Лекция №11

- Паттерны
  - Порождающие
  - Структурные
  - Поведенческие
- Антипаттерны
  - В ООП, в кодировании, методологические, управления конфигурацией и др.

Антипаттерны

- В объектно-ориентированном программировании
- В кодировании
- Методологические
- Управления конфигурацией
- Прочее

В ОПП

- Базовый класс-утилита

  Наследование функциональности из класса-утилиты вместо делегирования ему

- Anemic Domain Problem

  Боязнь размещать логику в объектах предметной области

В ООП

- Вызов предка

  Для реализации функциональности методу потомка приходится вызывать те же методы родителя
- Ошибка пустого подкласса

  Когда класс обладает различным поведением по сравнению с классом, который наследуется от него без изменений

В ООП

- Божественный объект

  Концентрация функциональности в одном классе/модуле/системе
- Объектная клоака

  Переиспользование объектов, находящихся в непригодном для переиспользования состоянии

В ООП

- Полтергейст

  Объекты, чье единственное предназначение – передавать данные другим объектам

- Проблема йо-йо

  Чрезмерная размытость сильно связанного кода по иерархии классов

В ООП

- Одиночество

  Неуместное использование паттерна синглтон
- Приватизация

  Сокрытие функциональности в приватной части, что затрудняет расширение в наследниках

В ООП

- Френд-зона

  Неуместное использование дружественных классов и функций
- Каша из интерфейсов

  Объединение нескольких интерфейсов, предварительно разделенных, в один

В ООП

- Висящие концы

  Интерфейс, большинство методов которого бессмысленные и являются «пустышками»
- Заглушка

  Попытка «натянуть» малоподходящий по смыслу интерфейс на класс

В кодировании

- Ненужная сложность
- Действие на расстоянии

  взаимодействие между широко разнесенными частями системы
- Накопить и запустить

  установка параметров подпрограмм в глобальных переменных

В кодировании

- Слепая вера

  недостаточная проверка корректности и полноты исправления ошибки или результата работы
- Лодочный якорь

  сохранение неиспользуемой части программы
- Активное ожидание

  потребление ресурсов в процессе ожидания запроса, путем выполнения проверок, чтений файлов и т.д., вместо асинхронного программирования

В кодировании

- Кэширование ошибки

  несбрасывание флага ошибки после ее обработки
- Воняющий подгузник

  сброс флага ошибки без ее обработки или передачи на уровень выше
- Проверка типа вместо интерфейса

  проверка на специфический тип, вместо требуемого определенного интерфейса

В кодировании

- Инерация кода

  избыточное ограничение системы из-за подрузамевания постоянной ее работы в других частях системы
- Кодирование путем исключения

  добавление нового кода для каждого нового особого случая
- Таинственный код

  использование аббревиатур/сокращений вместо логичных имен

В кодировании

- Жесткое кодирование

  внедрение предположение в слишком большое количество точек в системе
- Мягкое кодирование

  настраивается вообще все, что усложняет конфигурирование
- Поток лавы

  сохранение нежелательного кода из-за боязни последствий его удаления/исправления

В кодировании

- Волшебные числа

  использование числовых констант без объяснения их смысла
- Процедурный код

  когда стоило отказаться от ООП…
- Спагетти-код

  код с чрезмерно запутанным порядком выполнения

В кодировании

- Лазанья-код

  использование неоправданно большого числа уровней абстракции
- Равиоли-код

  объекты настолько склеены между собой, что невозможно провести рефакторинг
- Мыльный пузырь

  объект, инициализированный мусором (не инициализированный) слишком долго ведет себя как корректный

В кодировании

- Мьютексный ад

  внедрение слишком большого количества примитивов синхронизации в код

- (Мета-) шаблонный рак

  неадекватное использование шаблонов везде, где только получилось, а не где нужно

Методологические

- Использование паттернов

  значит, имеется недостаточный уровень абстракции
- Копирование-вставка

  нужно было делать более общий код
- Дефакторинг

  процесс уничтожения функциональности и замены ее документацией

Методологические

- Золотой молоток

  использование любимого решения везде, где только получилось
- Фактор невероятности

  гипотеза о том, что известная ошибка не проявится
- Преждевременная оптимизация

  оптимизация при недостаточной информации

Методологические

- Метод подбора

  софт разрабатывается путем небольших изменений
- Изобретение велосипеда

  создание с нуля того, для чего есть готовое решение
- Изобретение квадратного колеса

  создание плохого решения, когда уже есть хорошее готовое

Методологические

- Самоуничтожение

  мелкая ошибка приводит к фатальной
- Два тоннеля

  вынесение нового функционала в отдельное приложение
- Коммит-убийца

  внесение изменений без проверки влияния на другие части программы

Управления конфигурацией

- Ад зависимостей

  (DLL-hell в Windows)

Разрастание зависимостей до уровня, что раздельная установка/удаление программ становится если не невозможным, то крайне сложным

Прочее

- Дым и зеркала

  демонстрация того, как будут работать ненаписанные функции
- Раздувание ПО

  разрешение последующим версиям использовать все больше и больше ресурсов
- Функции для галочки

  превращение программы в «сборную солянку» плохо работающих и не связанных между собой функций