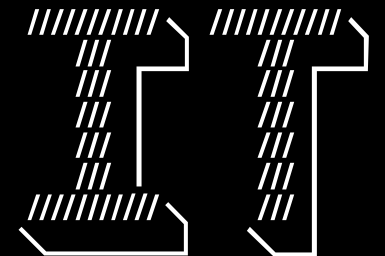


Паттерн «Спецификация»: назначение и применение

Райффайзен



Обо мне



Владислав Шакиров

Разработчик

Разработчик в Райффайзенбанк. Разрабатываю на .Net уже около 5 лет. Занимаюсь разработкой Backend и Desktop-приложений, а так же стараюсь делать архитектуру приложений проще и понятнее 😊

План доклада



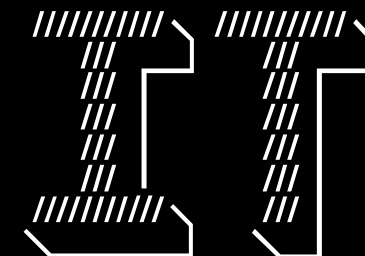
- Определение паттерна
- Применение
- Вариант реализации

Сначала было...

```
public class Member
{
    3 references
    public string Nick { get; set; }
    4 references
    public string Country { get; set; }
    1 reference
    public string Planet { get; set; }
    1 reference
    public string Address { get; set; }
    1 reference
    public decimal Money { get; set; }
    1 reference
    public Gender Gender { get; set; }
    3 references
    public int Age { get; set; }
}
```

```
if (member.Nick == "Ivan"
    && member.Country == "Russia"
    && member.Planet == "Earth"
    && member.Gender == Gender.Male
    && member.Address == "PrivateStreet"
    && member.Age == 42
    && member.Money == 1000000000)
{
    DoSomething();
}
```

Райффайзен



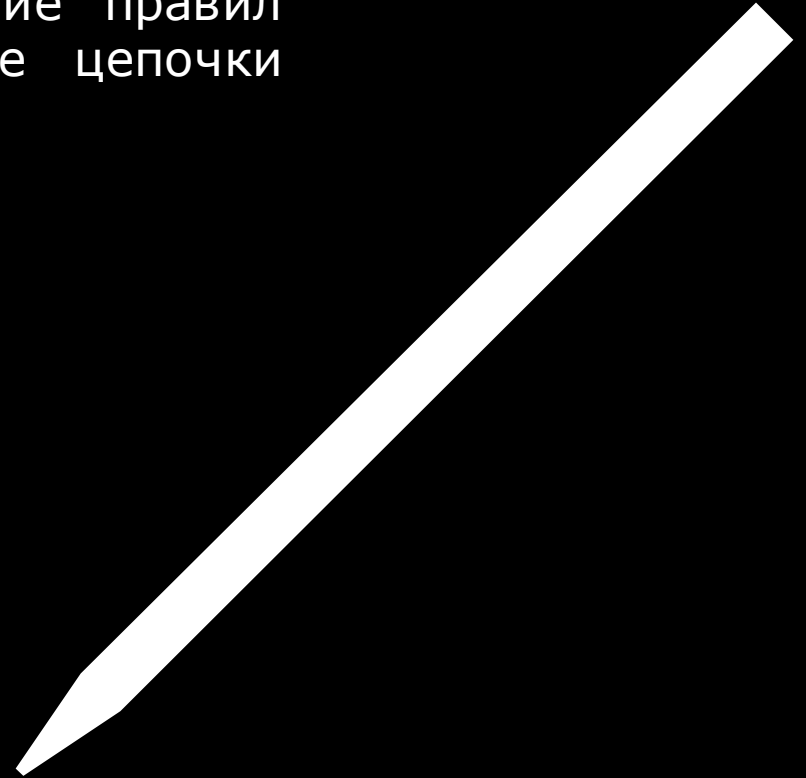
Сначала было...



