

# Теория категорий для описания архитектуры программ: проектирование и документирование

Viete

Станислав Капулкин

Владислав Цендровский

# Введение

Теория категорий - удобный математически строгий язык схем, и мы увидим, как эти схемы удобно применять в программировании.

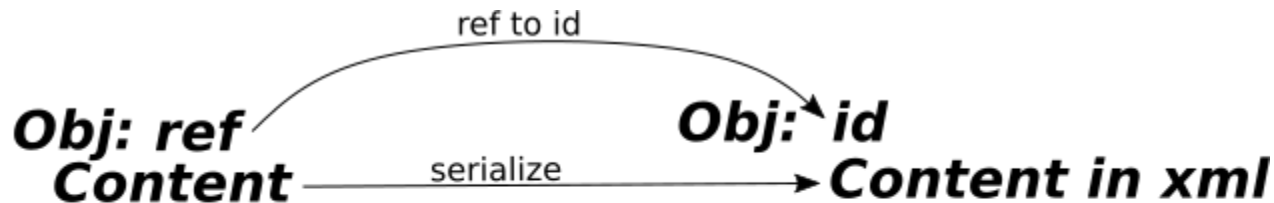
# Пример - сохранение/загрузка из файла



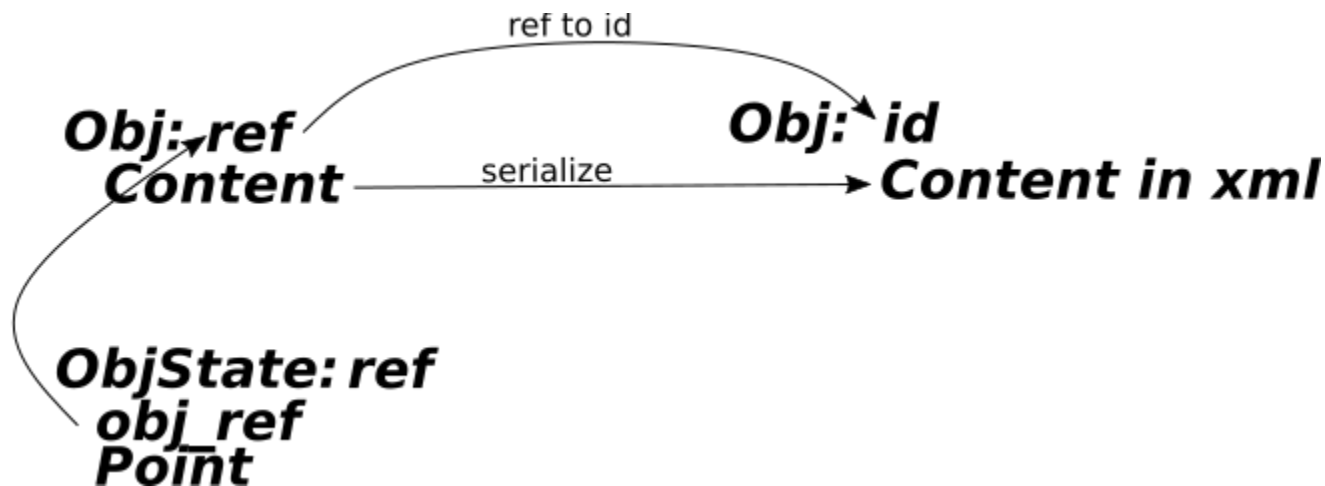
# Пример - сохранение/загрузка из файла



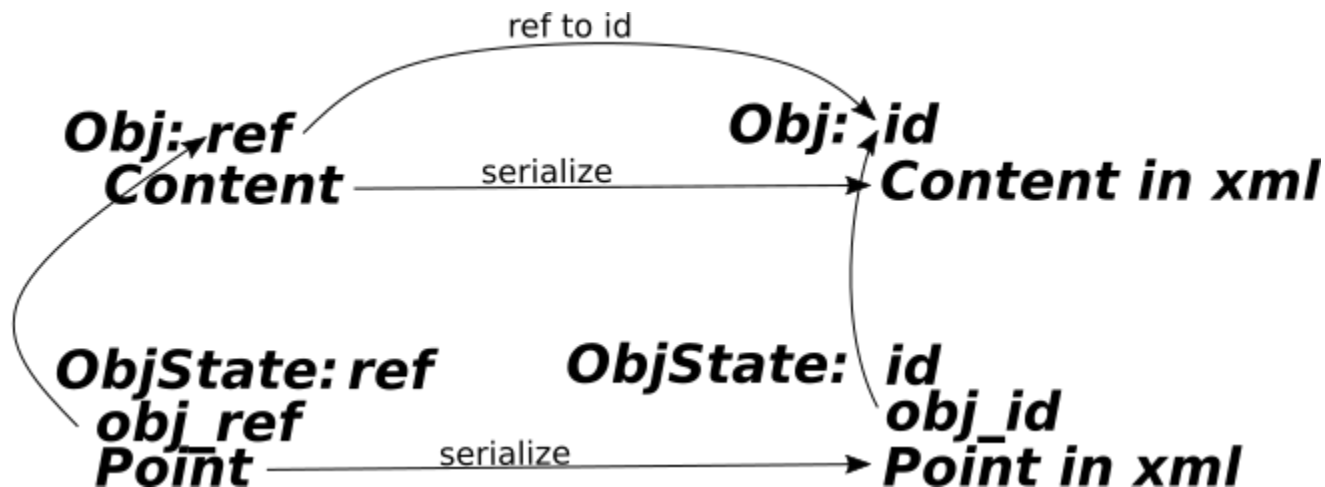
## Пример - сохранение/загрузка из файла



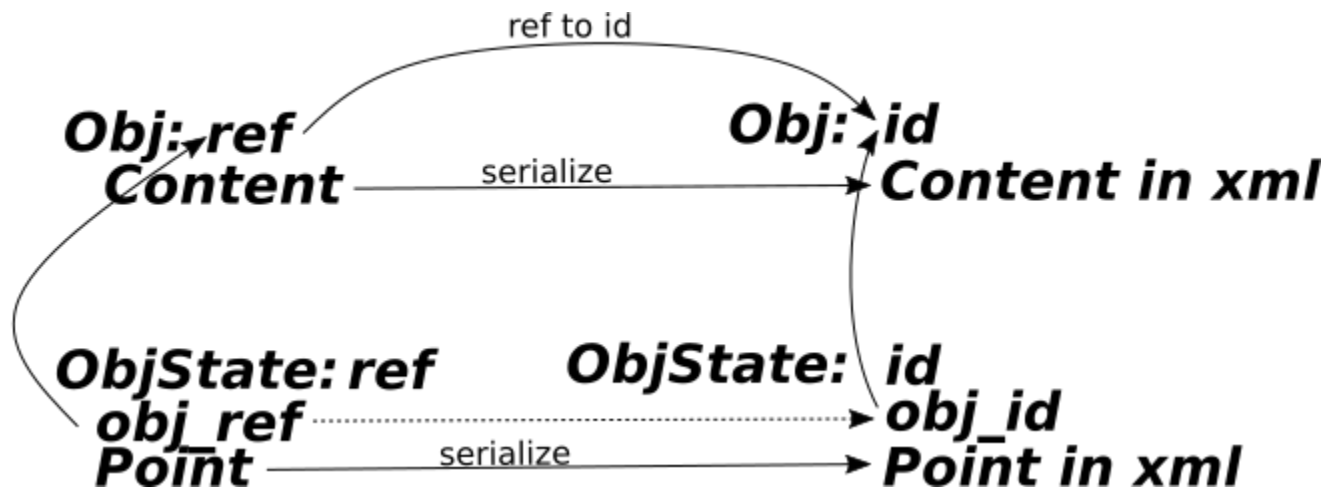
## Пример - сохранение/загрузка из файла



