Порождающие шаблоны проектирования

Builder, Singleton, Factory Method, Abstract Factory, Prototype

Иван Поляков
Разработчик Go/C++ в инфраструктуре поиска в Авито

Проверка связи

Если у вас нет звука:

- убедитесь, что на вашем устройстве и на колонках включён звук
- обновите страницу вебинара или закройте страницу и заново присоединитесь к вебинару
- откройте вебинар в другом браузере
- перезагрузите компьютер (ноутбук) и заново попытайтесь зайти

Поставьте в чат:

\+ если меня видно и слышно

— если нет

Иван Поляков

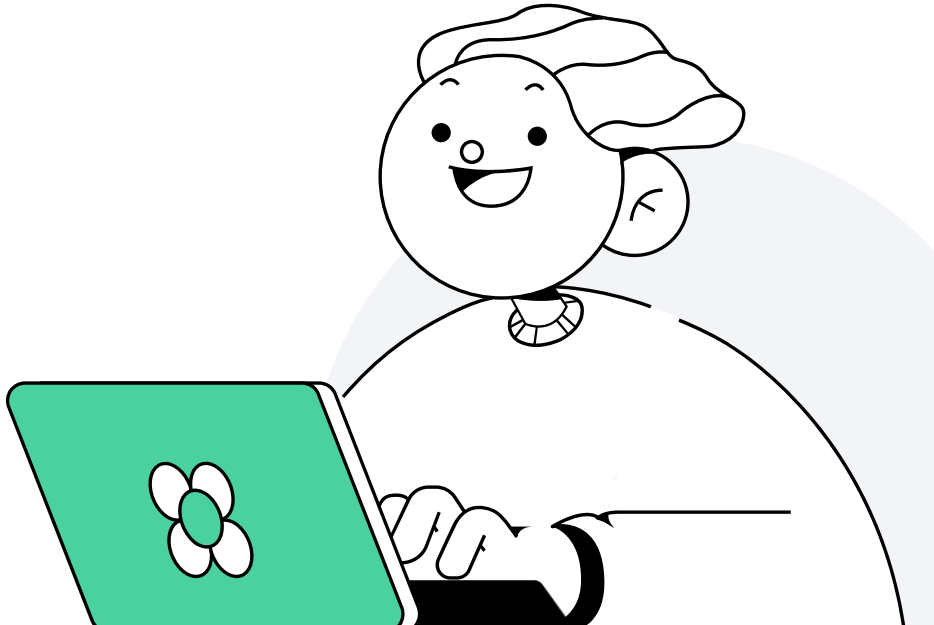
О спикере:

- Разработчик Go/C++ в инфраструктуре поиска в Авито
- 5 лет работал в Nexign, писал real time сервисы для телекома Мегафона на C++

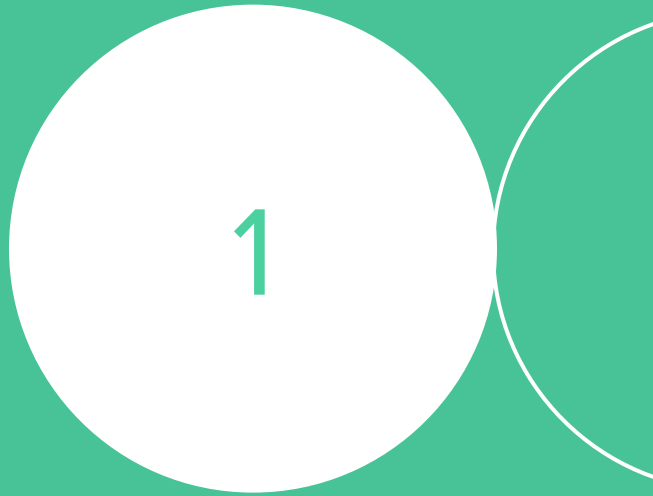
План занятия

1. Паттерны проектирования
2. Классификация паттернов
3. Фабричный метод
4. Абстрактная фабрика
5. Строитель (Builder)
6. Одиночка (Singleton)
7. Итоги
8. Домашнее задание

