1. The dependency injection pattern
2. Types of dependency injection patterns
3. Resolving dependency using Abstract Factory Pattern
4. Lookup-method injection pattern
5. Configuring beans using the Factory pattern
6. Configuring dependencies
7. Common best practices for configuring dependencies in an application.

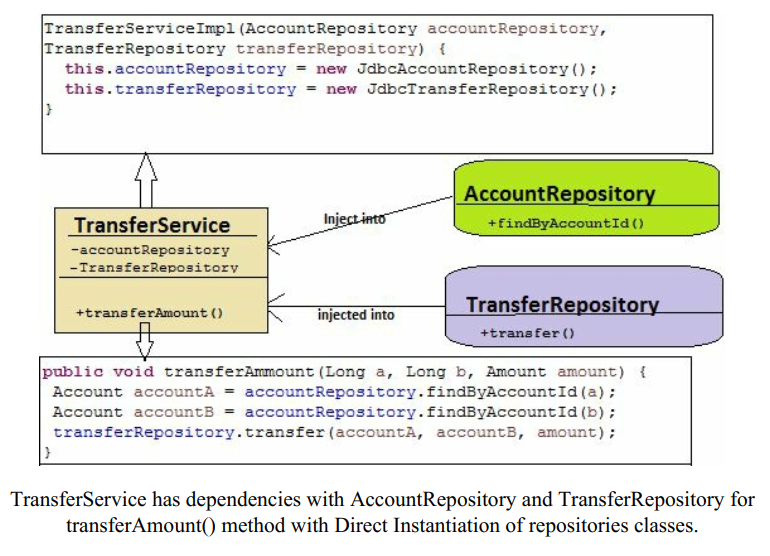
* Tight-coupling
* Loosely-coupling

1. The dependency injection pattern

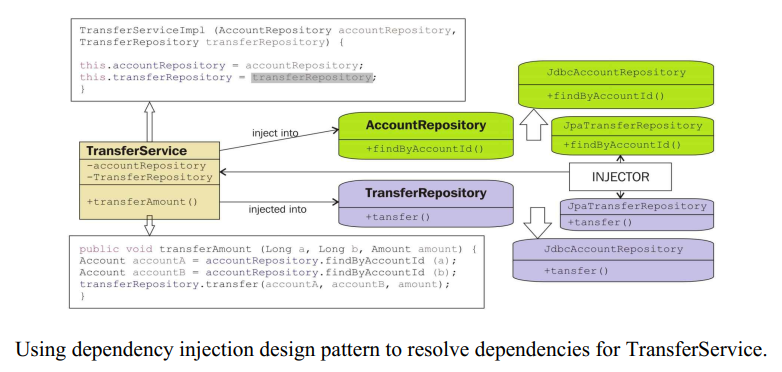
* Ưu tiên sử dụng interface thay cho class
* Giảm phụ thuộc giữa các class

1. Solving problems using the dependencies injection pattern.

* Đặt vấn đề:
  + Trong bất kỳ một dự án, một vấn đề chung cần xử lý là làm thế nào để configure và liên kết các yếu tố lại với nhau để tạo ra business goal.
  + VD: làm thế nào để kết nối các controller lại với nhau tại layer web với service và repository interfaces được viết bởi các thành viên khác nhau của team mà không biết về controller của layer web
* Without dependency injection



* With dependency injection



* Chú thích:
  + TransferService interface
  + TransferServiceImp class implements TransferService
  + AccountReponsitory interface
  + TransferReponsitory interface
  + JdbcAccountRepository class implements AccountReponsitory
  + JdbcTransferReposiroty class implements TransferReponsitory
* Spring DI dựa vào ý tưởng của IoC.
* Spring framework có một container là nơi tạo, quản lý và phá hủy object. Nó được biết đến là Spring IoC Container.
* Object nằm bên trong Spring container được gọi là Spring Bean.
* Có rất nhiều cách liên kết các bean trong một Spring application.

1. Type of dependency injection patterns

* Constructor-based dependency injection.
* Setter-based dependency injection.

1. Contructor – based dependency injection.

* Ưu điểm khi sử dụng contructor injection pattern
  + Phù hợp cho các dependencies bắt buộc và tạo ra một giao kèo dependencies mạnh mẽ
  + Cung cấp một mã gọn hơn những cấu trúc khác.
  + Hỗ trợ việc testing bằng cách sử dụng các tham số đầu vào cho các dependent class.
  + Không phá vỡ nguyên tắc hiding data.
* Nhược điểm: có thể gây ra sự phụ thuộc vòng tròn. A phụ thuộc B và B cũng phụ thuộc A.

Setter – based dependency injection