

Программирование на современном C++

Тестирование программного обеспечения.



Отметьтесь на портале!

- Посещение необязательное, но тем, кто пришёл, следует отмечаться на портале в начале каждого занятия
- Это позволяет нам анализировать, какие занятия были более или менее интересны студентам, и менять курс в лучшую сторону
- Также это даст возможность вам оставить обратную связь по занятию после его завершения


пн, 1 ноября	вт, 2 ноября	ср, 3 ноября	чт, 4 ноября	пт, 5 ноября	сб, 6 ноября
Нет занятий	<div>18:00 Углубленный C/C++ (w... п</div> <div>Обработка исключительных ситуаций. Шаблоны классов и методов. Обобщенное и безопасное программирование</div> <div>А. Халайджи</div> <div>18:00 Углубленный C/C++ (ML) п</div> <div>Обработка исключительных ситуаций. Шаблоны классов и методов. Обобщенное и безопасное программирование</div> <div>А. Халайджи</div>	Нет занятий	Нет занятий	Нет занятий	Нет занятий

Ссылка на Zoom РК сегодня в 18:00

Безопасность интернет-приложений (третий семестр)

Подключиться к конференции Zoom
<https://maillru.zoom.us/j/98586081818?pwd=eWxwSDdCbIhQNnNwQU5iblFvU2dTZz09>

Идентификатор конференции: 985 8608 1818
Код доступа: 785885

 Алексей Набережный 55 минут назад

★ 0 💬 0 ↓ 0 ↑

Запись прошлой лекции (+ что почитать перед РК)

Безопасность интернет-приложений (третий семестр)

Углублённый C/C++

Лекция 5

📍 **Онлайн - ML**

Отметьтесь, что вы пришли на занятие. Так вы улучшите свою посещаемость и вас увидит преподаватель в своём "Журнале посещений".

Оставьте отзыв о занятии и мы сможем улучшить учебный процесс.

План лекции

- Тестирование ПО
- Модульное тестирование
- Перерыв
- Дублеры компонентов
- Интеграционное тестирование

