NL.17: Use K&R-derived layout





ОНЛАЙН-ОБРАЗОВАНИЕ



Архитектура GRASP

Дмитрий Шебордаев

Ведущий специалист





Сегодня

- ► SoC, DRY, YAGNI, KISS, UML
- ► GRASP



GRASP

- ► General Responsibility Assignment Software Patterns
- ▶ Общие шаблоны распределения обязанностей в ПО



Общие шаблоны распределения обязанностей в ПО

- ► Слабое зацепление (Low Coupling)
- ► Высокая связность (High Cohesion)
- ▶ Создатель (Creator)
- ▶ Информационный эксперт (Information Expert)
- ▶ Контроллер (Controller)
- ▶ Полиморфизм (Polymorphism)
- Чистая выдумка (Pure Fabrication)
- ▶ Посредник (Indirection)
- Устойчивость к изменениям (Protected Variations)



Low Coupling

- Как снизить влияние изменений?
- Уменьшить количество связей.
- Помогает выбирать между разными реализациями.



High Cohesion

- Как обеспечить сфокусированность функционала?
- Кластеризовать функционал на этапе проектирования.
- ▶ Помогает выбирать между разными реализациями.



Creator

- Кто отвечает за создание экземпляра класса?
- ▶ Тот, кто содержит или является экспертом.
- Помогает снизить количество связей.



Information Expert

- ▶ Кто выполняет операцию?
- ▶ Тот, у кого достаточно данных.
- Помогает снизить количество связей.



Controller

- Кто должен отвечать за внешние операции?
- Классу системы или классу сценария.
- ► API, UI
- ▶ не раздуваем дробим



Polymorphism

- Как обрабатывать альтернативные варианты поведения?
- С помощью полиморфных вызовов.
- Не злоупотреблять! Альтернатива должна быть осязаема.



Pure Fabrication

- Что делать, если зашли в тупик?
- Придумали новую сущность.
- ► Не злоупотреблять! Предметная область не должна размываться.



Indirection

- Как избежать прямого связывания?
- ▶ Поручить другому.
- ► MVC.



Protected Variations

- ▶ Как защититься от изменений?
- ▶ Изолировать изменения или изменяемое.
- Полиморфизм.
- ▶ Конфиги.
- ▶ Скрипты.
- ▶ Внешние интерфейсы.



$O \mathcal{T} U S$

почитать

http://www.ozon.ru/context/detail/id/3105480/





Спасибо за внимание!