Нажмите на нужный раздел для перехода

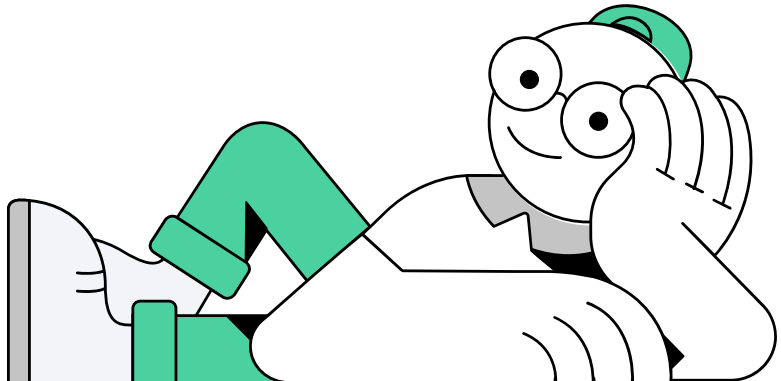


Паттерны проектирования



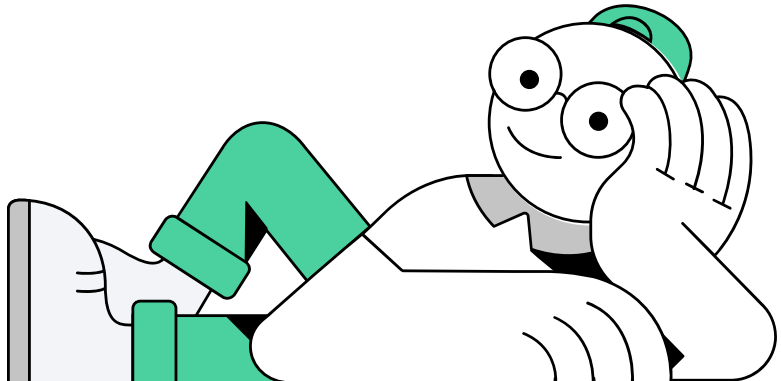
Что такое паттерн проектирования

- Типичный способ решения часто встречающихся проблем, задач при проектировании программ



Что такое паттерн проектирования

- Типичный способ решения часто встречающихся проблем, задач при проектировании программ
- В отличие от алгоритма, это не набор конкретных шагов, а образец (шаблон), по которому мы решаем задачи



Зачем нужны паттерны

1. Не переизобретать решения

Зачем нужны паттерны

1. Не переизобретать решения
2. Находить общий язык с другими программистами

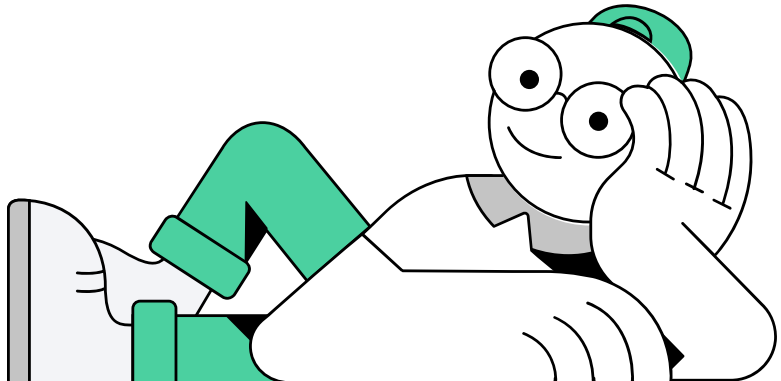
Зачем нужны паттерны

1. Не переизобретать решения
2. Находить общий язык с другими программистами
3. Отвечать на вопрос о любимом паттерне на собеседованиях

Что такое паттерн проектирования

Иногда это формализованный здравый смысл.

