codebase with SOLID principles today!



SIPOUFO Yvan
@yvansipoufo29



Upwork
@Yvan Sipoufo



Fiverr
@yvansipoufo

ABUI NGAN !

Thanks in Ewondo 😊



Follow us for more dev-fun