

APPLICATION FRAMEWORKS

49

49

Application Framework

- Top-down:
 - Third party e.g. ASP.NET
 - Dangers: Lock-in, code bloat
- Bottom-up:
 - Enterprise encoding of domain knowledge
 - E.g. ET++SwapManager

50

50

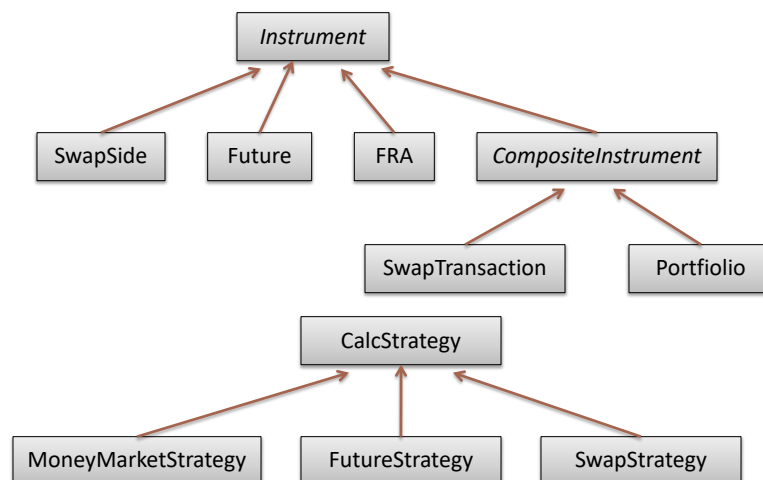
Application Framework

- Benefits:
 - Modularity
 - Reusability
 - Extensibility
 - *Inversion of Control*

51

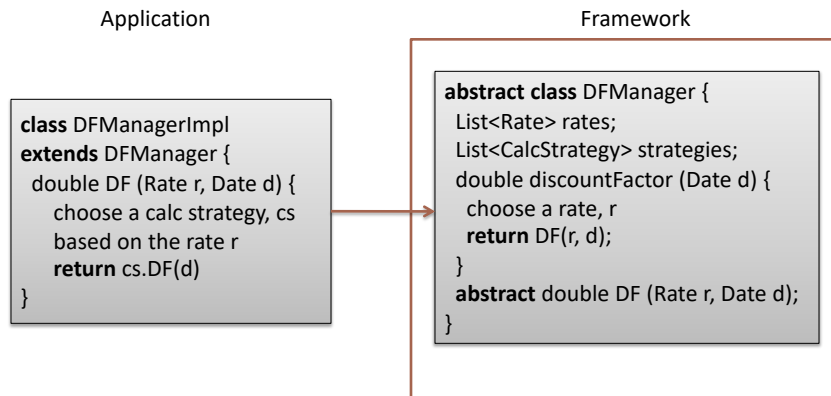
51

Encoding Domain Knowledge



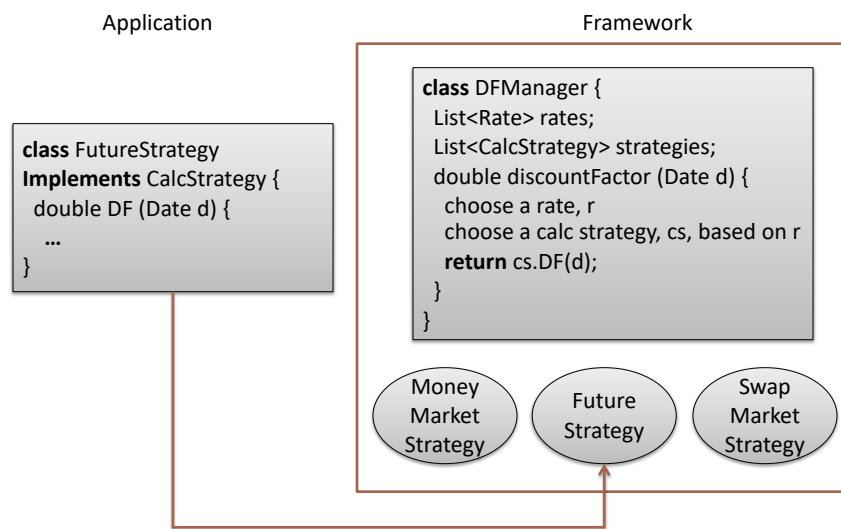
52

White-Box Reuse



53

Black-Box Reuse



54

Inversion of Control

- Traditional library reuse:
 - Library is passive
 - Application imports library and uses operations

```
import java.io;
InputStream is = new FileInputStream("...");
```
- Application framework reuse:
 - Framework starts as program
 - *Calls into client extensions*

55

55

Dependency Injection

- A form of IoC
- Container injects resources that a service needs to execute
 - Jakarta EE (CDI)
 - ASP.NET MVC

56

56

Dependency Injection

- Inject a “repository” object:

```
@PersistenceContext(unitName = "PatientPU")  
EntityManager em;
```

- Inject a message queue:

```
@Resource(name = "jms/Requests")  
Queue requests;
```

- Inject a “service” object:

```
@Inject  
IService server;
```

57