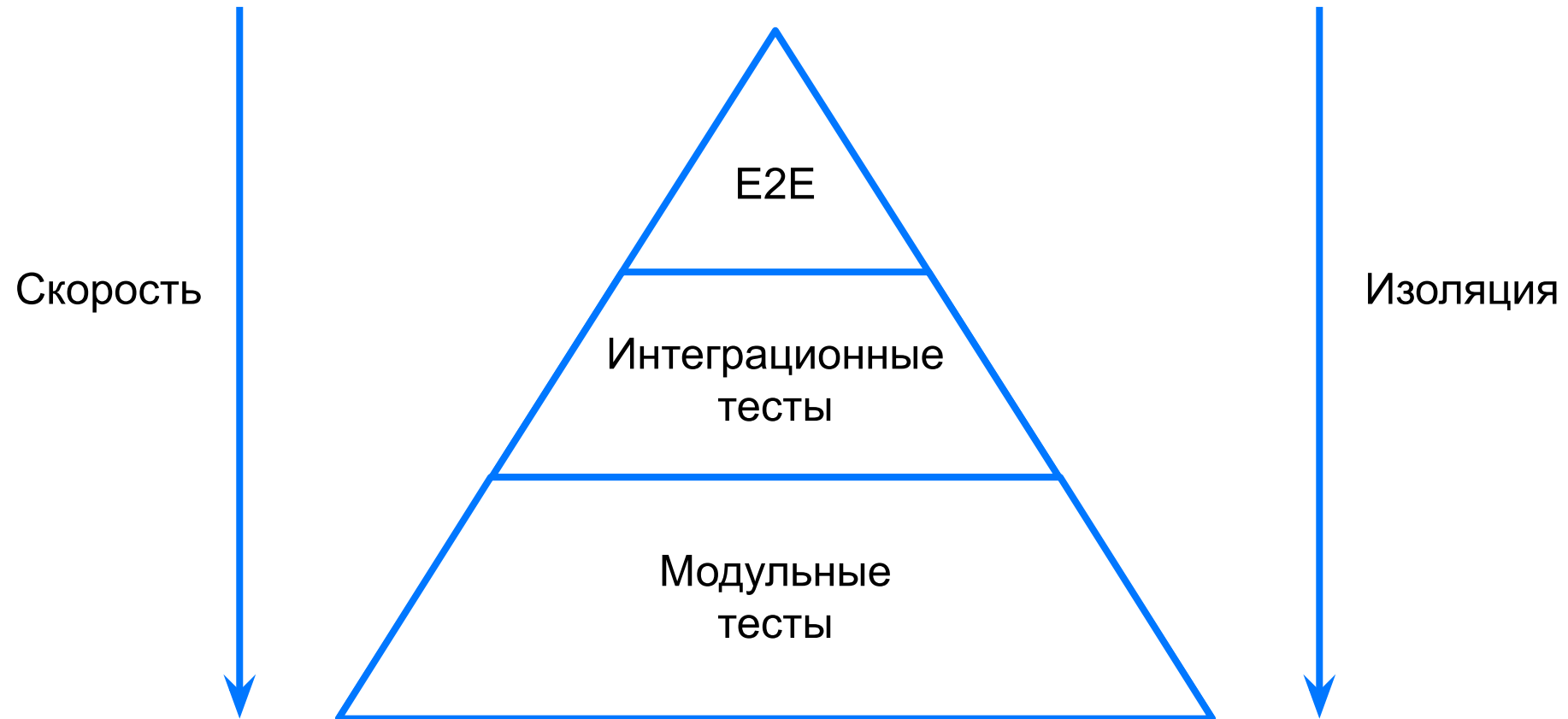
Модульное тестирование

Модульное тестирование стало неотъемлемой частью процесса создания программного обеспечения.

Тестирование программного обеспечения

Процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом.

Тестирование программного обеспечения



Регрессионное тестирование

Изменения в коде могут приводить к появлению ошибок. Такие ошибки называются регрессионными.

Регрессионное тестирование

Изменения в коде могут приводить к появлению ошибок. Такие ошибки называются регрессионными.

Модульное тестирование позволяет поддерживать устойчивое развитие программного обеспечения.

Регрессионное тестирование

Пример

GTest

```
#include <gtest/gtest.h>
```

```
TEST(SplitStringTest, BasicCase) {  
    EXPECT_EQ(SplitString(""), std::vector<std::string>{});  
}
```

Оценка полноты покрытия тестами

Как понять хватает тестов или нет?

Оценка полноты покрытия тестами

Как понять хватает тестов или нет?

Покрытие кода (Code coverage) - процент исходного кода, выполненный во время выполнения тестов.

Оценка полноты покрытия тестами

Как понять хватает тестов или нет?

Покрытие кода (Code coverage) - процент исходного кода, выполненный во время выполнения тестов.

Высокий процент покрытия - не означает отсутствие багов.

Когда “работает” модульное тестирование

- Тесты встроены в процесс разработки
- Тесты проверяют самые важные части кода (бизнес логику)
- Тесты минимизируют стоимость поддержки

Основные свойства модульного теста

- Проверяет малую часть кода (класс или метод)
- Делает это быстро
- Делает это **изолировано**