Возможно, вы уже используете паттерны, но не знаете, как они называются

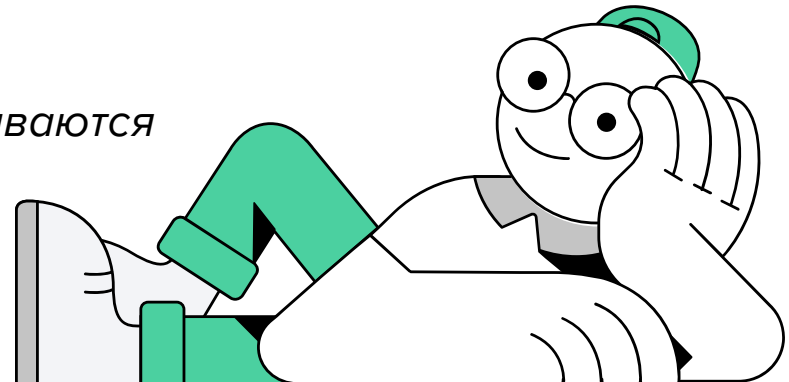


Что такое паттерн проектирования

Иногда это формализованный здравый смысл.

Возможно, вы уже используете паттерны, но не знаете, как они называются.

Программист проходит медкомиссию. Окулист ему:
— Закройте левый глаз. Какая буква?
Программист молчит.
— Закройте левый глаз. Какая буква?
Программист молчит.
— Вы что, совсем ничего не видите?
— Нет, вижу отлично. Просто забыл, как они называются



Классификация паттернов

2

Немного истории

Паттерны пришли из архитектуры.

В 1994 году вышла знаменитая книга «банды четырёх» «Приёмы объектно-ориентированного проектирования. Паттерны проектирования».

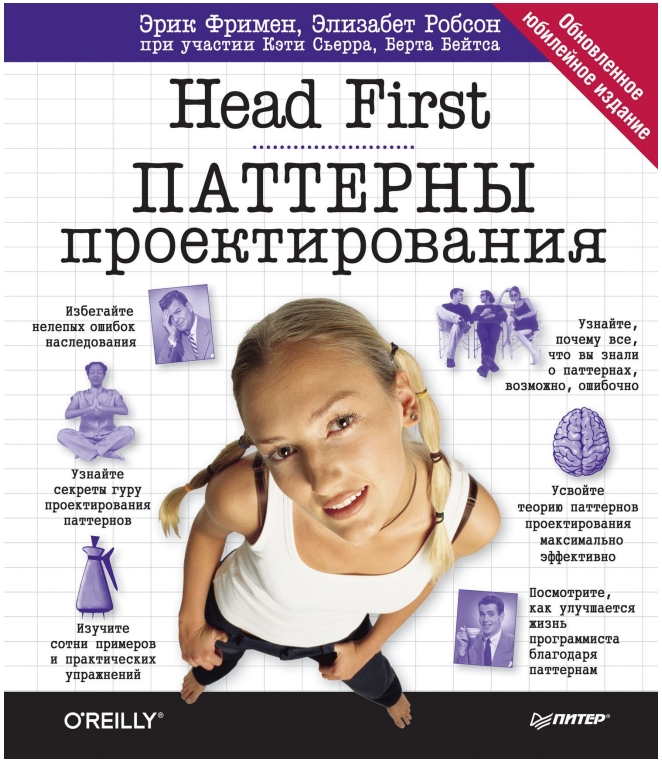
В ней были описаны 23 паттерна.

В дальнейшем были разработаны другие паттерны

Источник: www.artlebedev.ru

Книги о паттернах

Источник



Источник

Другие паттерны

1. Паттерны проектирования систем
2. Паттерны параллельного программирования
3. Паттерны баз данных
4. Антипаттерны
5. ...

