

СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
SIBERIAN FEDERAL UNIVERSITY

Институт космических и информационных технологий
Объектно-ориентированное программирование

Слабое связывание и высокое зацепление

к.т.н. Якунин Юрий Юрьевич

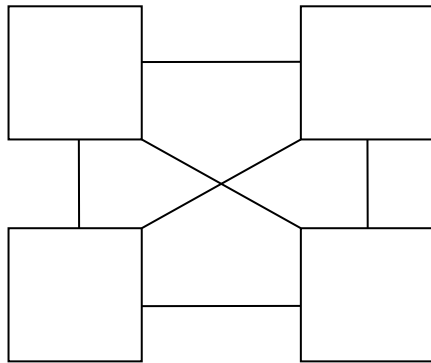
Шаблон Low Coupling

Слабое связывание

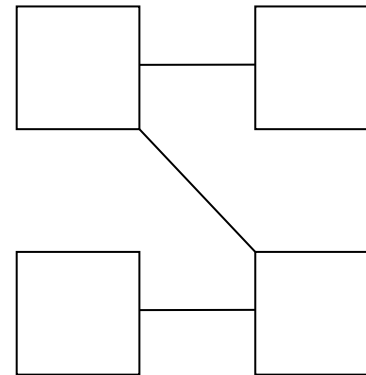
- **Проблема.** Как обеспечить незначительное влияние изменений и повысить возможность повторного использования?
- **Решение.** Распределить обязанности таким образом, чтобы степень связанности оставалась низкой

Степень связанности (coupling)

- **Степень связанности (coupling)** – это мера, определяющая насколько жестко один элемент связан с другими элементами, либо каким количеством данных о других элементах он обладает