# Пример - сохранение/загрузка из файла

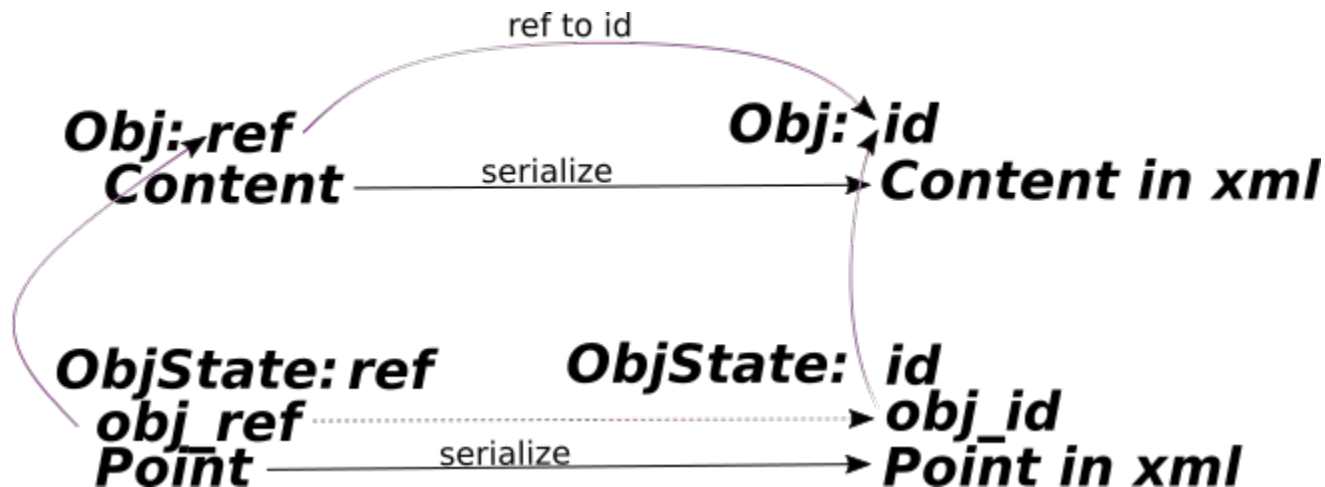


## Пример - сохранение/загрузка из файла

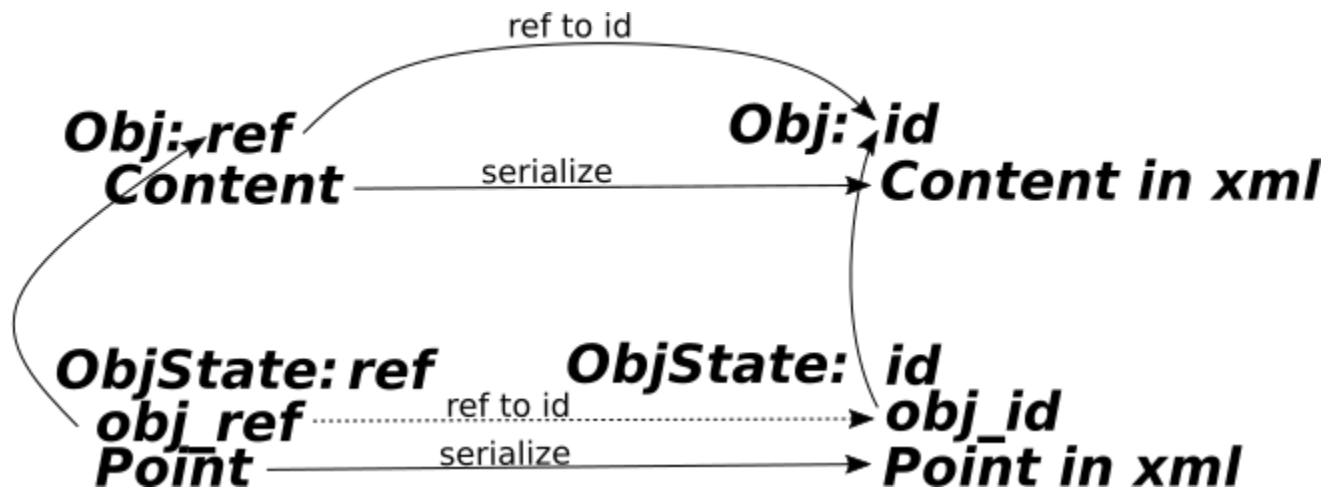




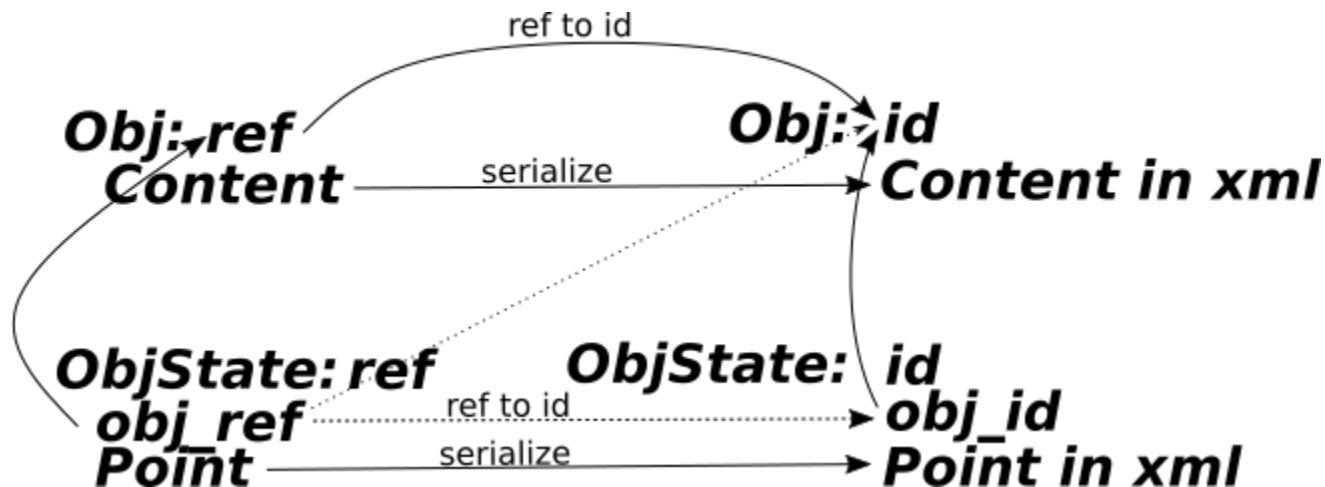
# Пример - сохранение/загрузка из файла



## Пример - сохранение/загрузка из файла



## Пример - сохранение/загрузка из файла



# Элементы теории категории

Объекты

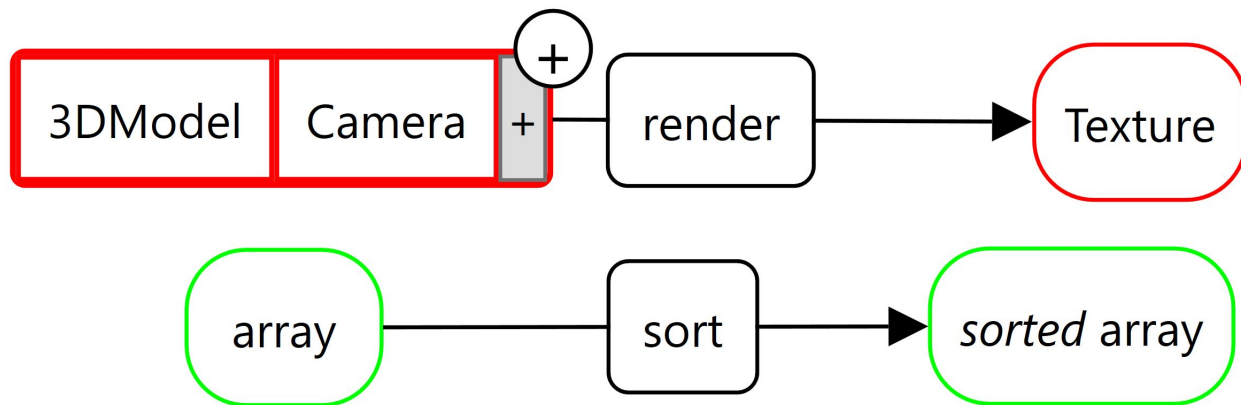
String

**Event**



# Элементы теории категории (продолжение)

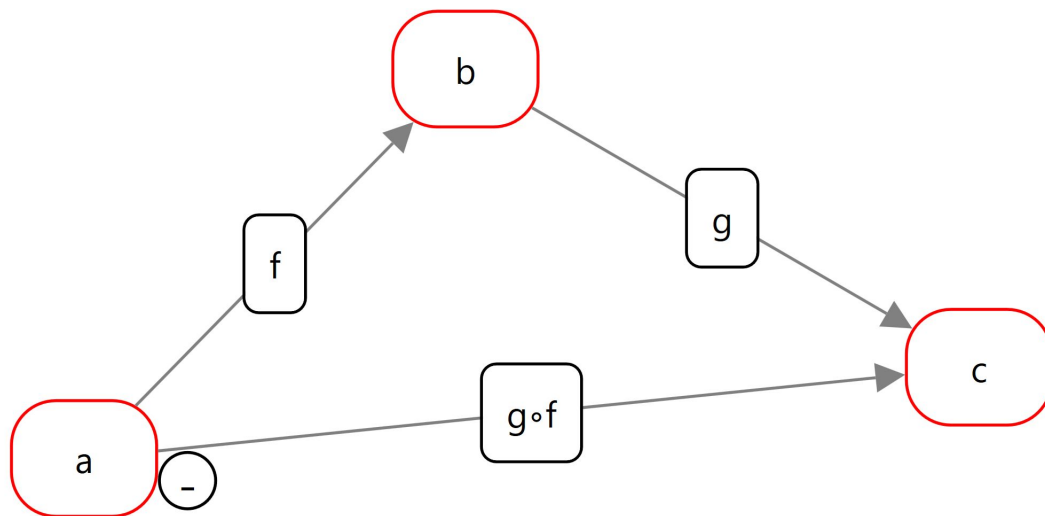
Стрелки - Морфизмы



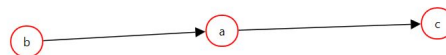
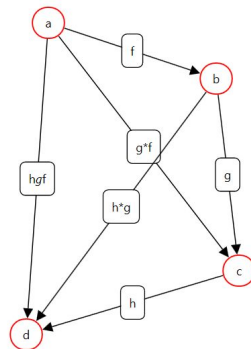
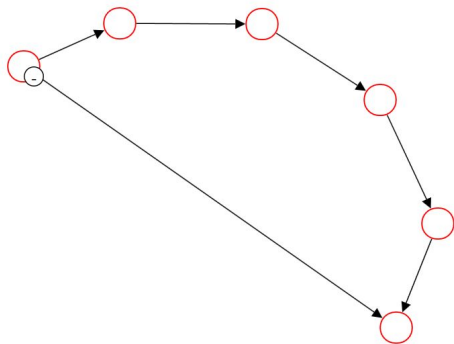
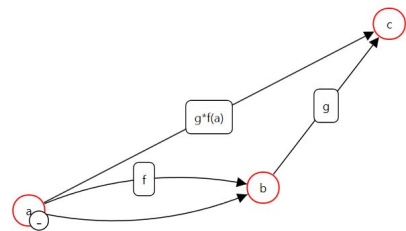
# Элементы теории категории (продолжение)