Критика паттернов

1. Устаревание

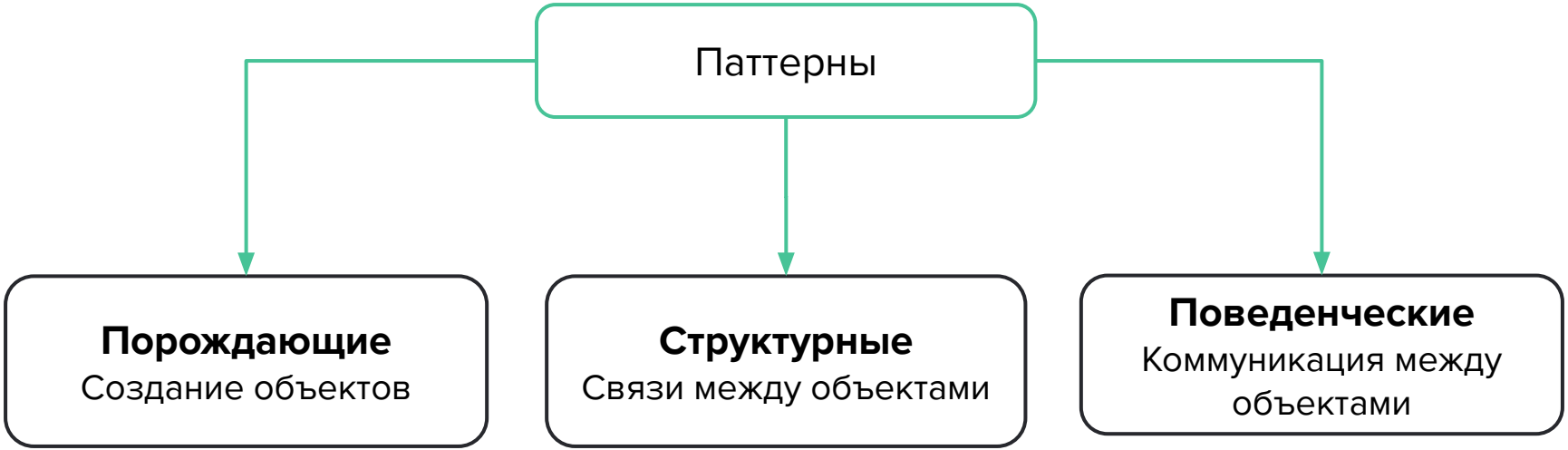
Критика паттернов

1. Устаревание
2. Усложнение

Критика паттернов

1. Устаревание
2. Усложнение
3. «Универсальность»

Классификация паттернов



Порождающие паттерны

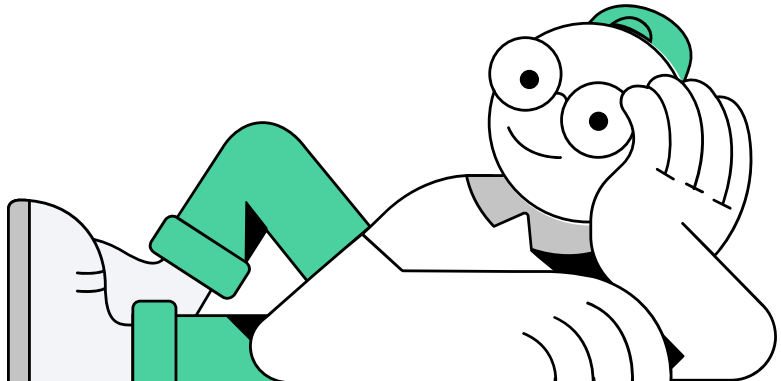
1. Фабричный метод
2. Абстрактная фабрика
3. Строитель
4. Одиночка
5. Прототип