**Высокая степень
связанности**

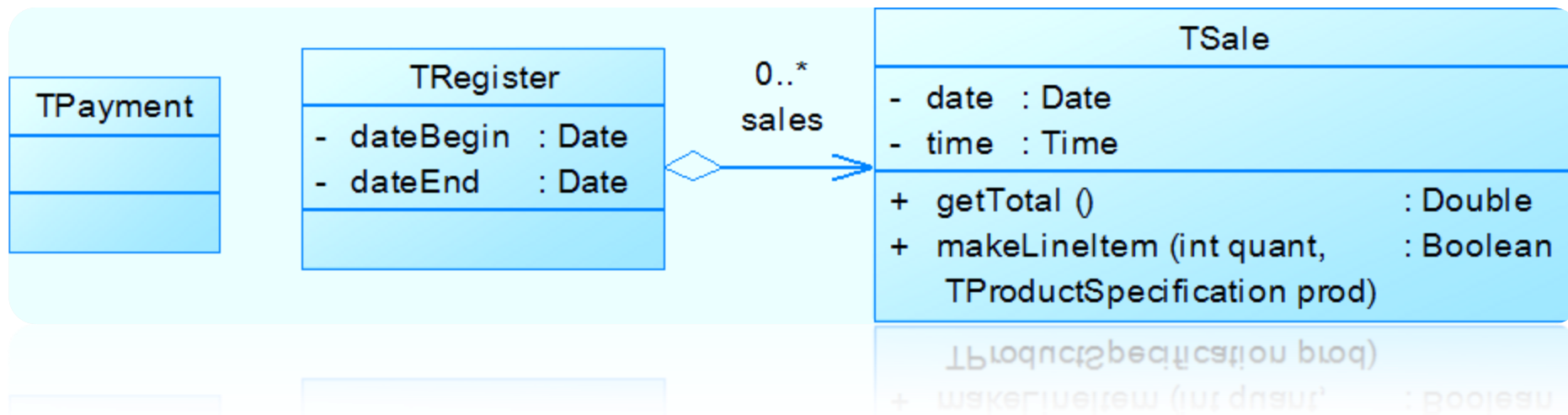


**Низкая степень
связанности**

Шаблон Low Coupling

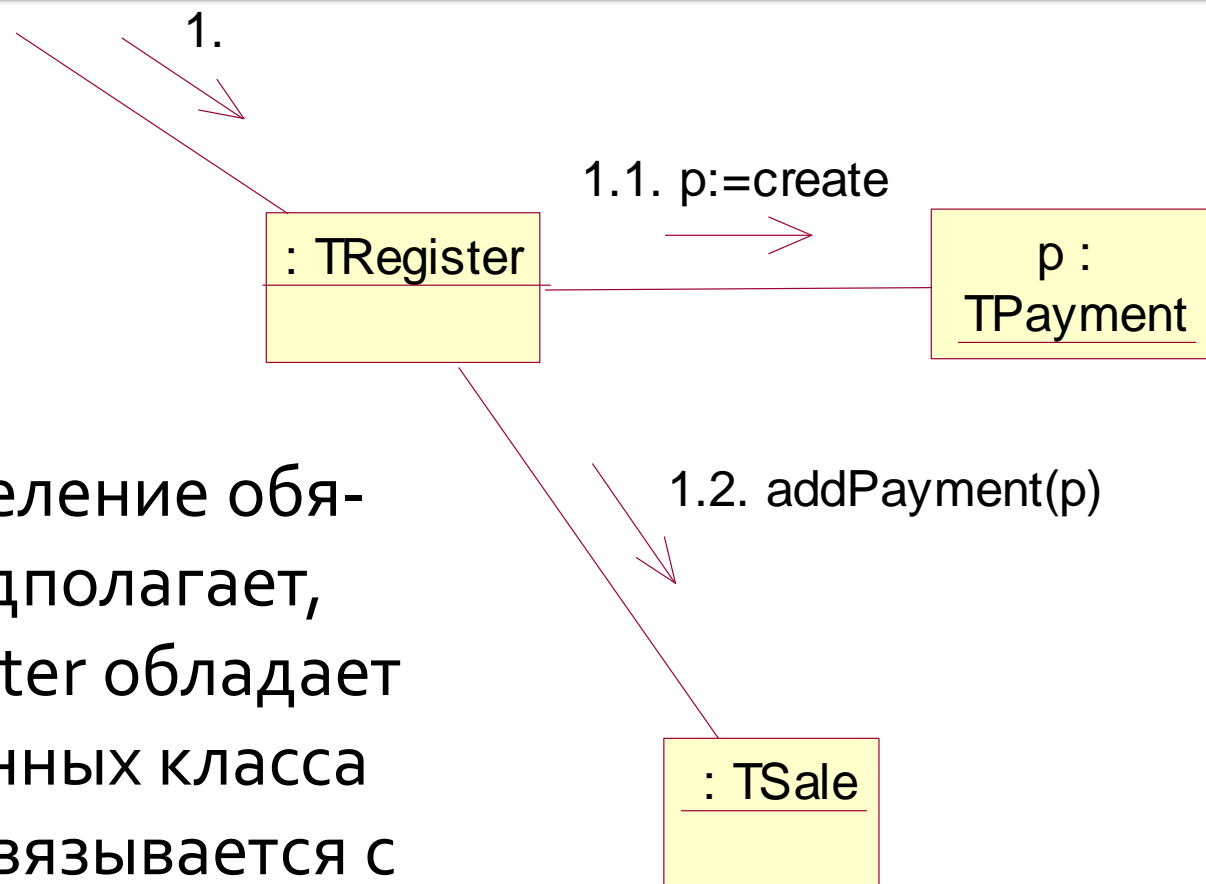
Пример

- Необходимо создать объект класса TPayment и связать его с объектом класса TSale
- Какой класс должен отвечать за выполнение этой операции?



Шаблон Low Coupling

Решение I вариант



Такое распределение обязанностей предполагает, что класс Register обладает знаниями о данных класса Payment (т.е. связывается с ним)

Шаблон Low Coupling

Решение II вариант

1. makePayment()



: Register

1.1. makePayment()



: Sale

1.1.1. create()



: Payment

С точки зрения числа связей между объектами, более предпочтительным является данный вариант, поскольку в этом случае обеспечивается низкая степень связывания

Шаблон Low Coupling

Преимущества

- Изменения компонентов мало сказываются на других объектах
- Принципы работы и функции компонентов можно понять, не изучая другие объекты
- Удобство повторного использования