Равенство путей -

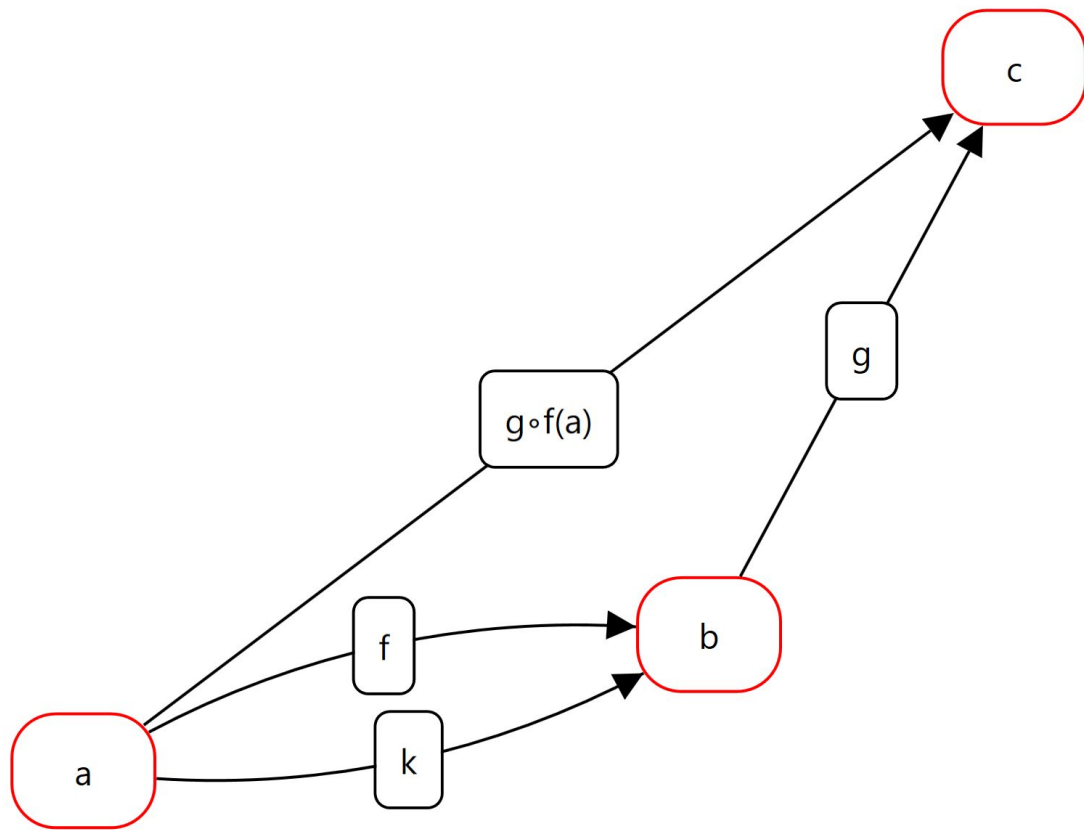
Коммутативная диаграмма



# Категория, аксиомы



# Композиция





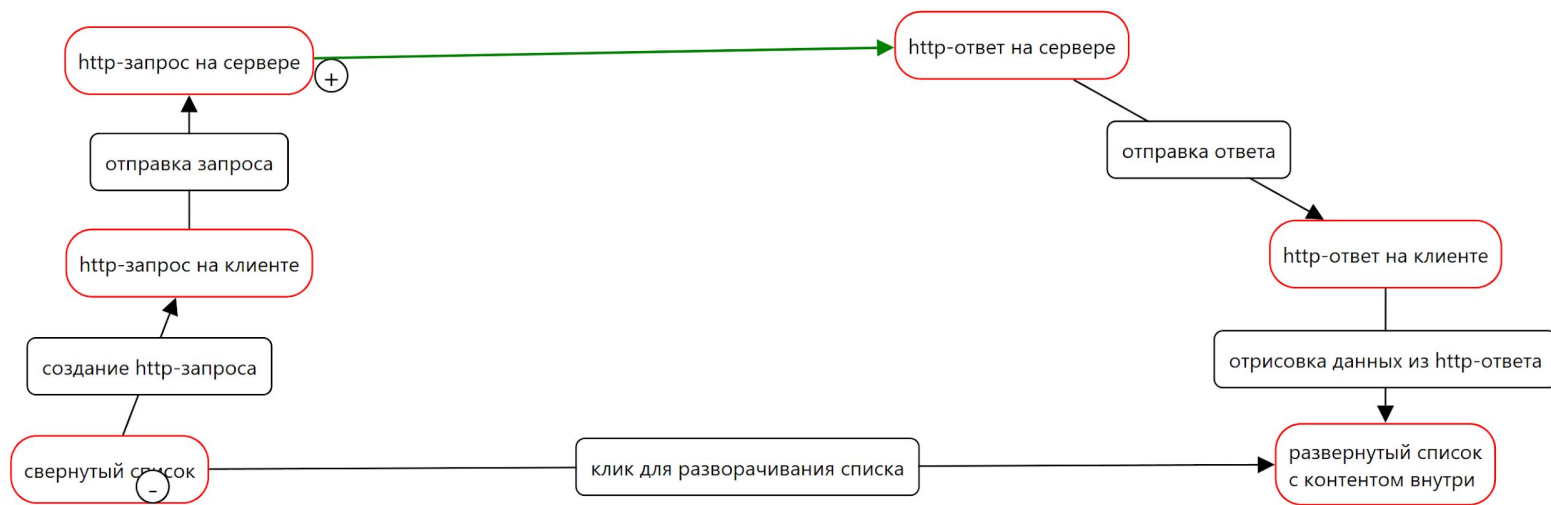
# Коммутативная диаграмма



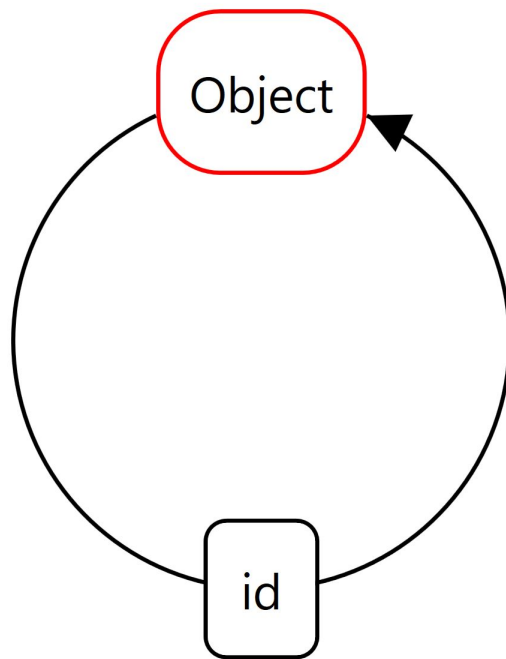
# Больше коммутативных диаграмм



# Меньше коммутативных