Фабричный метод

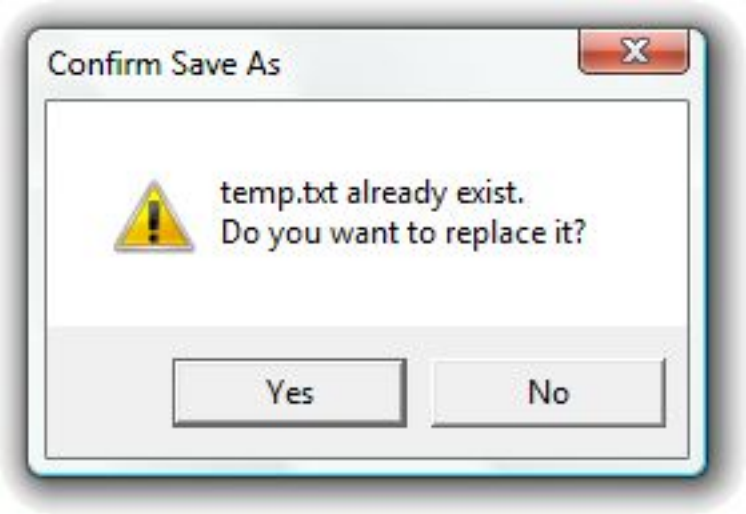
3

Фабричный метод

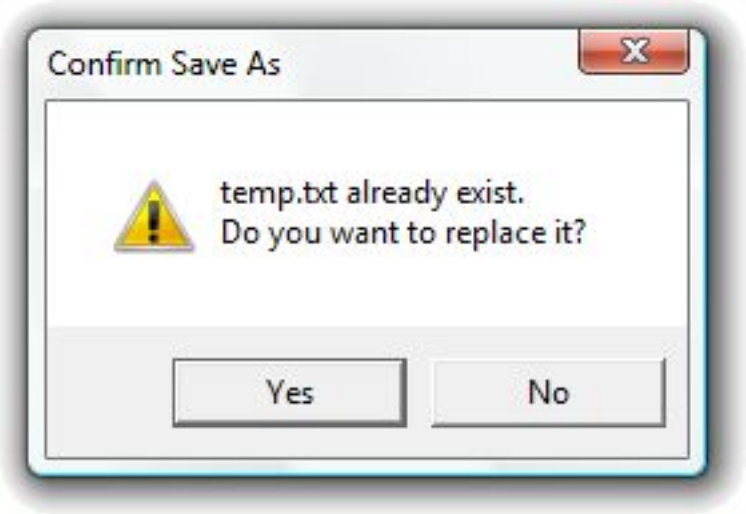
- Определяем интерфейс создания объектов в родительском классе
- Позволяем наследникам изменять тип создаваемых объектов

