

Spring-Data-***

Гибкость и велосипедостроение

Беляев Андрей, CUBA Platform

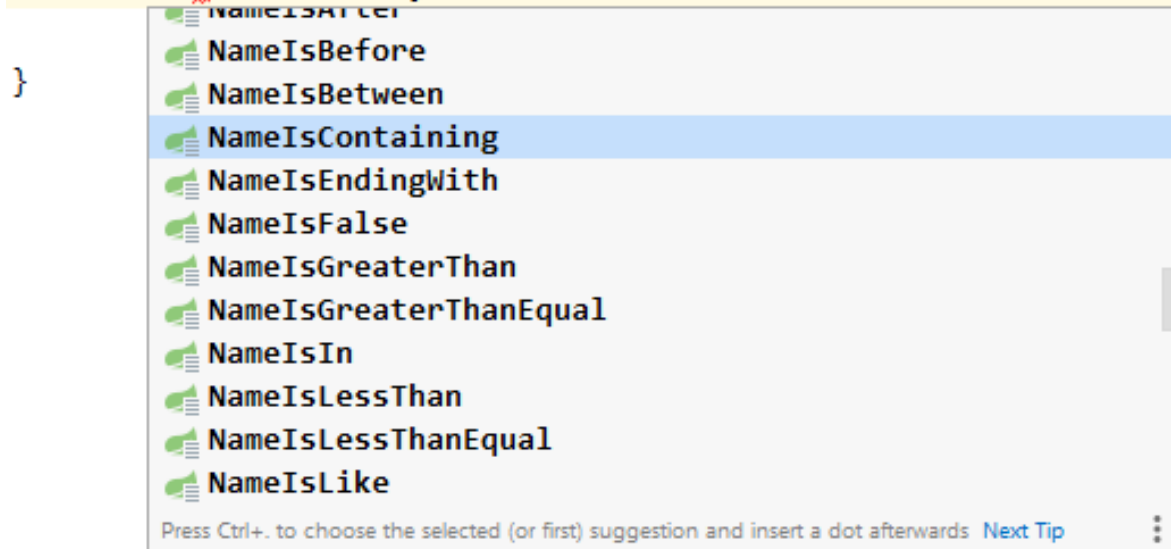
 [belyaev_andrey](#)

Все (почти) любят Spring

Там есть Data Repositories

```
public interface PersonRepository extends CrudRepository<Person, Long> {  
  
    long count();  
  
    void delete(Person person);
```

Person.findPersonBy



Data Repositories

ГИБКИЕ

- **Base Repository**

Содержит все необходимые CRUD методы

- **Custom Repository + *Impl**

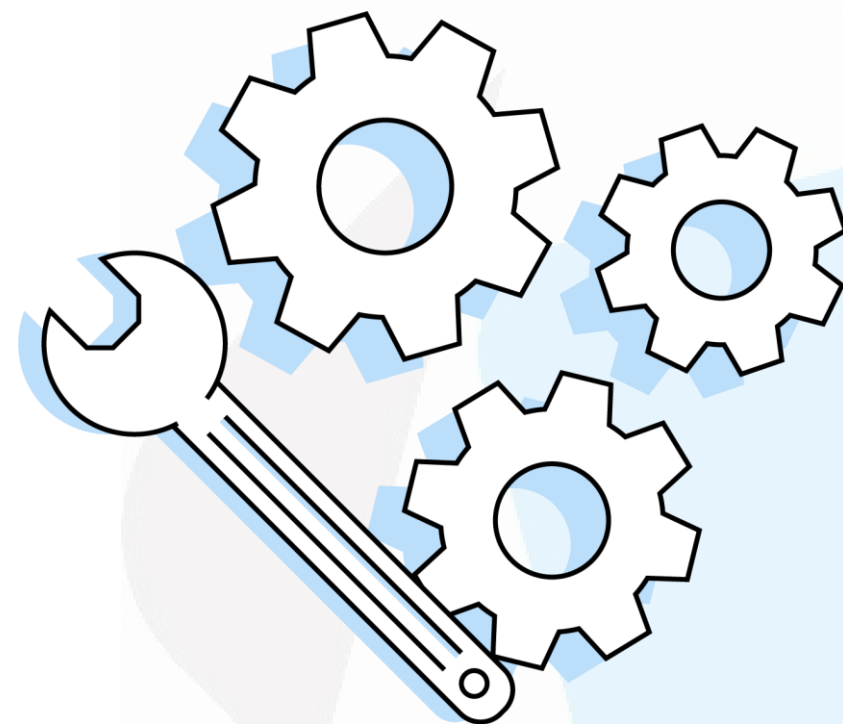
Можно перекрывать методы базового репозитория