диаграмм

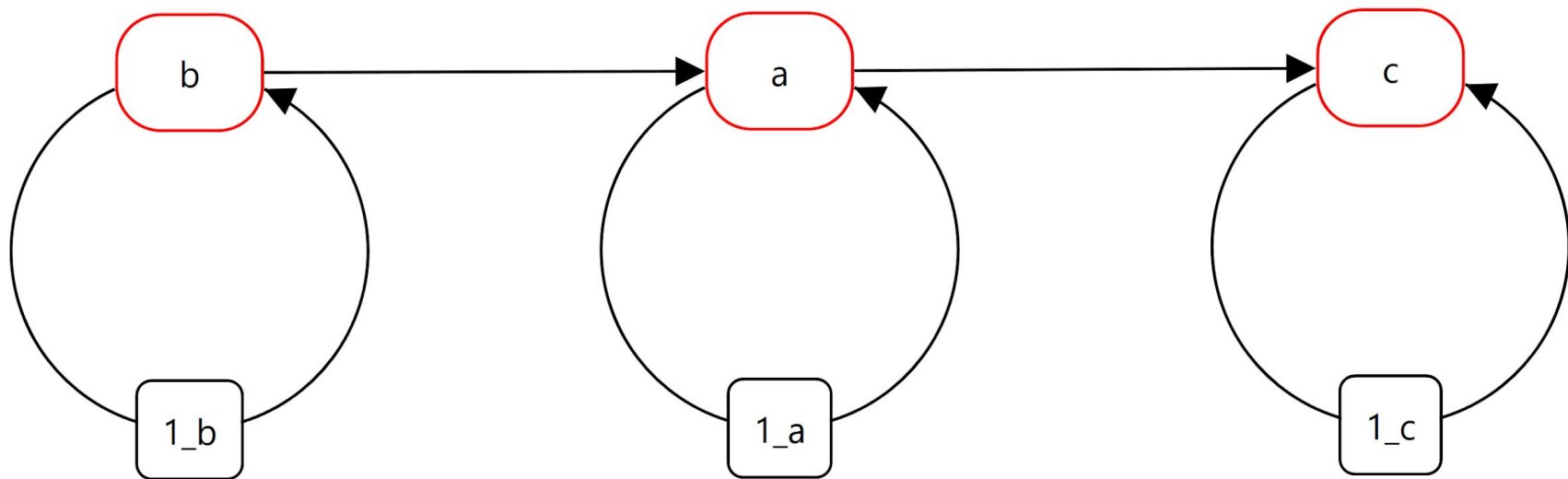


# Единичный морфизм

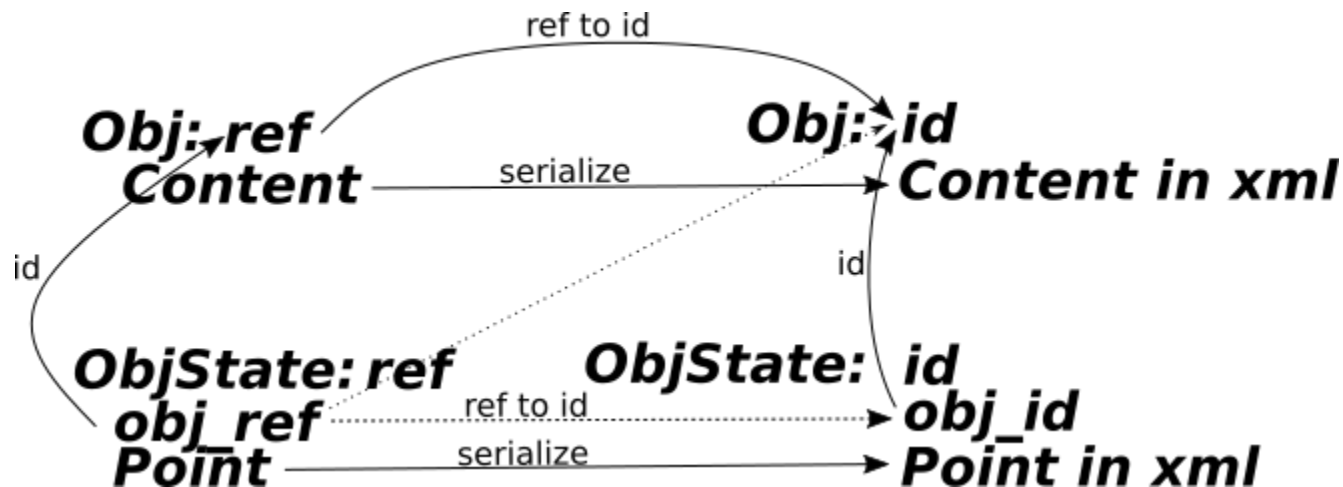


# Единичный морфизм

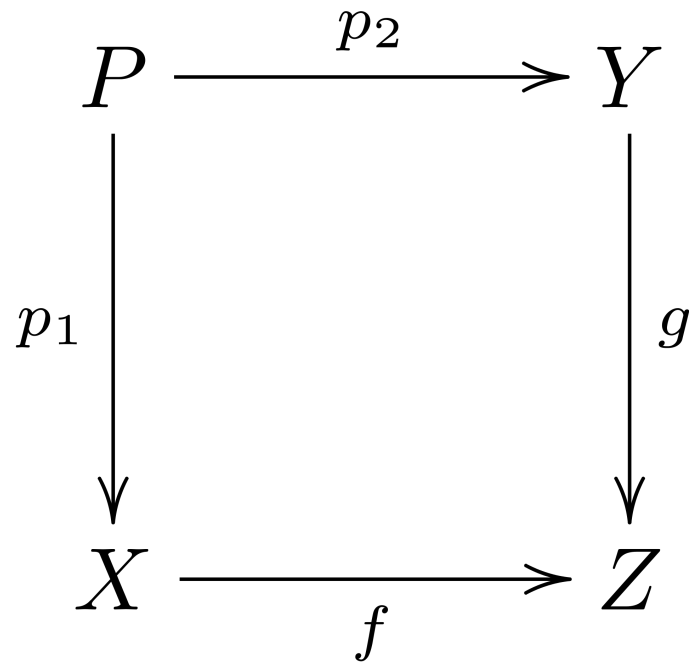
композиция с единичным морфизмом



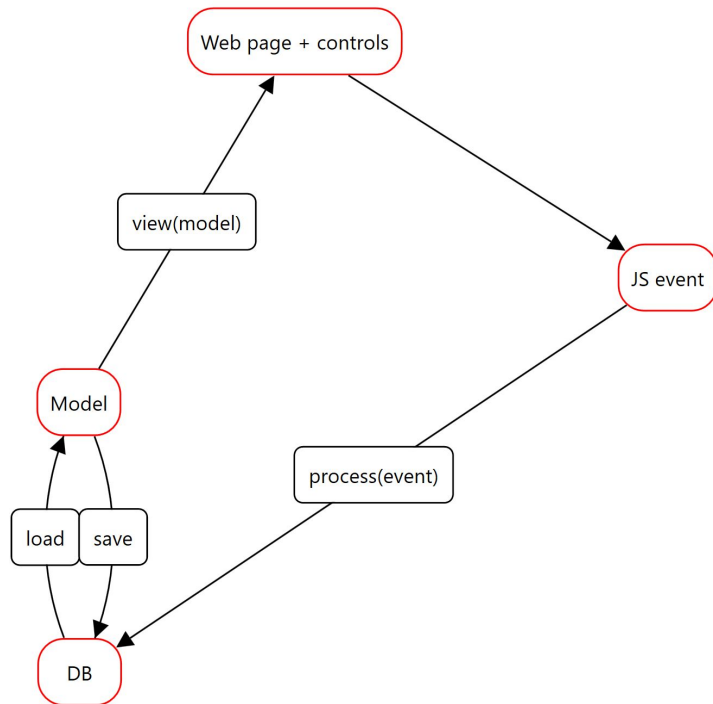
## Пример - сохранение/загрузка из файла



