

DAO

Тема 12

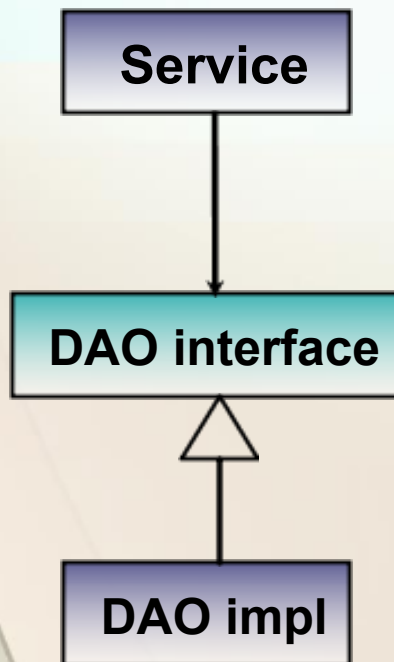
Технологии проектирования программных систем

Вопросы

- **Что такое DAO?**
- **Шаблон «Template Method»**
- **Spring DAO Templates**
- **Spring DAO Support**
- **Оптимизация загрузки объектов**

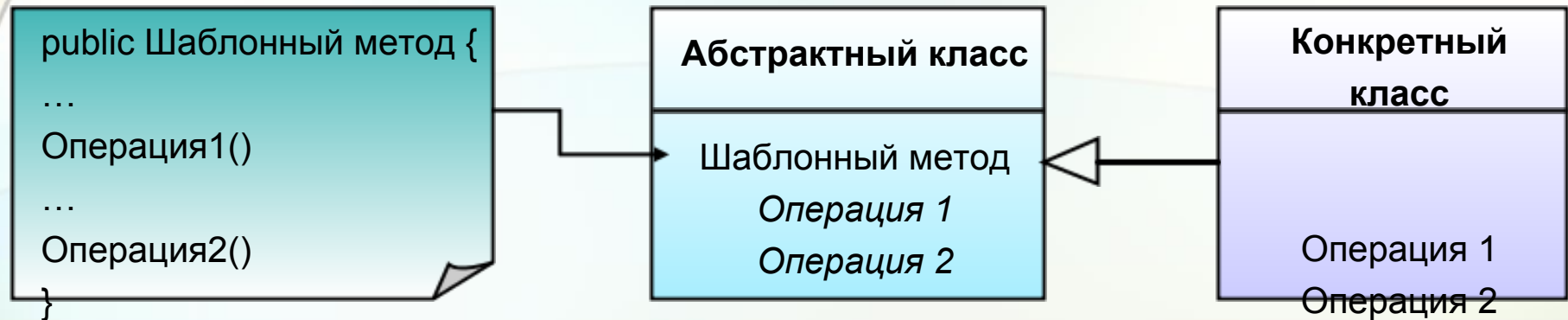
Что такое DAO?

Data Access Object – Объект, выполняющий операции с БД и предоставляющий доступ к этим операциям через интерфейс



Шаблон «Template Method»

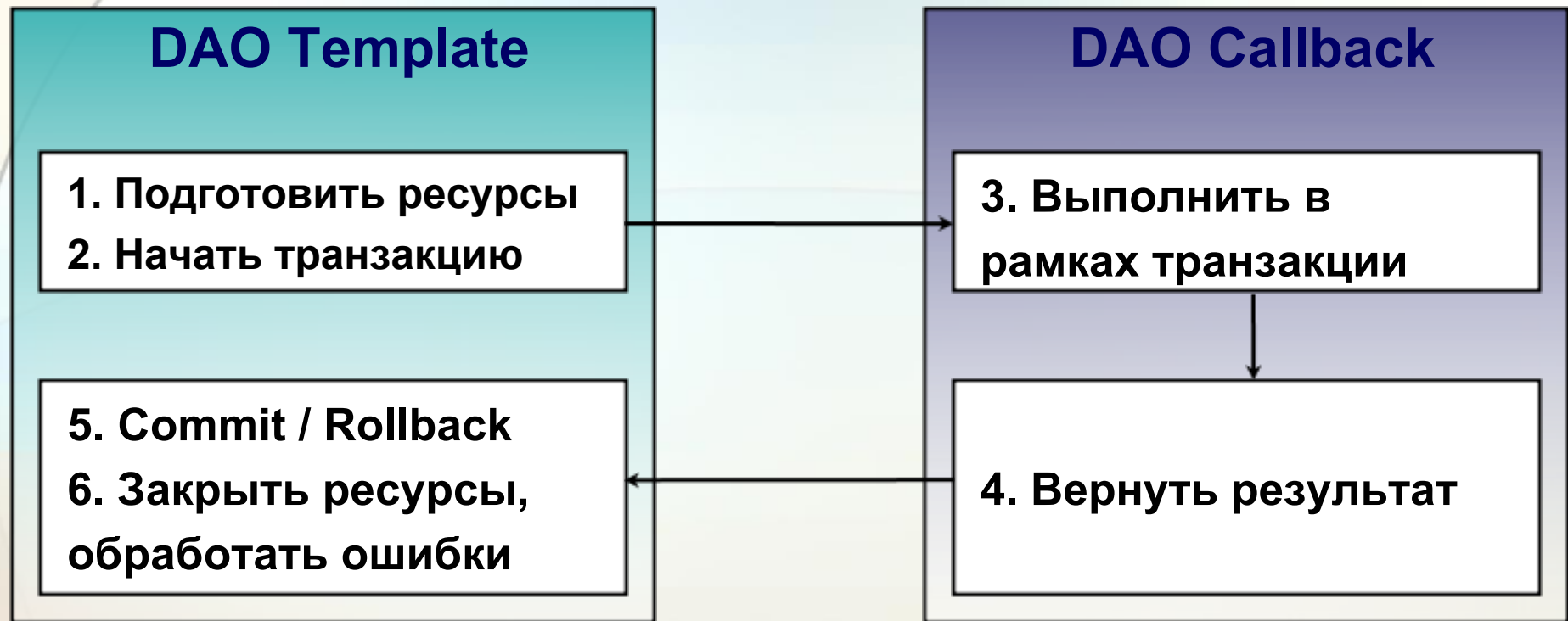
Определяет основу алгоритма и позволяет подклассам переопределить некоторые шаги алгоритма, не изменяя его структуру в целом