- **Методы со специальными названиями**

`findPersonByNameAndAddress`

- **@Query – аннотация**

Можно писать JPQL и SQL запросы



Data Repositories - JPA

```
public interface PersonRepository extends CrudRepository<Person, Long>, PersonExtension {  
  
    long count();  
  
    void delete(Person person);  
  
    Person save(Person p);  
  
    Person findPersonByName(String name);  
  
    List<Person> findPersonByBirthDateBefore(LocalDate date);  
}
```

```
public interface PersonExtension {  
  
    Person save(Person s);  
  
    Person findPersonByName(String name);  
}
```

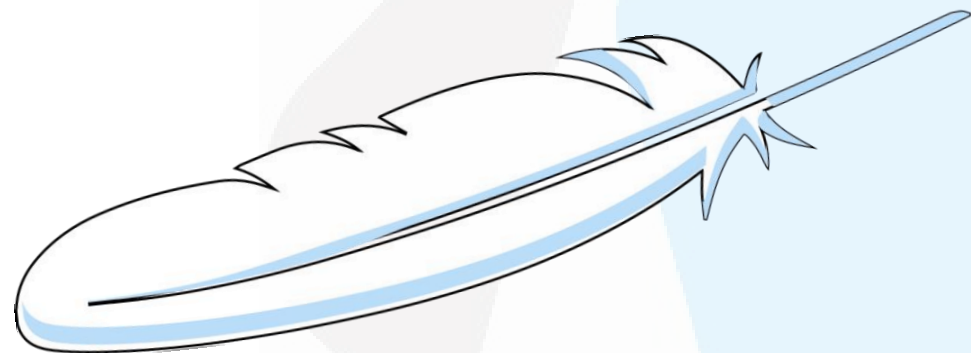
```
public class PersonExtensionImpl implements PersonExtension {  
  
    @PersistenceContext  
    private EntityManager entityManager;  
  
    @Override  
    @Transactional  
    public Person save(Person s) {  
        //Custom save code  
    }  
  
    @Override  
    public Person findPersonByName(String name) {  
        //Custom search code  
    }  
}
```

Хочется добавить немного CUVA

- Row-Based Security
 - Предикаты для ролей
- Soft Delete
 - На уровне Data Repositories
 - Желательно отключаемый

Добавляем Soft Delete

- “Логическое” удаление
- **DELETE** - обновление специального атрибута
- **SELECT, UPDATE** – не должны затрагивать “удаленные” строки
- **Решение** – добавить WHERE ко всем запросам
 - Желательно – с наименьшими усилиями
 - Можно сделать в Hibernate (@Where)
 - Но это – привязка к провайдеру



Soft Delete

Идея не новая

<https://jira.spring.io/browse/DATAJPA-307>

✓ Add support for soft deletes

Details

Type: + New Feature

Status: OPEN

Priority: ^ Major

Resolution: Unresolved

<https://jira.spring.io/browse/DATAJPA-948>

✓ Provide extensibility of query creation

✓ ● **Jens Schauder** added a comment - 28/May/19 12:36 PM

We won't implement this as it currently is proposed.