# Pull-back

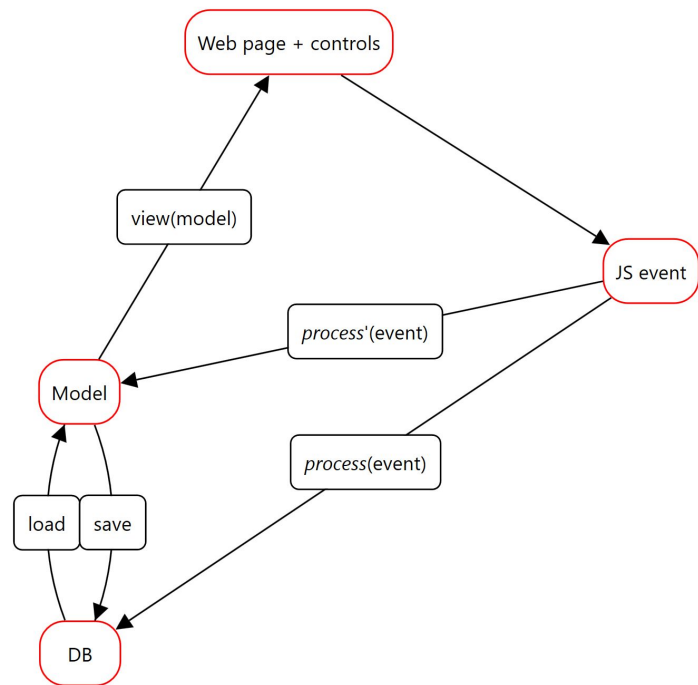


# Пример - MVC

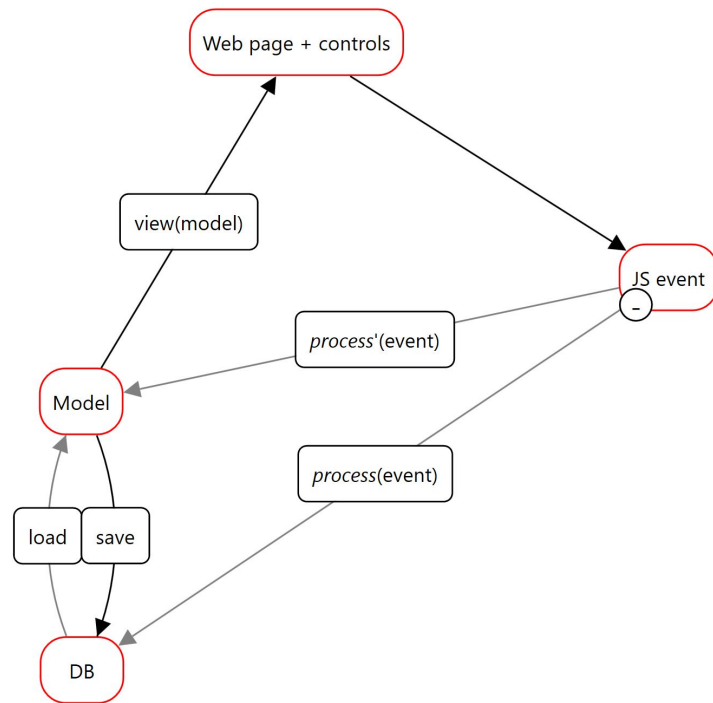




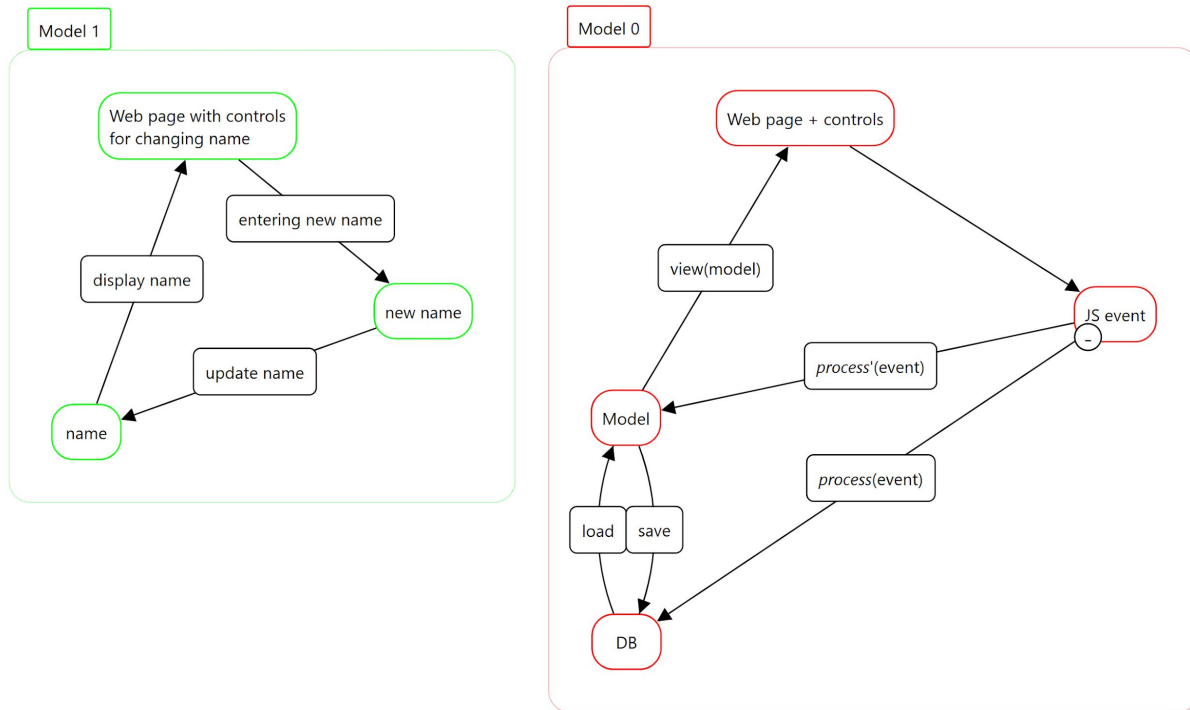
# Пример - MVC, второй сценарий



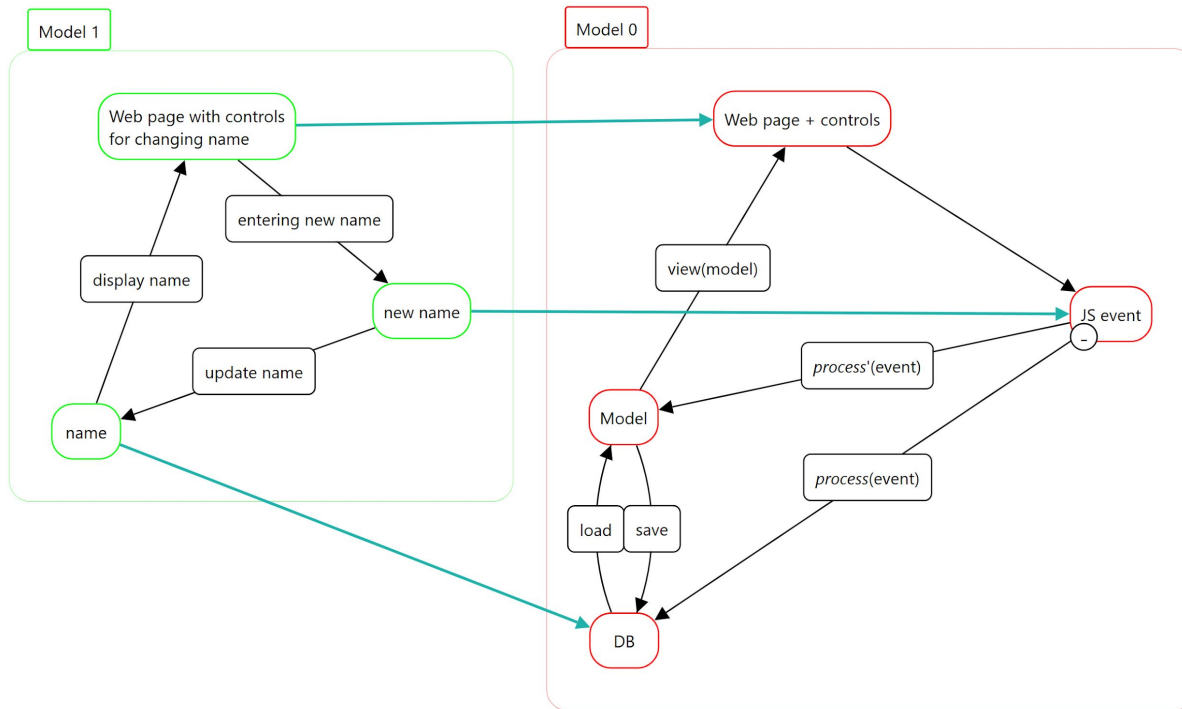
# Пример - MVC, unit-тест



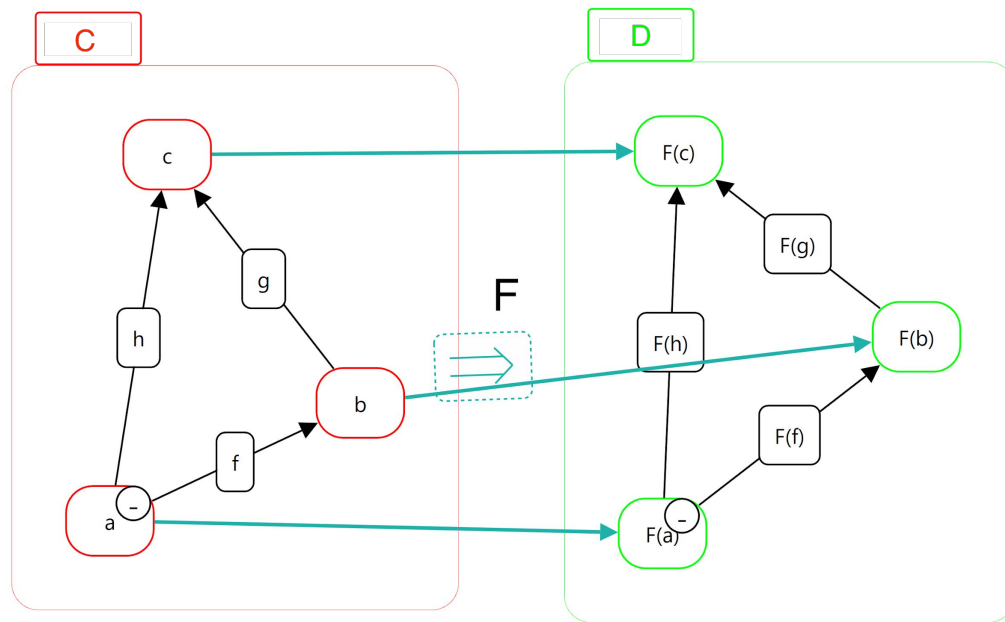
# Use-case от product owner-а: сохранение имени