1 Определение паттерна

«**Спецификация**» в программировании — это шаблон проектирования, посредством которого представление правил бизнес логики может быть преобразовано в виде цепочки объектов, связанных операциями булевой логики.

```
interface ISpecification<T>
{
  7 references | 0 changes | 0 authors, 0 changes
  bool IsSatisfiedBy(T entity);
}
```



Применение



- Фильтрация объектов

Применение



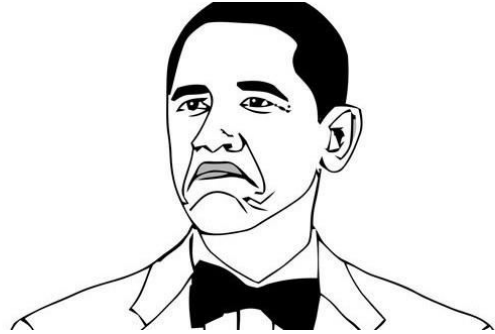
- Фильтрация объектов
- Составление универсальных запросов к репозиторию

Применение



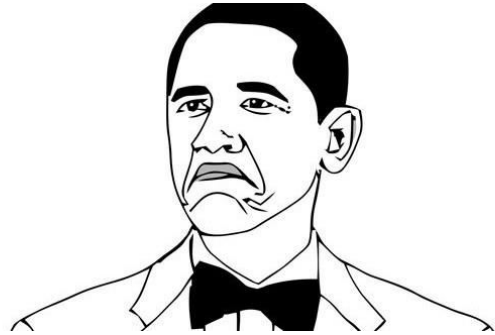
- Фильтрация объектов
- Составление универсальных запросов к репозиторию
- Уменьшение дублирования кода

2 Применения: особенности



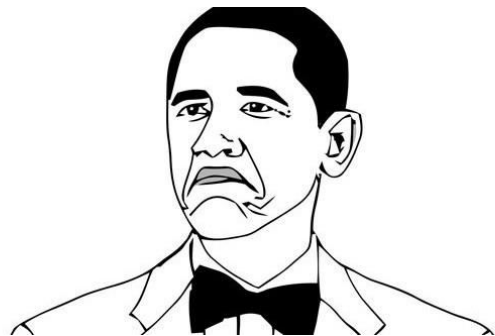
- Чистый репозиторий

2 Применения: особенности



- Чистый репозиторий
- Возможность переиспользовать спецификации

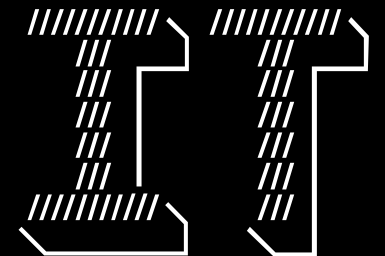
2 Применения: особенности



- Чистый репозиторий
- Возможность переиспользовать спецификации
- В случае изменения бизнес требования, мы можем изменить запрос в одном месте, не меняя кучу кода

3 Реализация

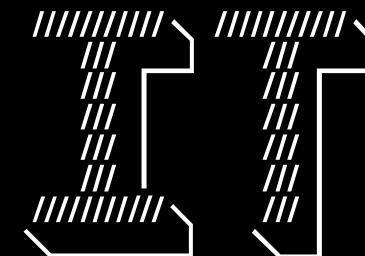
Райффайзен



3 Реализация



Райффайзен



3 Реализация



- Плохая:

Построение запросов
поверх репозитория:

```
_membersRepository.Query.Where(m => m.Age == 31 && m.Country == "Russia");
```

3 Реализация



- Плохая:

Построение запросов
поверх репозитория:

- Нормальная:

Метод в контракте
репозитория

```
var membersFromRussia = _membersRepository.GetMembersFromRussia();
```


3 Реализация



- Плохая:

Построение запросов
поверх репозитория:

- Нормальная:

Метод в контракте
репозитория

- Отличная

Спецификация

```
var membersFromRussia = _membersRepository.Query.Where(memberFromRussiaSpecification.ToExpression());
```

3 Реализация: пример

1 reference | 0 changes | 0 authors, 0 changes

```
interface ISpecification<T>
```

```
{
```