But I created a separate issue in an attempt to come up with a maintainable API for this kind of stuff: [DATAJPA-1551](#)

Soft Delete

<https://jira.spring.io/browse/DATAJPA-1551>

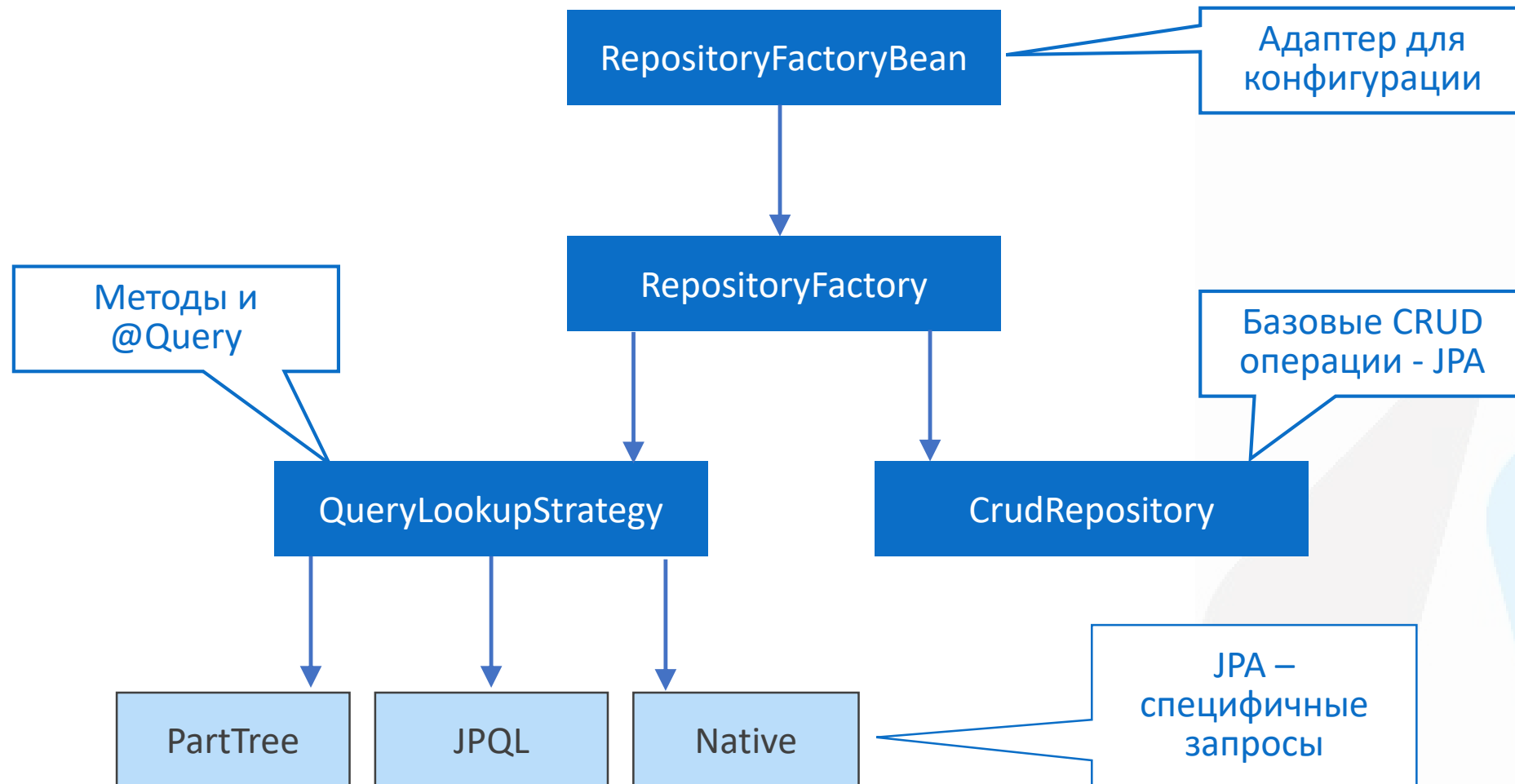
- ✓ Create an API to customize query creation and query processing

Details

Type:  New Feature
Priority:  Minor

Status: **OPEN**
Resolution: Unresolved

Внутренности Spring-Data-(JPA)



Spring-Data-Commons

```
public interface CrudRepository<T, ID extends Serializable> extends Repository<T, ID>
{
    <S extends T> S save(S entity);

    T findOne(ID id);

    boolean exists(ID id);

    Iterable<T> findAll(Iterable<ID> ids);

    long count();

    void delete(ID id);

    void deleteAll();
    ...
}
```

```
public interface QueryLookupStrategy {
    RepositoryQuery resolveQuery(Method method, RepositoryMetadata metadata,
                                ProjectionFactory factory, NamedQueries namedQueries);
}
```