# Use-case от product owner-a: сохранение имени

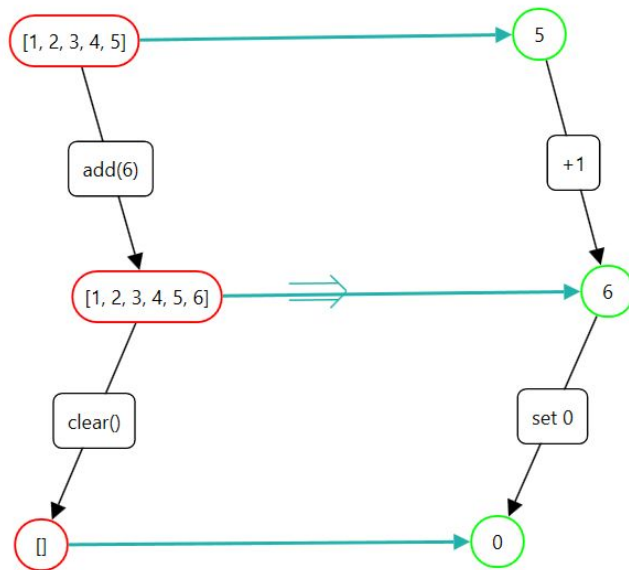


# Определение функтора

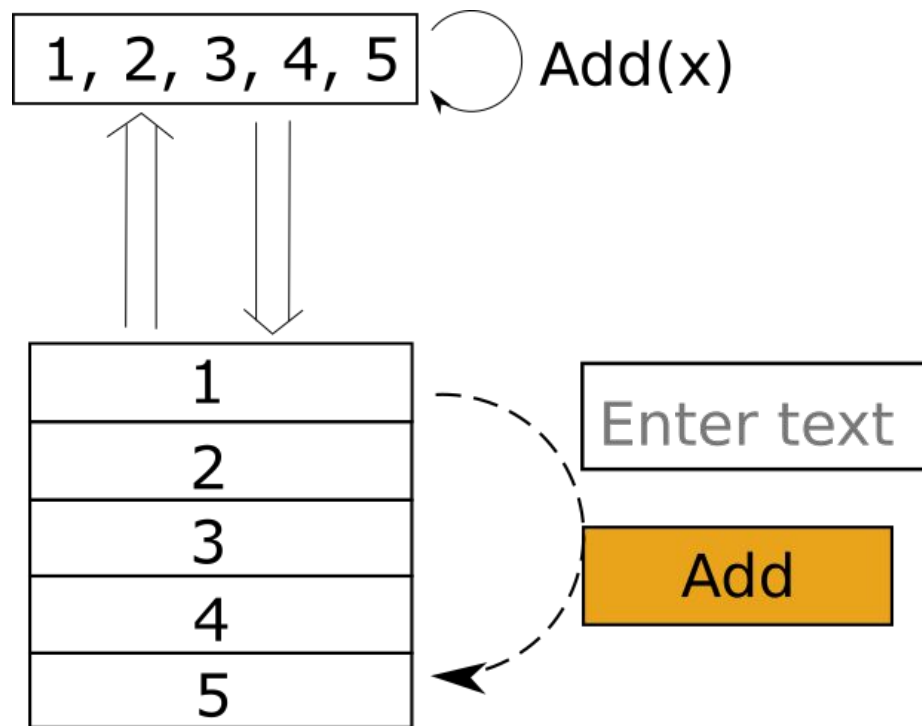


$$F(h) = F(g) \circ F(f)$$

# Пример - Size

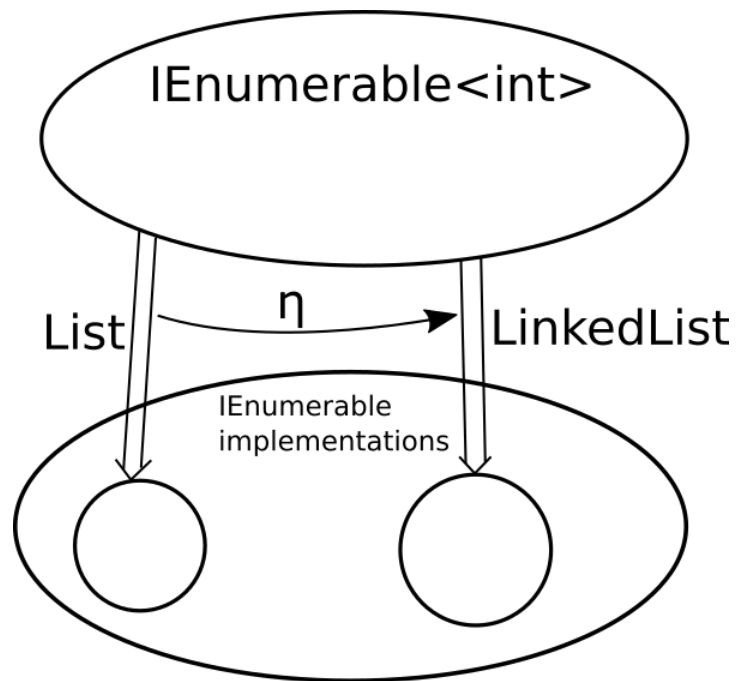


# Пример - MVVM



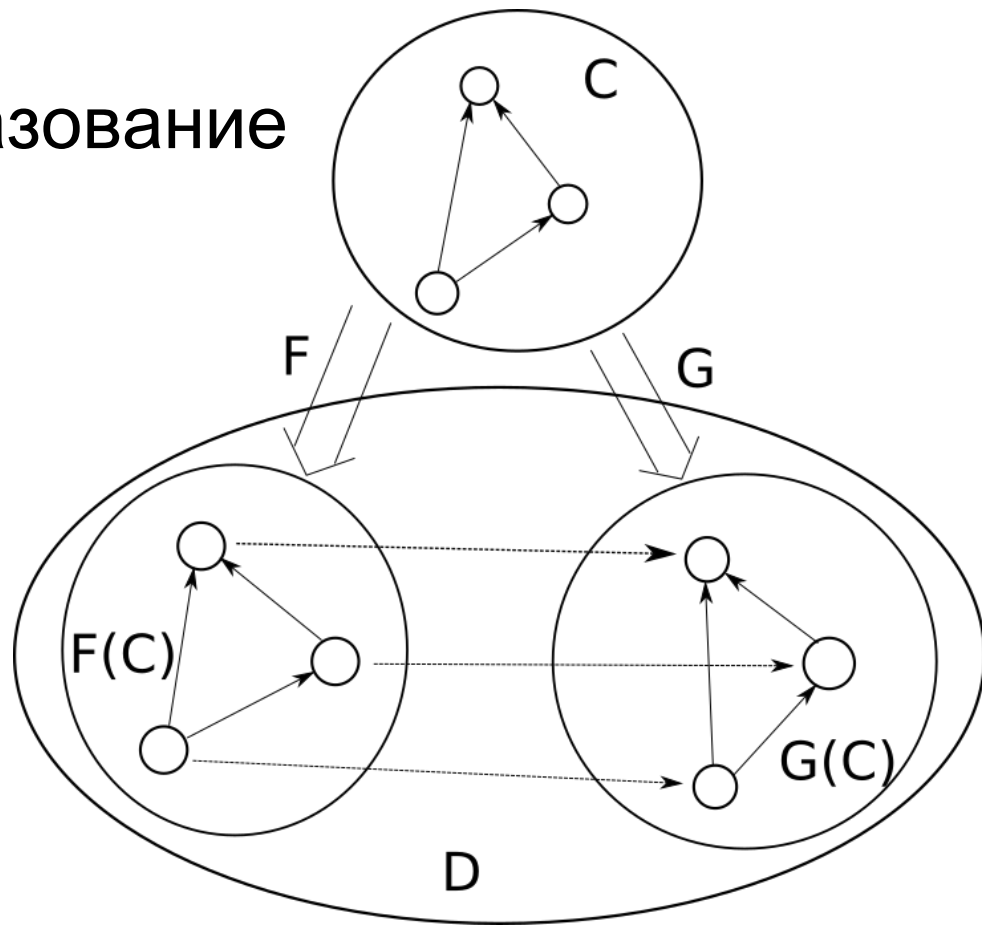
# Теоркат и ООП

```
var list = new List<int>(new int [] {1, 2, 3, 4, 5});  
  
// new LinkedList<int>(new int [] {1, 2, 3, 4, 5});  
  