Применимость:

- Однократное использование неизменяемых частей алгоритма; подклассы определяют изменения
- Избежание дублирования кода в разных подклассах
- Управление расширением функциональности через подклассы

Spring DAO Templates



- **orm.jpa.JpaTemplate**
- jdbc.core.JdbcTemplate
- orm.hibernate3.HibernateTemplate

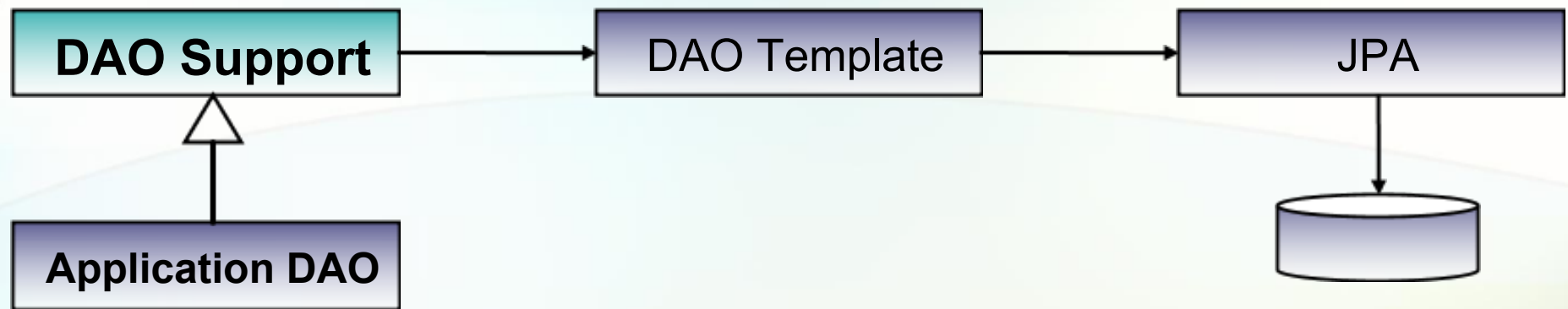
Использование DAO Template

```
<bean id="jpaTemplate"
      class="org.springframework.orm.jpa.JpaTemplate"
>
    <property name="entityManagerFactory" ref="emf" />
</bean>
```

```
public class UserService implements IUserService {
    public UserService() {}
    ...
    private JpaTemplate jpaTemplate;
    public void setJpaTemplate(JpaTemplate template) {
        this.jpaTemplate = template;
    }
}
```

```
<bean id="userService" class="service.impl.UserService">
    <property name="jpaTemplate" ref="jpaTemplate" />
</bean>
```

Spring DAO Support



- **orm.jpa.support.JpaDaoSupport**
- **jdbc.core.support.JdbcDaoSupport**
- **orm.hibernate3.support.HibernateDaoSupport**

Использование DAO Support

```
<bean id="userService" class="service.impl.UserService">  
  <property name="entityManagerFactory" ref="emf" />  
</bean>
```

```
public class UserService extends JpaDaoSupport  
                           implements  
    IUserService {  
    public void saveUser(User user) {  
        getJpaTemplate().persist(user);  
    }  
}
```


Выводы

Организация слоя интеграции:

- Используйте DAO для доступа к объектам, которые хранятся не в БД
- Для доступа к объектам в БД используйте JPA (напрямую или через Spring) прямо из сервисов



Еще раз об инъекции

```
public class UserService implements IUserService {  
    private JpaTemplate template;  
    private ICommentService commentService;  
    private IRSSDao rssDao;  
}
```

```
<bean id="userService" class="service.impl.UserService">  
    <property name="template" ref="jpaTemplate" />  
    <property name="commentService" ref="commentService" />  
    <property name="rssDao" ref="rssDao" />  
</bean>
```

```
<bean id="commentService" class="service.impl.CommentService">  
    <property name="template" ref="jpaTemplate" />  
</bean>
```

```
<bean id="rssDao" class="dao.impl.RomeRssDao" />
```

Стратегии выборки объектов

Lazy fetching («ленивая» выборка) – Объект выбирается только при первом обращении к нему.

Eager fetching («активная» выборка) – Объект выбирается вместе с родительским объектом, используя SQL оператор *outer join*.
Дочерние объекты выбираются в дополнительных SELECT запросах.

Hibernate:

Batch fetching (пакетная выборка) –
Используется для улучшения «ленивой» выборки, когда выбирается сразу несколько объектов.

Настройка стратегии выборки в JPA

```
public class Category {  
    @OneToMany(fetch=LAZY)  
    private List<Item> items;  
}  
  
public class Item {  
    @ManyToOne(fetch=EAGER)  
    private Category category;  
}
```

Пакетная выборка в хибернейт

```
public class Category {  
    @OneToMany(fetch=LAZY)  
    @BatchSize(size=10)  
    private List<Item> items;  
}
```

```
    @BatchSize(size=10)  
public class Item {  
}
```