



C++ developer. Professional

Шаблоны GoF. Порождающие



Проверить, идет ли запись

Меня хорошо видно && слышно?



Ставим “+”, если все хорошо
“-”, если есть проблемы

Тема вебинара

Шаблоны GoF. Порождающие



Карина Дорожкина

Research Development Team Lead

10 лет опыта разработки на C/C++.

dorozhkinak@gmail.com

Правила вебинара



Активно
участвуем



Off-topic обсуждаем
в telegram C++-2023-08



Задаем вопрос
в чат

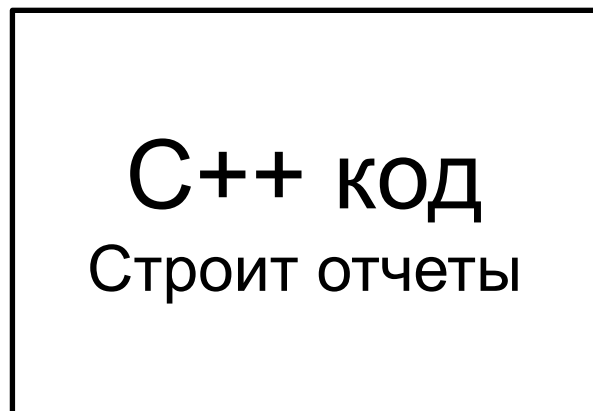


Вопросы вижу в чате,
могу ответить не сразу

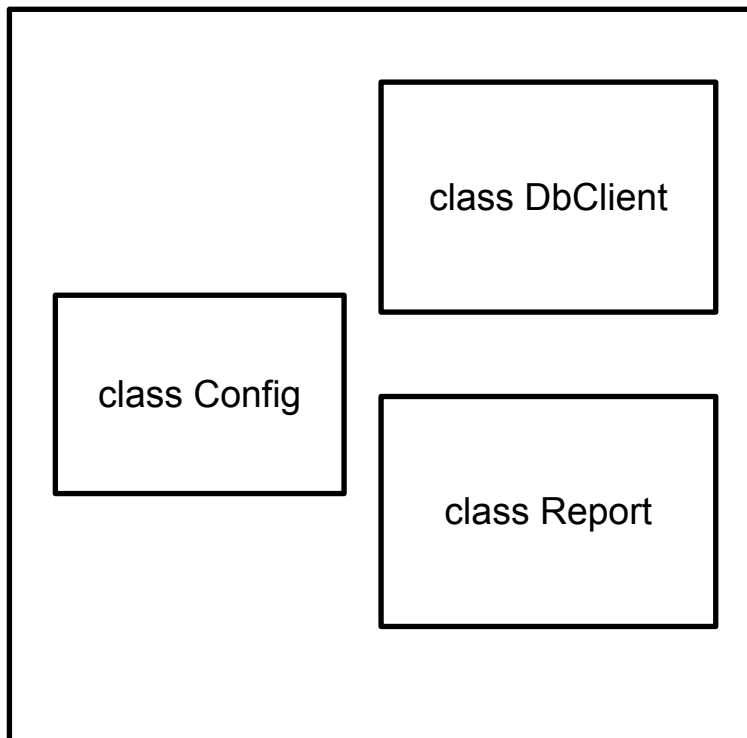
Куда идем



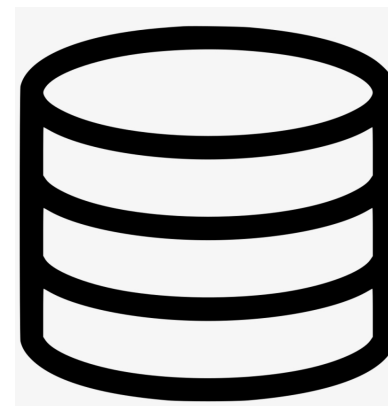
1. Шаблоны проектирования GoF
2. Порождающие паттерны