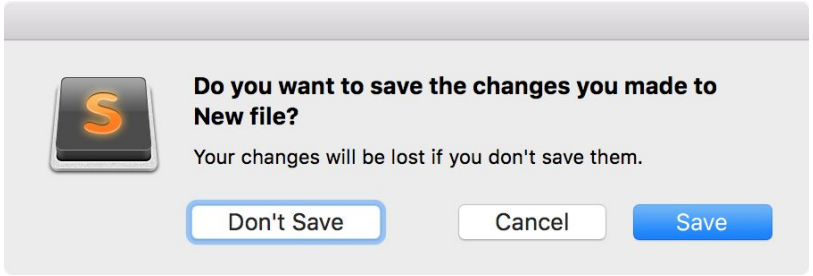


Фабричный метод. Пример

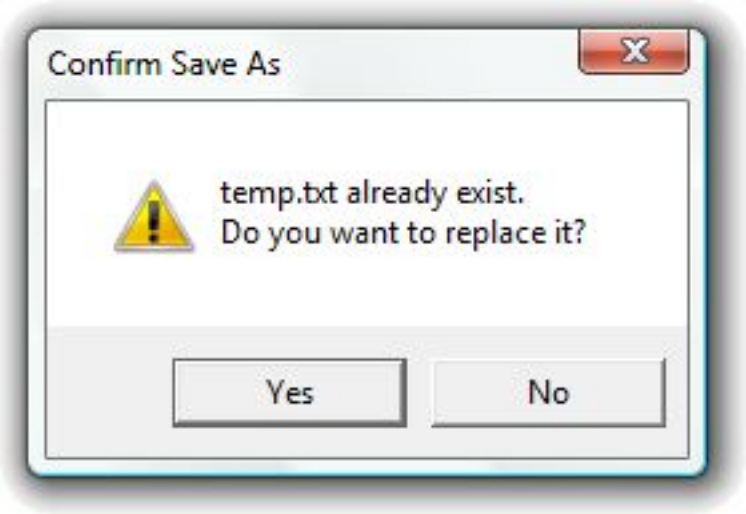


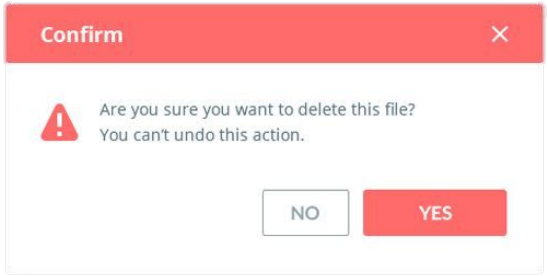
Фабричный метод. Пример

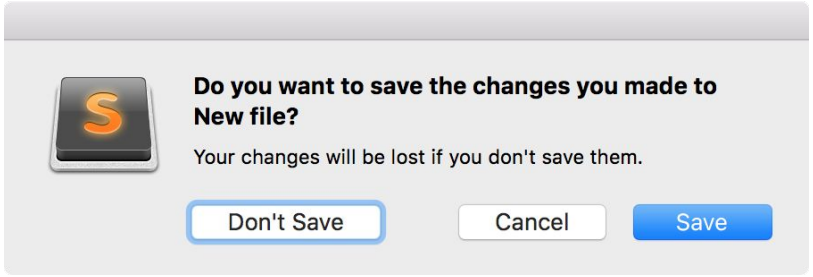




Фабричный метод. Пример







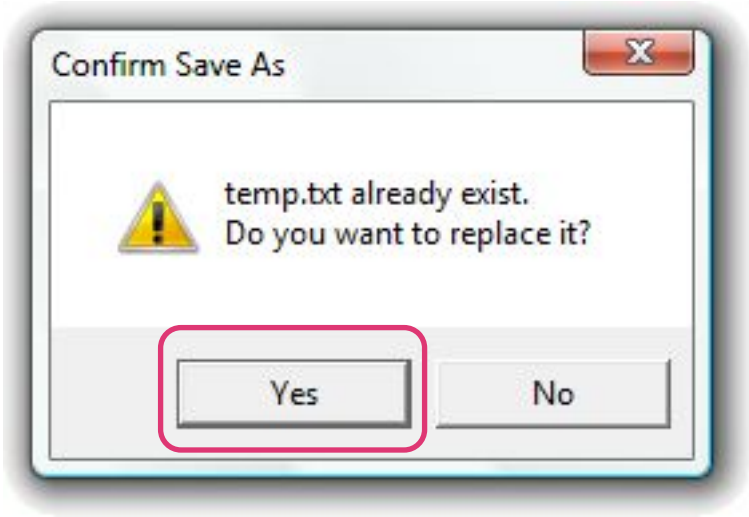
Абстрактная фабрика

4

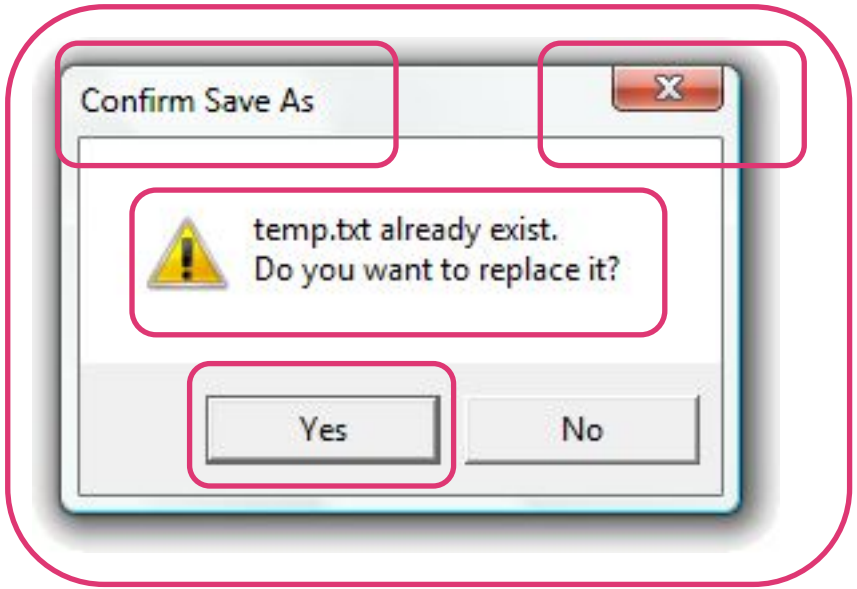
Абстрактная фабрика

- Развитие идеи фабричного метода
- Позволяет создавать семейства связанных объектов

Абстрактная фабрика. Пример



Абстрактная фабрика. Пример



Строитель

4

Строитель

Выносит построение объектов в отдельную сущность с набором методов

Строитель

Выносит построение объектов в отдельную сущность с набором методов.

Примеры

1. Избавление от большого количества конструкторов или параметров конструктора
2. Создание различных представлений (пример с шаблонами)
3. Сборка сложных объектов, например рекурсивных

Одиночка

5

Одиночка (Singleton)

Гарантирует наличие единственного экземпляра объекта

Одиночка (Singleton)