Spring-Data-***

```
public class CassandraRepositoryFactoryBean<T extends Repository<S, ID>, S, ID>
    extends RepositoryFactoryBeanSupport<T, S, ID> {
    . . .
}
```

```
public class MongoRepositoryFactoryBean<T extends Repository<S, ID>, S, ID extends Serializable>
    extends RepositoryFactoryBeanSupport<T, S, ID> {
    . . .
}
```

```
public class SimpleCassandraRepository<T, ID> implements CassandraRepository<T, ID>
{
    . . .
}
```

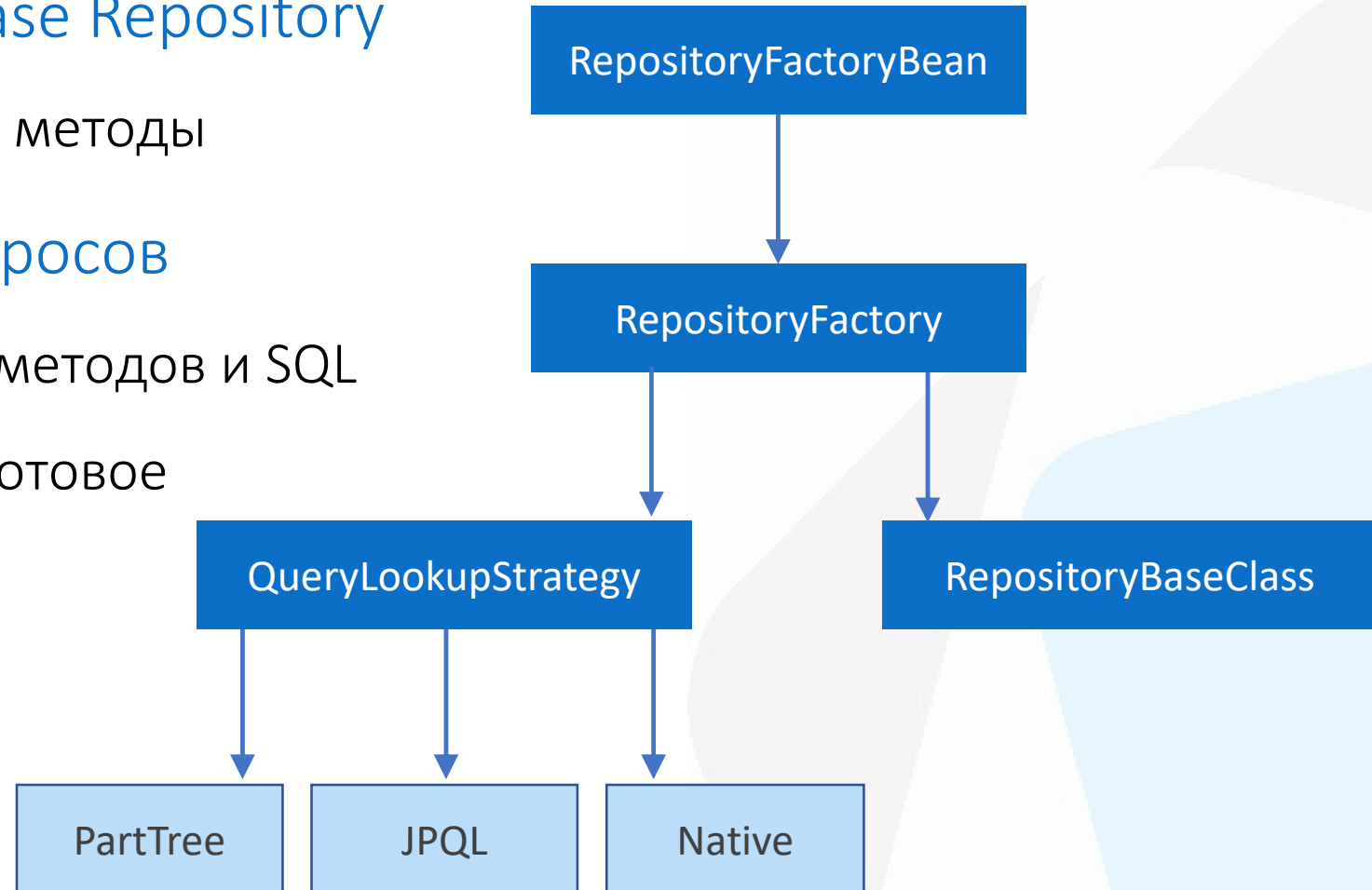
```
public class SimpleMongoRepository<T, ID> implements MongoRepository<T, ID> {
    . . .
}
```

```
private class CassandraQueryLookupStrategy implements QueryLookupStrategy {
    . . .
}
```

```
private static class MongoQueryLookupStrategy implements QueryLookupStrategy {
    . . .
}
```