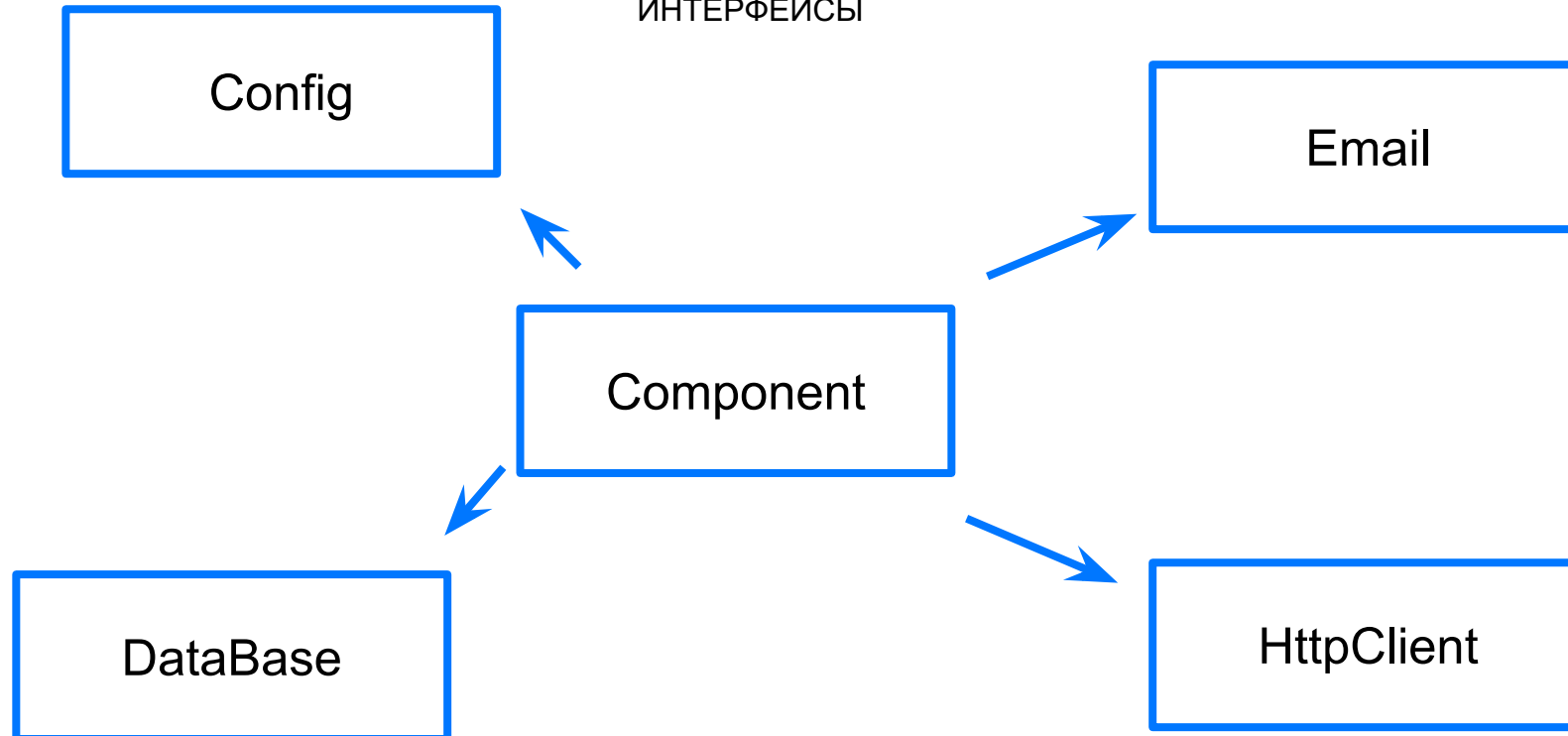
Основные свойства модульного теста

- Проверяет малую часть кода (класс или метод)
- Делает это быстро
- Делает это **изолировано** НЕ ВЛИЯЮТ ДРУГ НА ДРУГА