Гарантирует наличие единственного экземпляра объекта.

1. Контроль доступа к объекту из разных частей программы

Одиночка (Singleton)

Гарантирует наличие единственного экземпляра объекта.

1. Контроль доступа к объекту из разных частей программы
2. «Ленивая» инициализация тяжёлого объекта

Одиночка (Singleton)

Гарантирует наличие единственного экземпляра объекта.

1. Контроль доступа к объекту из разных частей программы
2. «Ленивая» инициализация тяжёлого объекта
3. **Будьте внимательны с многопоточностью**

Итоги занятия

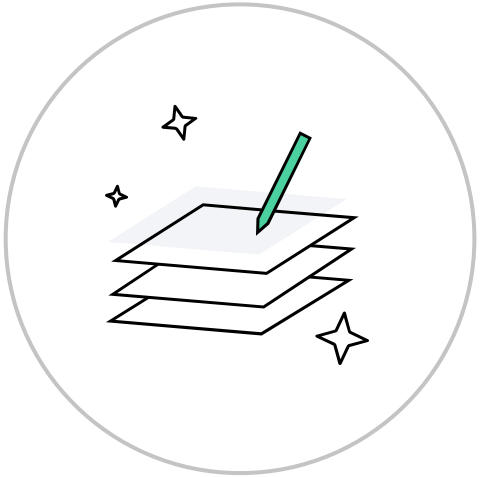
Сегодня мы:

- узнали, что такое паттерны проектирования и зачем они нужны
- разобрались в классификации паттернов
- рассмотрели подробнее порождающие паттерны

Домашнее задание

Давайте посмотрим ваше домашнее задание.

1. Вопросы по домашней работе задавайте в чате группы
2. Задачи можно сдавать по частям
3. Зачёт по домашней работе ставят после того, как приняты все задачи



Задавайте вопросы и пишите отзыв о лекции

Иван Поляков
Разработчик Go/C++ в инфраструктуре поиска в Авито