Что нужно сделать?

- Поправить реализацию Base Repository
 - Реализовать нужные CRUD методы
- Поправить генерацию запросов
 - Не хочется парсить имена методов и SQL
 - Лучше переиспользовать готовое
- Будем делегировать или наследоваться!



Soft Delete

```
public class BaseEntity implements Serializable {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;

    @Version
    private Long version = 0L;

    @CreatedDate
    private LocalDate createdAt;

    @LastModifiedDate
    private LocalDate lastModified;

    @DeletedDate
    private LocalDate deletedDate;
```

```
@NoRepositoryBean
public interface SoftDeleteRepository<T, ID> {

    void setSoftDeleteEnabled(boolean softDeleteEnabled);

    boolean isSoftDeleteEnabled();

    void purge(T entity);

    void purgeById(ID id);

    void purgeAll(Iterable<? extends T> entities);

    void purgeAll();
}
```

Soft Delete

```
public class CubaCrudRepositoryImpl<T, ID>  
    extends SimpleJpaRepository<T, ID>  
    implements SoftDeleteRepository<T, ID> {
```

...

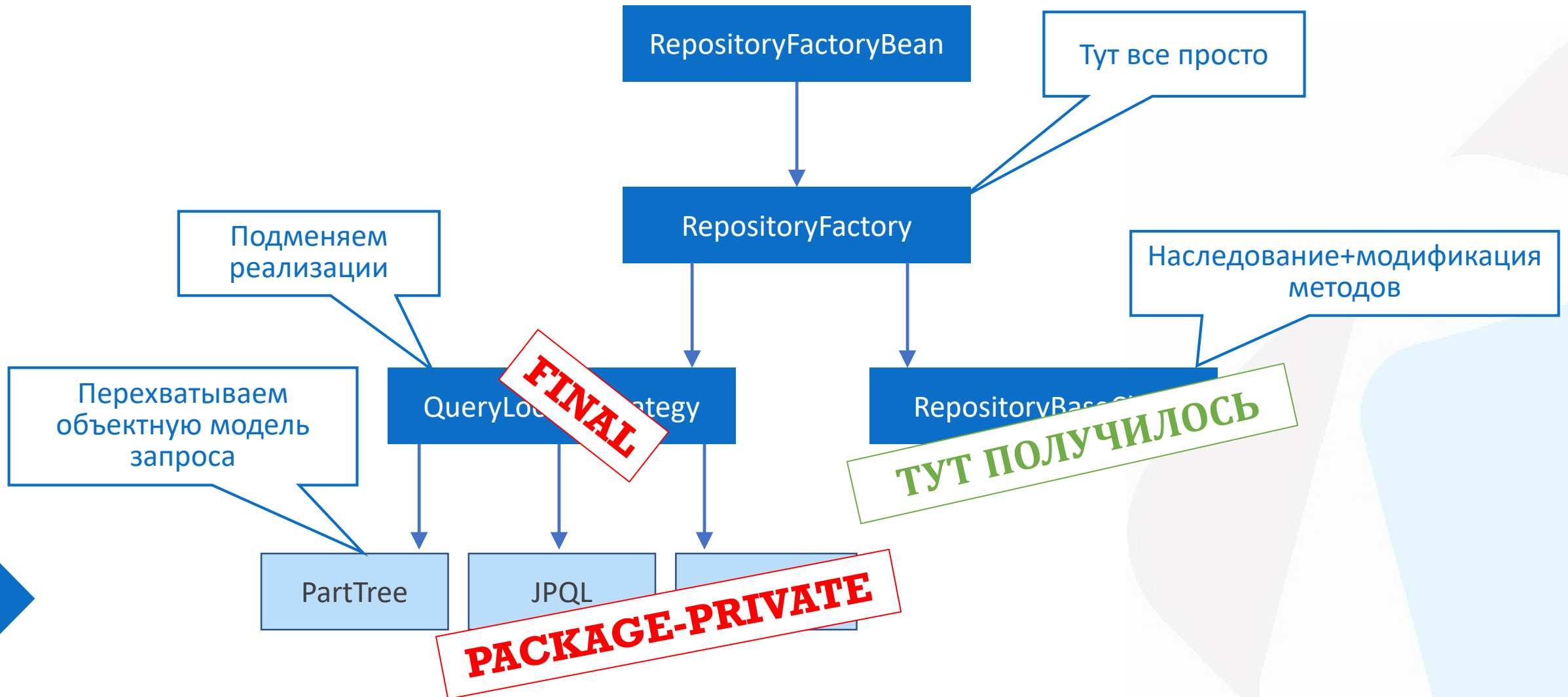
@Override

```
public void deleteById(ID id) {  
    if (softDeleteEnabled) {  
        T entity = em.find(getDomainClass(), id);  
        markAsDeleted(entity);  
        em.merge(entity);  
    } else {  
        super.deleteById(id);  
    }  
}
```