НЕ ИСПОЛЬЗУЮТ ОБЩИЕ РЕСУРСЫ, ГЛОБАЛЬНЫЕ ПЕРЕМЕННЫЕ

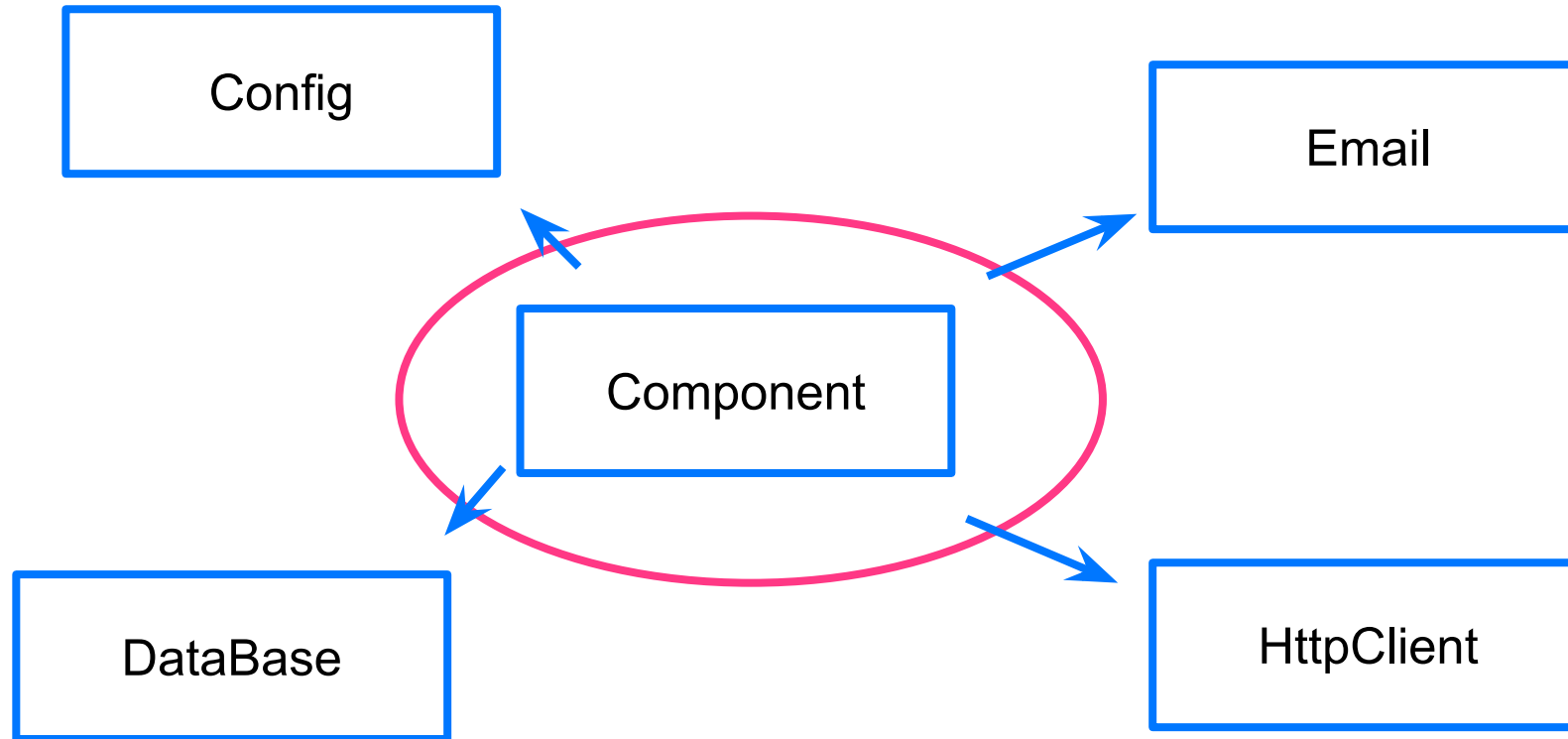
Как протестировать сложный компонент

ЗАДАЧА ИЗОЛИРОВАТЬ ЗАВИСИМОСТИ ОТ КОМПОНЕНТА - ЧЕРЕЗ ИНТЕРФЕЙСЫ



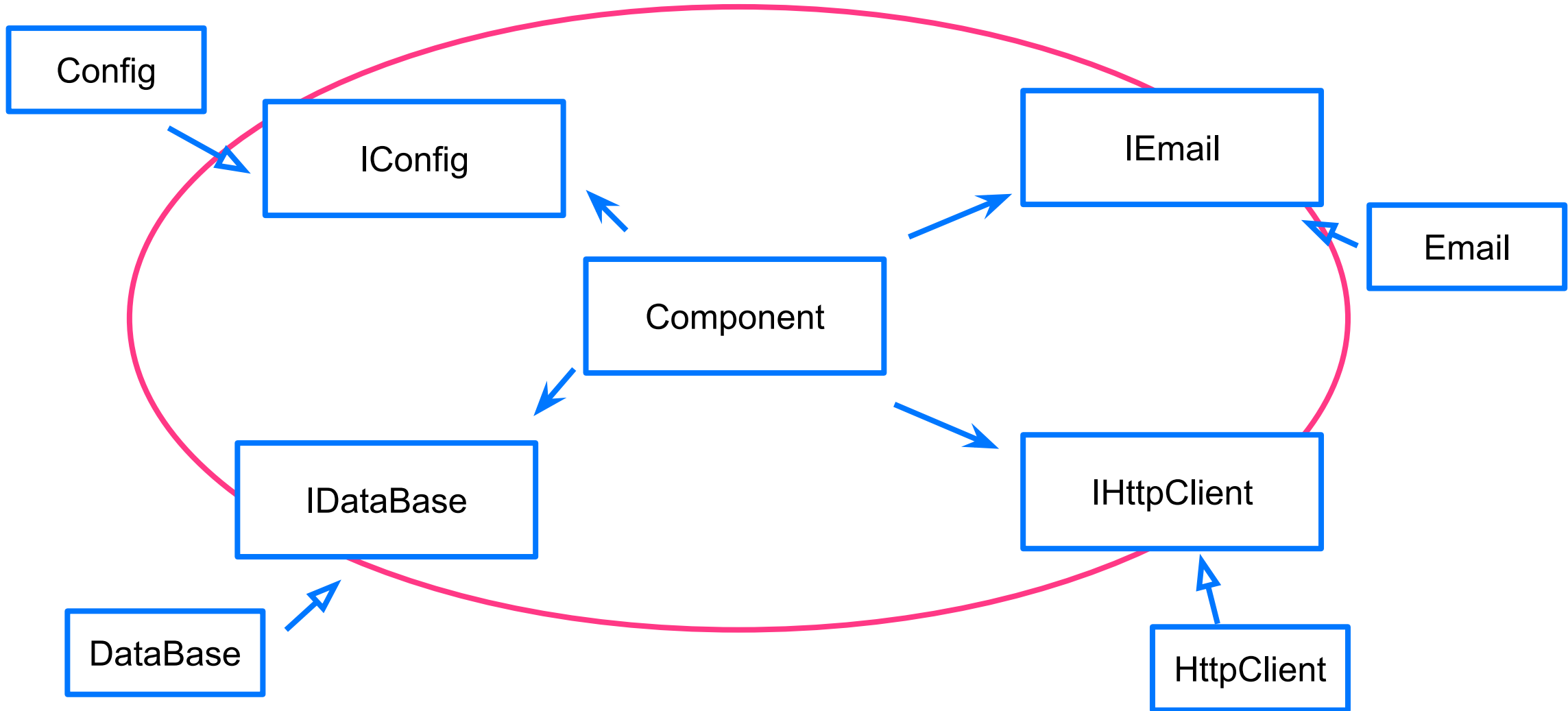
Практика

Как протестировать сложный компонент

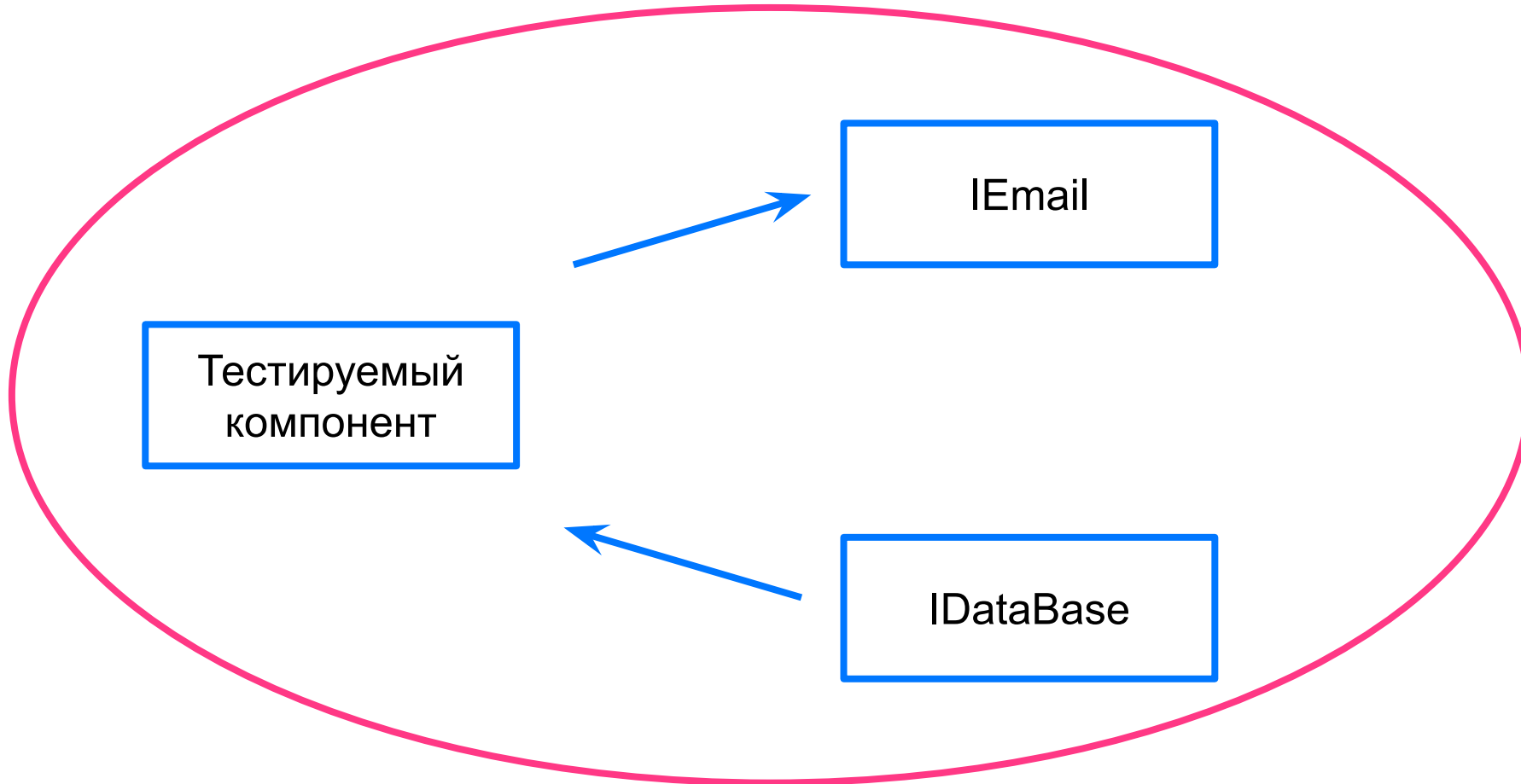


НЕОБХОДИМО ОБЕСПЕЧИТЬ ЗАВИСИМОСТЬ КОМПОНЕНТА ОТ ИНТЕРФЕЙСНЫХ КЛАССОВ, ТОГДА ЭТИ ИНТЕРФЕЙСЫ МОЖНО БУДЕТ ПОДМЕНИТЬ, В ПРОДАКШЕНЕ УЖЕ БУДЕМ ИСПОЛЬЗОВАТЬ РЕАЛЬНЫЕ СЛОЖНЫЕ ЗАВИСИМОСТИ