Шаблон High Cohesion

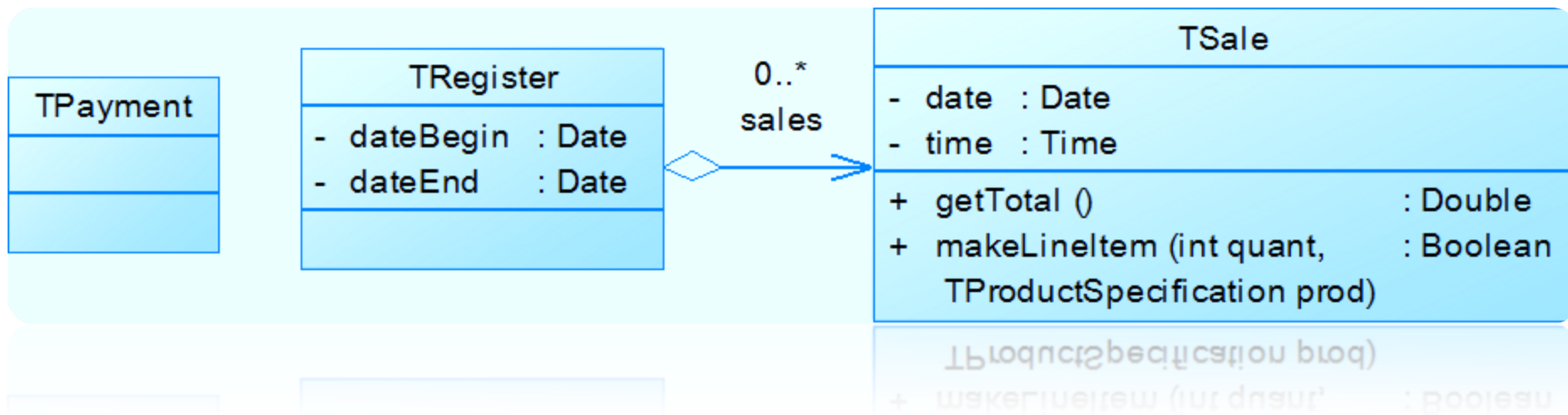
Высокое зацепление

- **Проблема.** Как обеспечить возможность управления сложностью?
- **Решение.** Распределение обязанностей, поддерживающее высокую степень зацепления
- **Зацепление** – это мера зависимости и сфокусированности обязанностей класса
 - Класс обладает высокой степенью зацепления, если его обязанности тесно связаны между собой
 - Класс с низкой степенью зацепления выполняет много разнородных функций