foreach (var value in list)  
{  
    Console.WriteLine(value);  
}
```



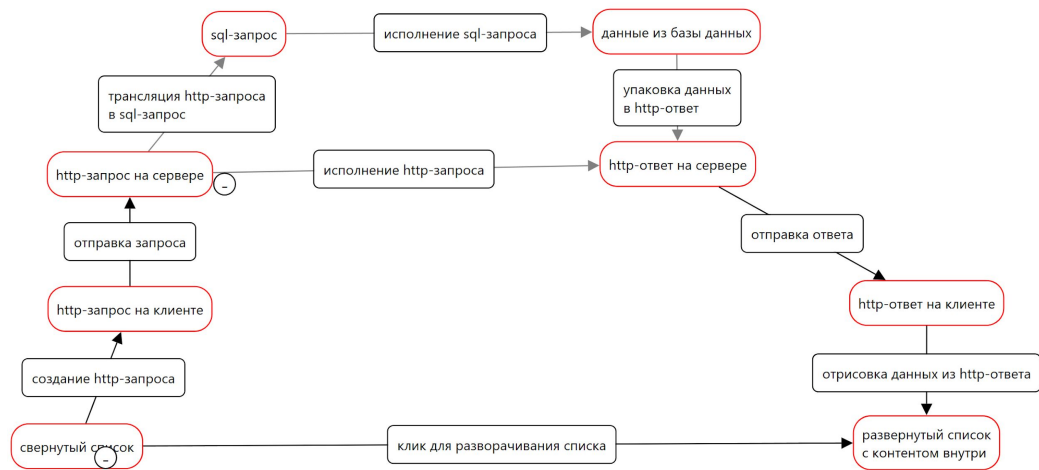
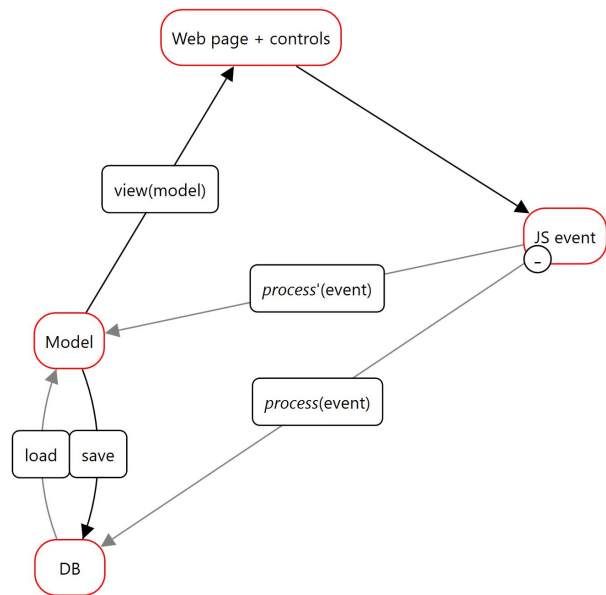


# Естественное преобразование

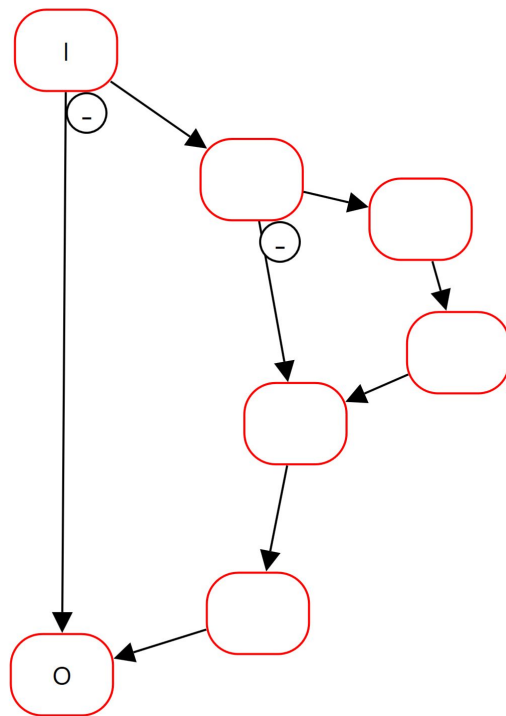


# Подход к проектированию

выбираем в диаграмме стрелку и расписываем ее детальнее

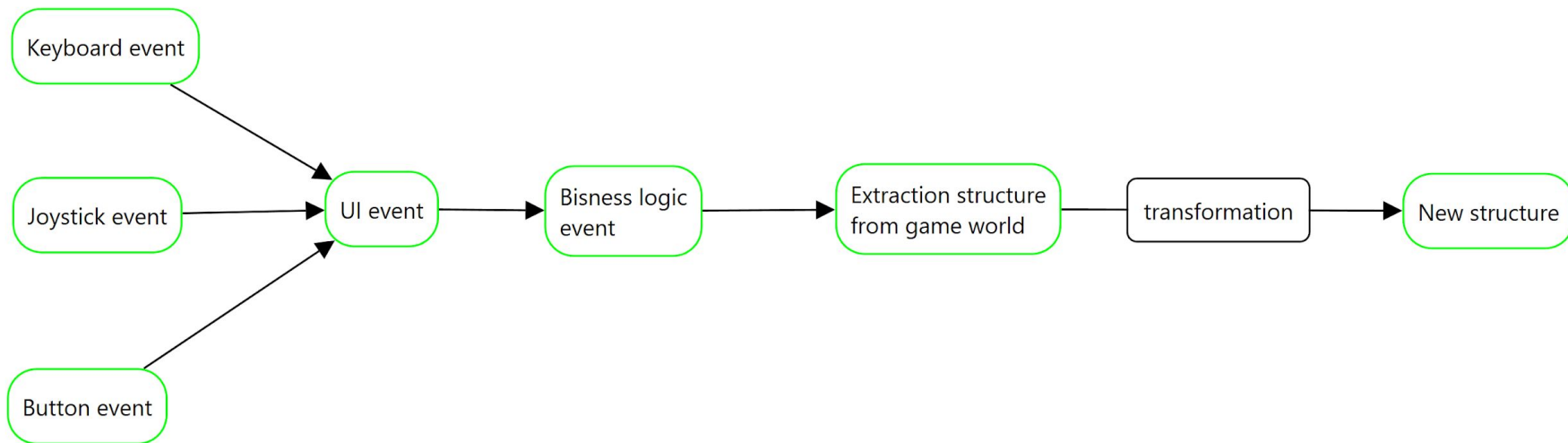


# Подход к проектированию



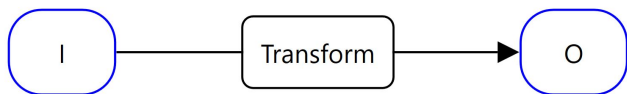
# Подход к проектированию (пример)

Обработка нажатия кнопки в игре



# Выделение классов по цепочке шагов

стрелка - функция или метод

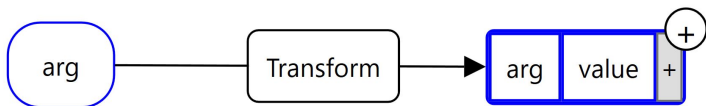


```
void Transform(ref int arg);
```

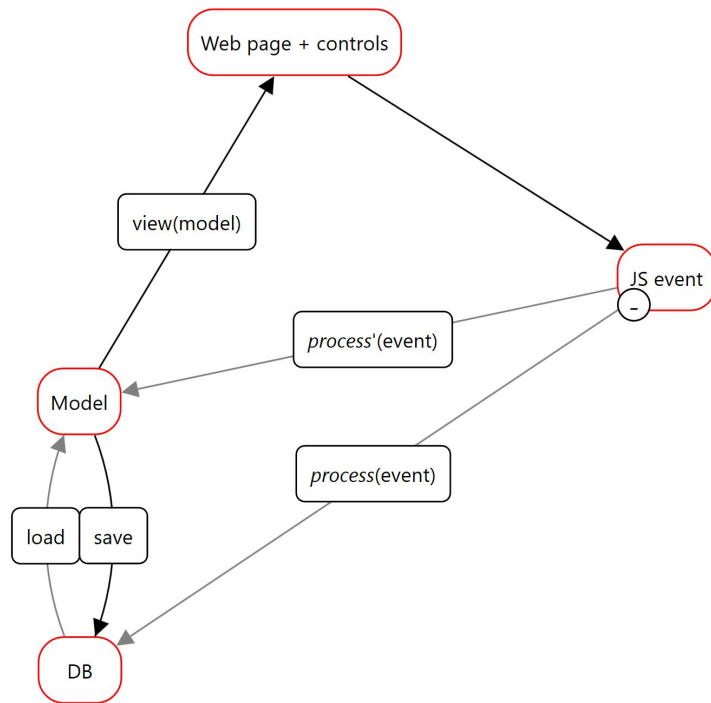
```
void Transform() { this.field = Math.Sin(field); }
```



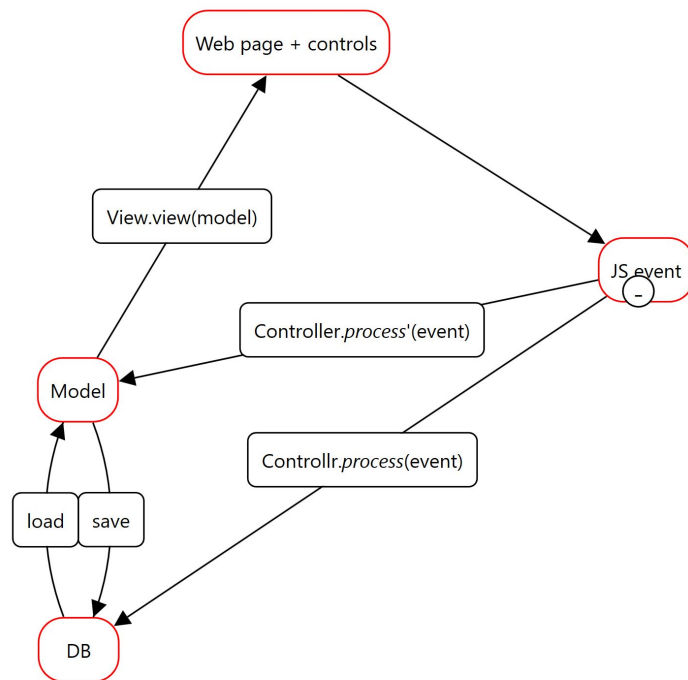
```
int Transform(int arg);
```



# Кодогенерация архитектуры классов

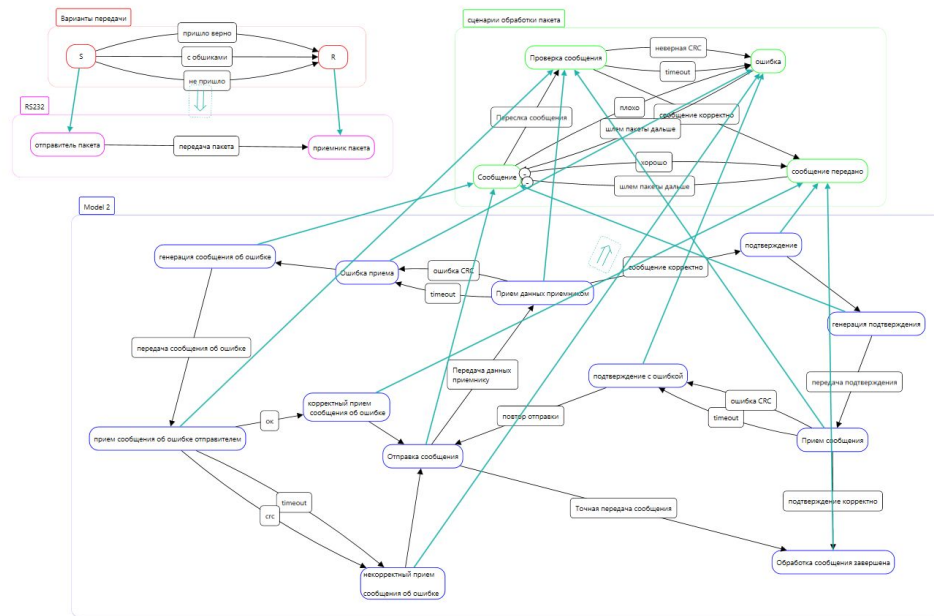


# Кодогенерация архитектуры классов



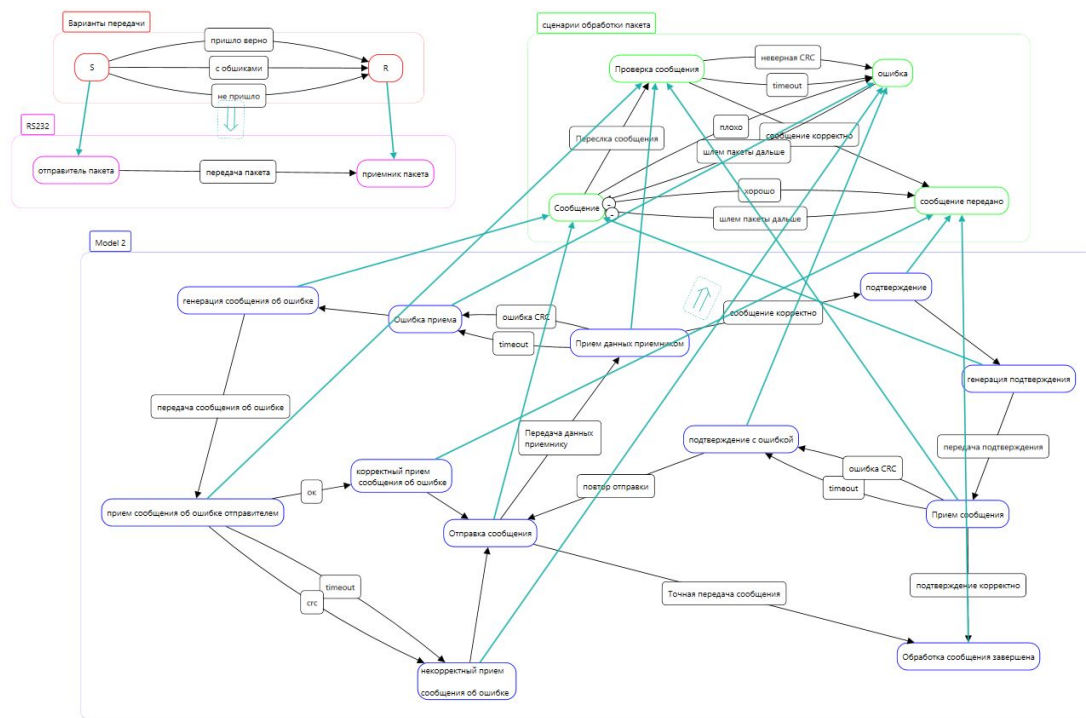
# Послесловие

1. CAP-теорема. Хорошо расписав систему мы четко поймем, чем мы жертвуем
2. Пример с протоколом связи





# Пример с протоколом связи



# Спасибо за внимание!

Вопросы?

[t.me/catdotnet](https://t.me/catdotnet)

[dev.viete.io](https://dev.viete.io)

Литература

1. ["Розеттский камень", Баез Дж. К., Стэй М.](#)
2. ["Теория категорий для работающего математика", С. Маклейн](#)