C++ код



DB



GoF patterns



GoF patterns

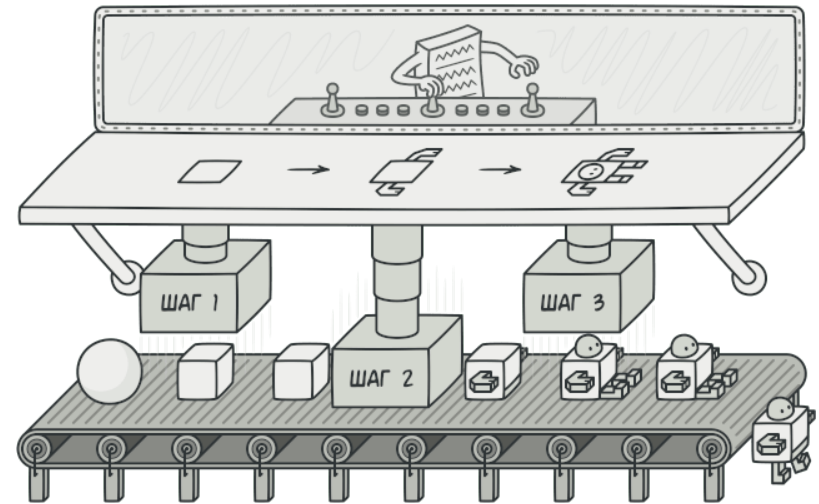
Шаблоны проектирования:

- порождающие (Creational pattern)
- структурные (Structural pattern)
- поведенческие (Behavioral pattern)
- конкурентные (Concurrency pattern)



Порождающие

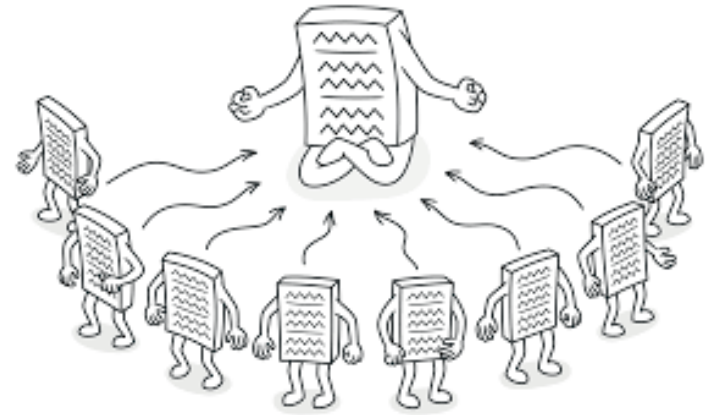
- Одиночка (Singleton)
- Пул одиночек (Multiton)
- Объектный пул (Object pool)
- Фабричный метод (Factory method)
- Абстрактная фабрика (Abstract factory)
- Строитель (Builder)
- Прототип (Prototype)
- Внедрение зависимости (Dependency Injection)
- Ленивая инициализация (Lazy initialization)



Одиночка

Singleton

- Единственный и неповторимый (+ -)
- Отложено инициализируемый (+)
- Простой (+)
- Глобально доступный (-)
- Сложный - многопоточность (-)
- Статическое время жизни (-)
- Порядок инициализации (-)



Примеры реализаций logger

- log4cplus

<https://github.com/log4cplus/log4cplus>

- Boost.Log

<https://www.boost.org/doc/libs/master/libs/log/doc/html/index.html>

Пул одиночек

Multiton

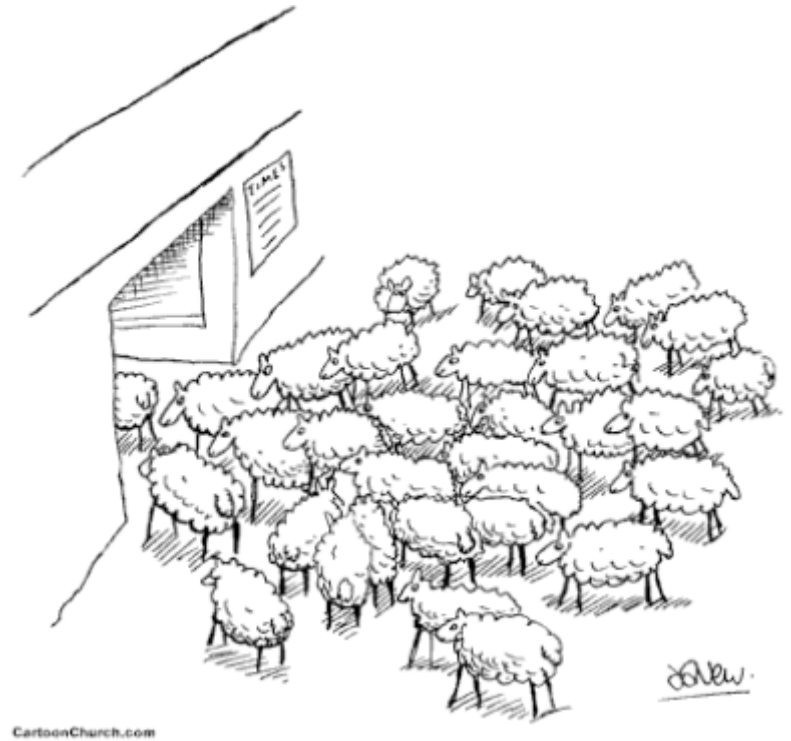
- Несколько однотипных одиночек (+ -)
- Отложенная инициализация (+)
- Простой способ получить объект (+)
- Статическое время жизни каждого объекта (-)