Как протестировать сложный компонент

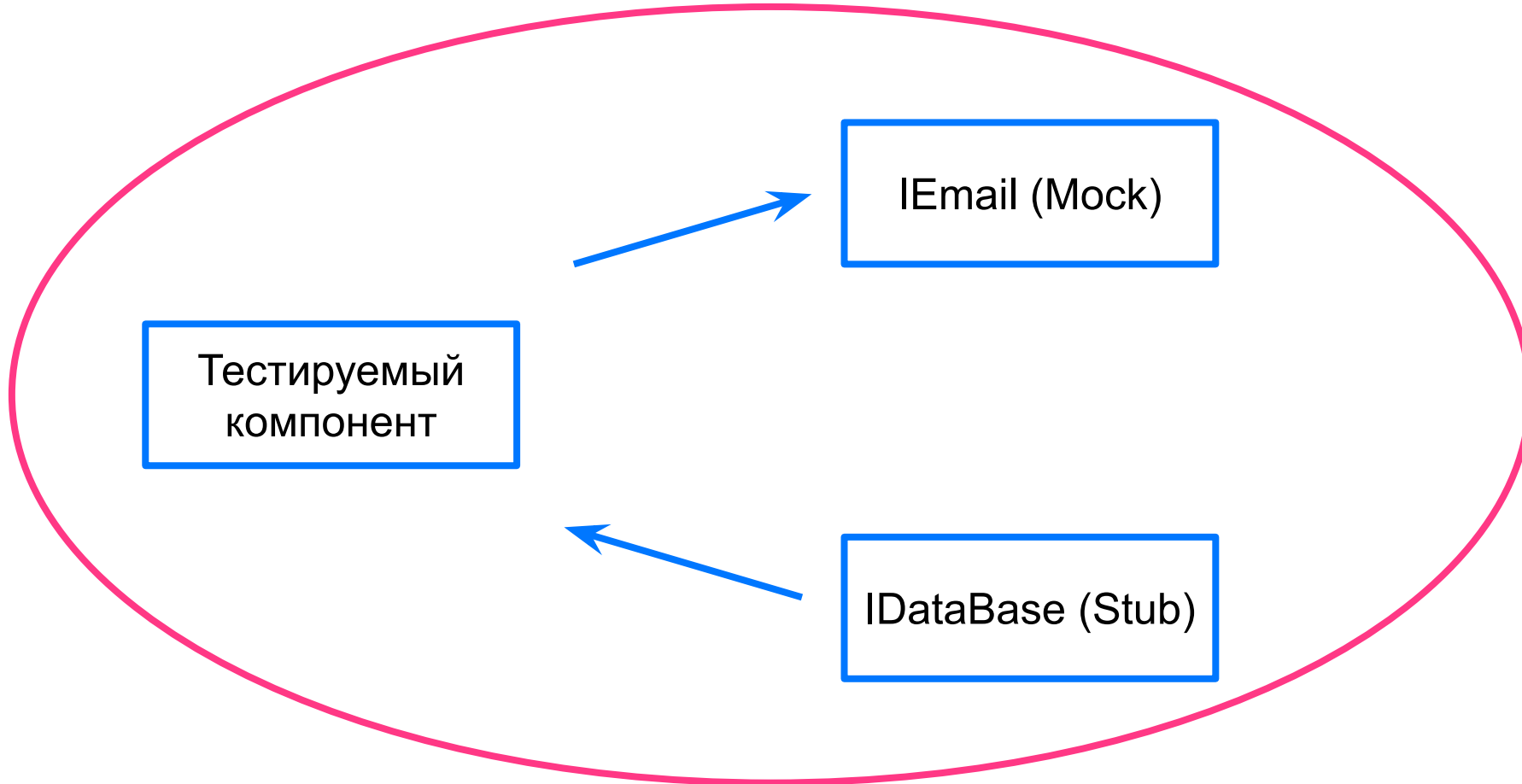


Дублиры зависимостей



STUB - ЗАВИСИМОСТЬ, ИЗ КОТОРОЙ ЗАБИРАЕМ ДАННЫЕ, ВНУТРИ СТАБА НЕ МОЖЕТ БЫТЬ НАРУШЕНИЯ ЛОГИКИ, ОН ЧТО-ТО ОТДАЕТ

Дублиры зависимостей



Дублиры зависимостей

Пример

Модульное тестирование и архитектура

Модульное тестирование неразрывно связано с хорошей архитектурой приложения.