...

```
public interface VetRepository extends  
SoftDeleteRepository<Vet, Integer> { . . . }
```

Ожидание и реальность



Объектная модель запроса

```
public interface RepositoryQuery {  
  
    /**  
     * Executes the {@link RepositoryQuery} with the given parameters.  
     *  
     * @param parameters must not be {@literal null}.  
     * @return execution result. Can be {@literal null}.  
     */  
    @Nullable  
    Object execute(Object[] parameters);  
  
    /**  
     * Returns the related {@link QueryMethod}.  
     *  
     * @return never {@literal null}.  
     */  
    QueryMethod getQueryMethod();  
}
```

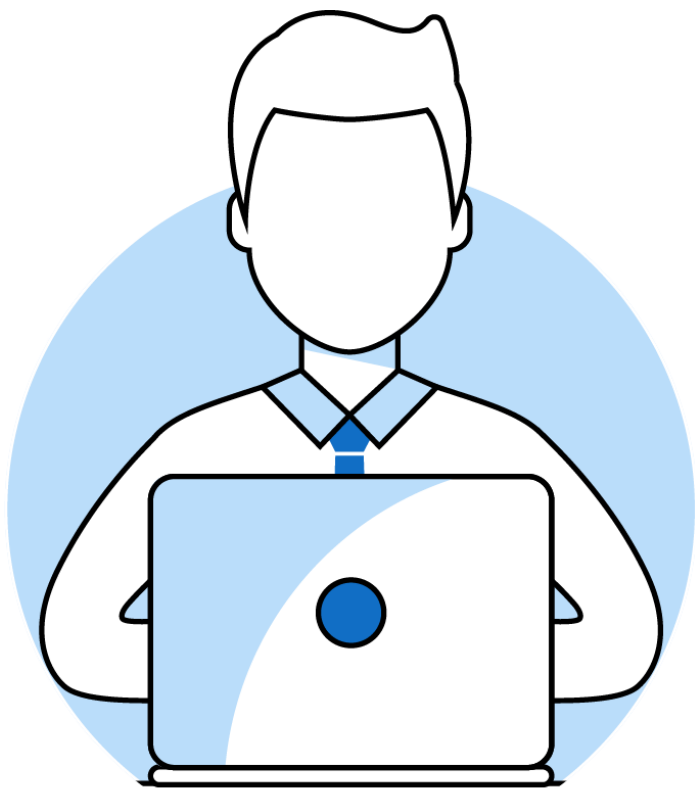


Аналогичные попытки

<https://github.com/lordlothar99/strategy-spring-security-acl>

<https://github.com/BerryCloud/spring-data-jpa-acl>

- Примерно те же проблемы
- Там тоже копипастят код



Спасибо за внимание!

И это мы ещё не затрагивали
CASCADE...