Шаблон High Cohesion

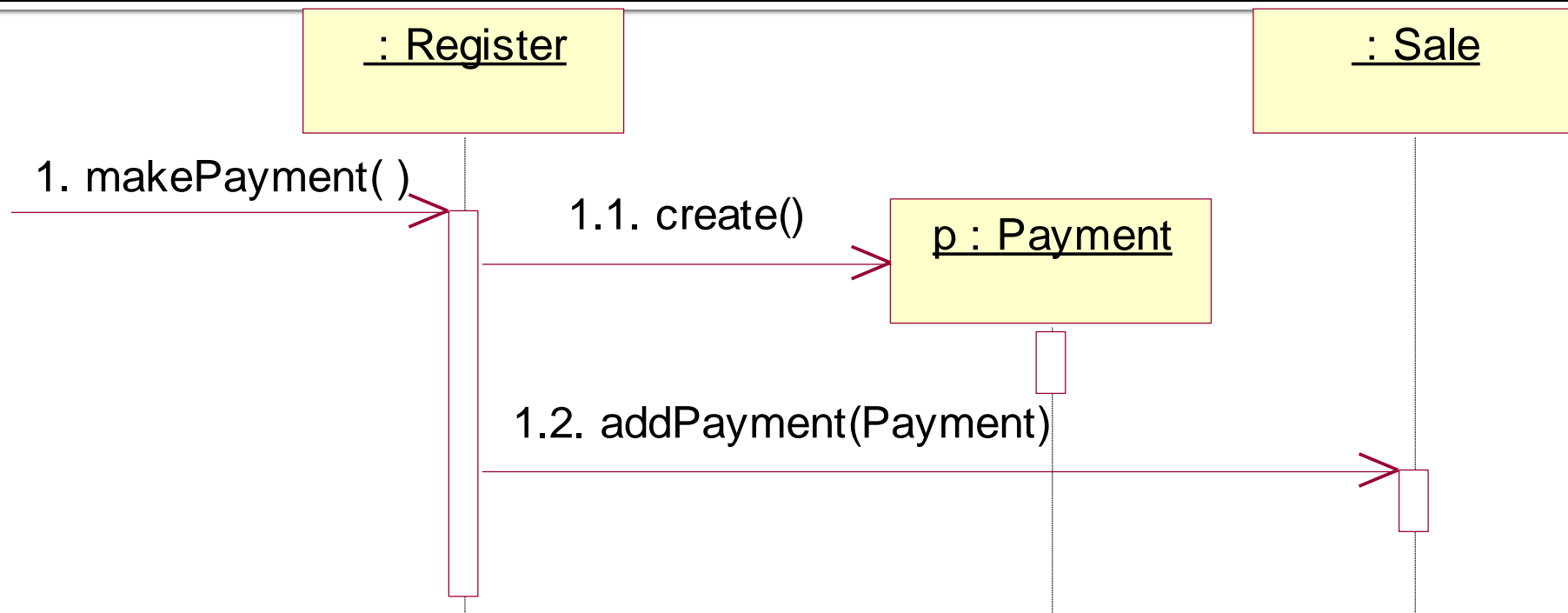
Пример

- Необходимо создать объект класса TPayment и связать его с объектом класса TSale
- Какой класс должен выполнять эту обязанность?



Шаблон High Cohesion

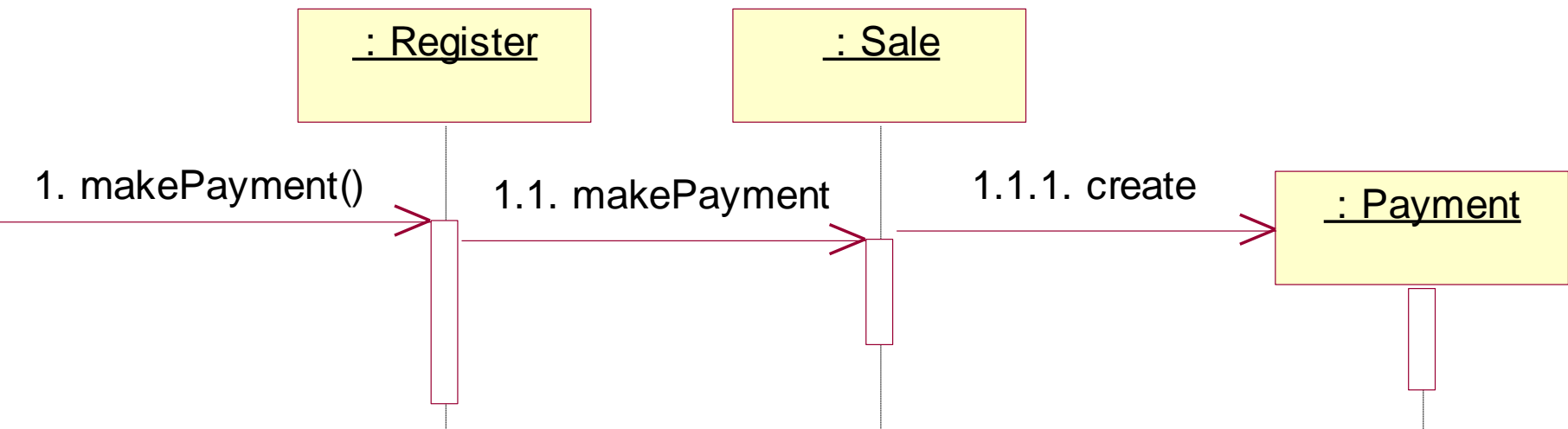
Решение I вариант



При таком распределении обязанностей платежи выполняет объект Register, т.е. объект Register частично несёт ответственность за выполнение операции `makePayment`

Шаблон High Cohesion

Решение II вариант



Здесь функция создания экземпляра платежа делегирована объекту Sale. Благодаря этому поддерживается более высокая степень зацепления объекта Register. Поскольку такой вариант распределения обязанностей обеспечивает низкий уровень связывания и более высокую степень зацепления, он является более предпочтительным

Сценарии различной степени функционального зацепления

- **Очень слабое зацепление.** Только один класс отвечает за выполнение множества операций в самых различных функциональных областях
- **Слабое зацепление.** Класс несёт единоличную ответственность за выполнение сложной задачи из одной функциональной области
- **Среднее зацепление.** Класс имеет среднее количество обязанностей из одной функциональной области и для выполнения своих задач взаимодействует с другими классами
- **Сильное зацепление.** Класс имеет несложные обязанности в одной функциональной области, логически связанные с концепцией этого класса

Шаблон High Cohesion

Преимущества

- Повышаются ясность и простота проектных решений
- Упрощаются поддержка и доработка
- Зачастую обеспечивается слабое связывание
- Улучшаются возможности повторного использования, поскольку класс с высокой степенью зацепления выполняет конкретную задачу