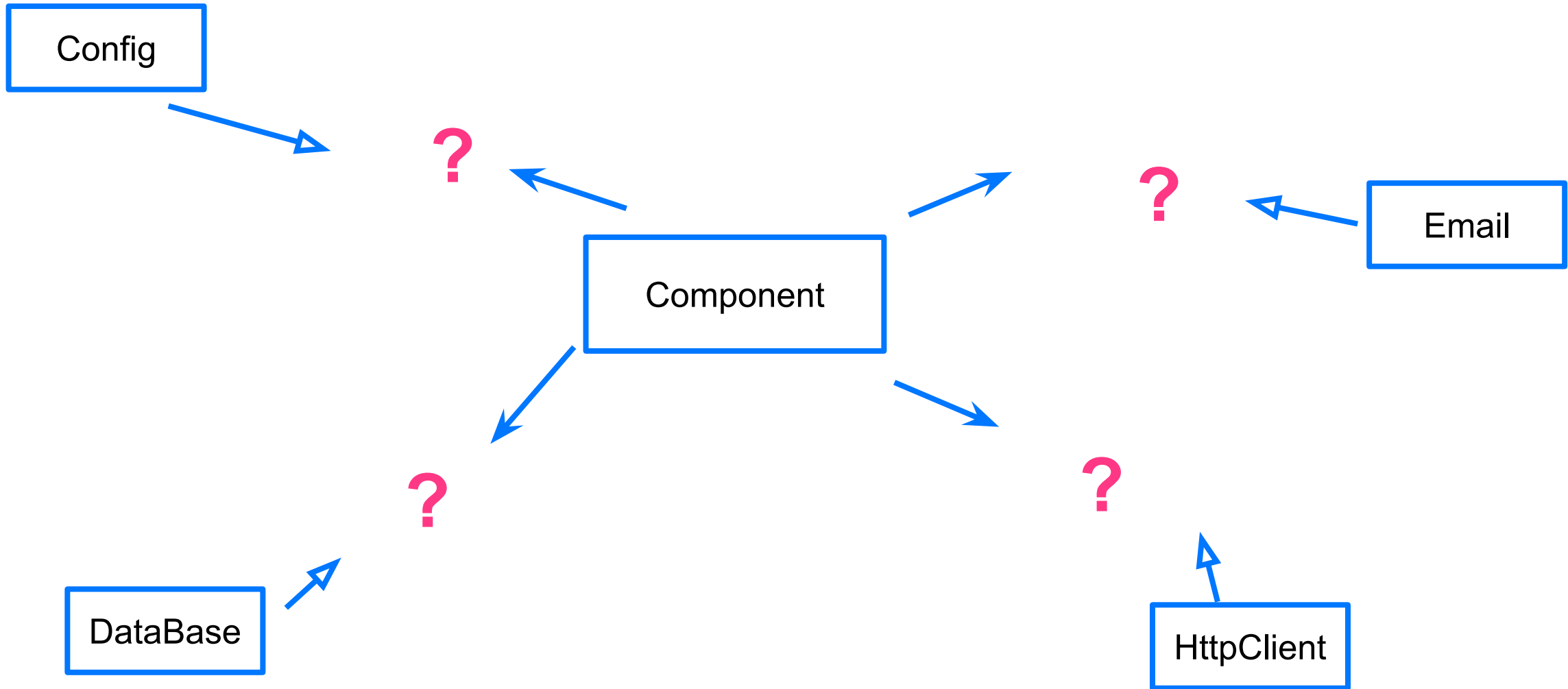
Модульное тестирование и архитектура

Модульное тестирование неразрывно связано с хорошей архитектурой приложения.

- Dependency Inversion
- IoC
- SRP
- Низкая связность кода

Могут ли еще где-то быть ошибки?

Могут ли еще где-то быть ошибки?



Интеграционное тестирование

Тестирование отдельных программных модулей в группе

Интеграционное тестирование

Тестирование отдельных программных модулей в группе

- Допускается подмена упрощенными реализациями
- Поверяется взаимодействие

Спасибо за внимание!