Объектный пул

Object pool

- Отложенная инициализация (+)
- Переиспользование объектов (+)
- Сложность реализации (-)
- Сложность использования (-)
- Необходимость компромиссов (+ -)



Фабричный метод

Factory method

- Тебе нужно – ты и создавай (+)
- Простота использования (+)
- Простота реализации (+)
- Расширяемость (+)
- Много ответственностей (-)
- Большая иерархия классов - дублирование (-)



Фабричный метод

Creator GRASP

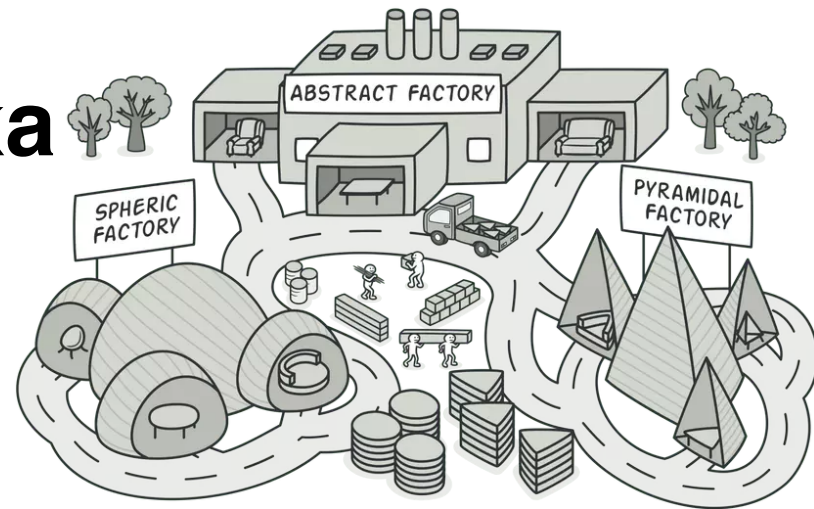
Класс должен создавать экземпляры:

- агрегированных классов
- классов, которые содержит
- классов, который активно использует
- классов, для инициализации которых обладает
наибольшей информацией

Абстрактная фабрика

Abstract factory

- Создание семейств объектов (+)
- Легко расширяем (+)
- Прозрачно для клиентов (+)
- Гарантирует сочетаемость объектов (+)
- Принуждает к работе на уровне интерфейса (+)
- Требует единообразия семейств (+ -)
- Трудно менять состав семейств (-)



Строитель

Builder

- Пошаговое строительство объектов (+ ??)
- Создание разных объектов (+)
- Использование прототипов (+)
- Изолированность кода создания (+)
- Зависимость от конкретных классов (-)



Прототип

Prototype



- Простое клонирование объектов (+)
- DRY – «нет» повторной инициализации (+)
- Глубокое или поверхностное? (+ -)
- Требует клонирования базовых классов (-)
- Клонировать не клонируемое? (-)

Внедрение зависимости

Dependency Injection

- Используем постоянно (+)
- Поддерживается языком напрямую (+)
- Виртуальные функции (+ -)
- Срезка при передаче по значению (-)

INJECT ALL THE THINGS!!!



Ленивая инициализация

Lazy initialization

- Отложенные вычисления (+)
- Делаем только по требованию (+)
- Непредсказуемое замедление (-)
- Потокобезопасность (-)



Что нового сегодня узнали?

Порождающие паттерны

- Одиночка (Singleton)
- Пул одиночек (Multiton)
- Объектный пул (Object pool)
- Фабричный метод (Factory method)
- Абстрактная фабрика (Abstract factory)
- Строитель (Builder)
- Прототип (Prototype)
- Внедрение зависимости (Dependency Injection)
- Ленивая инициализация (Lazy initialization)

Спасибо за внимание!

Вопросы?



Ставим “+”,
если вопросы есть



Ставим “-”,
если вопросов нет

**Не забудьте принять
участие в опросе**

Следующий вебинар



18 октября 2023

Шаблоны GoF. Структурные



Ссылка на вебинар будет в ЛК за 15 минут



Материалы к занятию в ЛК — можно изучать



Обязательный материал обозначен красной лентой