7 references | 0 changes | 0 authors, 0 changes

```
bool IsSatisfiedBy(T entity);
```

```
}
```

```
public abstract class Specification<T> : ISpecification<T>
```

```
{
```

3 references

```
public abstract Expression<Func<T, bool>> ToExpression();
```

```
protected Expression<Func<T, bool>> Expression = null;
```

2 references

```
public bool IsSatisfiedBy(T entity)
```

```
{
```

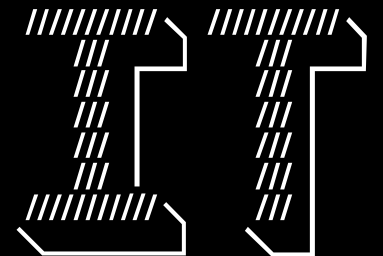
```
    Func<T, bool> predicate = ToExpression().Compile();
```

```
    return predicate(entity);
```

```
}
```

```
}
```

Райффайзен



3 Реализация: пример

```
public class MemberFromRussiaSpecification : Specification<Member>
{
    1 reference
    public MemberFromRussiaSpecification()
    {
        Expression = member => member.Country == "Russia";
    }

    2 references
    public override Expression<Func<Member, bool>> ToExpression()
    {
        //return member => member.Country == "Russia";
        return Expression;
    }
}
```

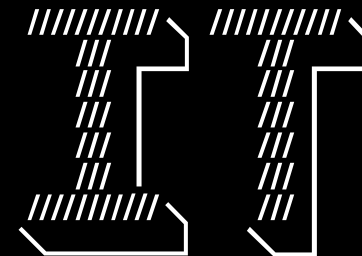
Райффайзен

```
MemberFromRussiaSpecification memberFromRussiaSpecification = new MemberFromRussiaSpecification();
var membersFromRussia = _membersRepository.Query.Where(memberFromRussiaSpecification.ToExpression());
```

3 Реализация: пример

```
Member memberIvan = new Member { Nick = "Ivan", Country = "Russia", Age = 42 };  
  
bool fromRussia = memberFromRussiaSpecification.IsSatisfiedBy(memberIvan);
```

Райффайзен



3 Реализация: пример

One more thing

```
public class AndSpecification<T> : Specification<T>
{
    private readonly Specification<T> _spec1;
    private readonly Specification<T> _spec2;

    1 reference
    internal AndSpecification(Specification<T> spec1, Specification<T> spec2)
    {
        _spec1 = spec1;
        _spec2 = spec2;

        expression = Expression.Lambda<Func<T, bool>>
            (Expression.And(_spec1.ToExpression().Body,
                Expression.Invoke(_spec2.ToExpression(), _spec1.ToExpression().Parameters)),
            _spec1.ToExpression().Parameters);
    }

    9 references
    public override Expression<Func<T, bool>> ToExpression()
    {
        return expression;
    }
}
```

измен



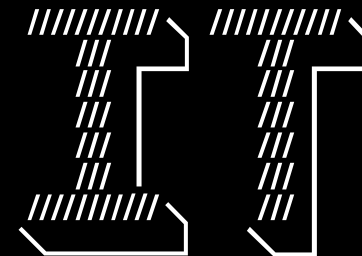
3 Реализация: пример

One more thing

```
MemberFromRussiaSpecifiction memberFromRussiaSpecifiction = new MemberFromRussiaSpecifiction();  
MemberIvanSpecifiction ivanSpecifiction = new MemberIvanSpecifiction();  
  
AndSpecifiction<Member> ivanFromRussia = new AndSpecifiction<Member>(memberFromRussiaSpecifiction,  
                                                                           ivanSpecifiction);
```

```
var membersFromRussia = _membersRepository.Query.Where(ivanFromRussia.ToExpression());  
  
if (ivanFromRussia.IsSatisfiedBy(memberIvan))  
{  
    DoSomething();  
}
```

Райффайзен





Владислав Шакиров

Разработчик

shakirov.vr@yandex.ru
vk.com/vr.shakirov



"That's all Folks!"