Программная инженерия

Анализ и проектирование программного обеспечения – 2

Вопросы

- Объектный подход
  - Общие положения
  - Абстрагирование, модульность, инкапсуляция
  - Типизация, параллелизм, сохраняемость
- Архитектура:
  - Зачем?
  - Как?
  - *Какая?*
  - *Как описать?*

Общие положения

Абстрагирование, модульность, инкапсуляция

ОБЪЕКТНЫЙ ПОДХОД

Парадигмы программирования

- Процедурный
- Объектно-ориентированный
- Ориентированный на правила

- Алгоритмы
- Классы и объекты
- Правила "если-то"

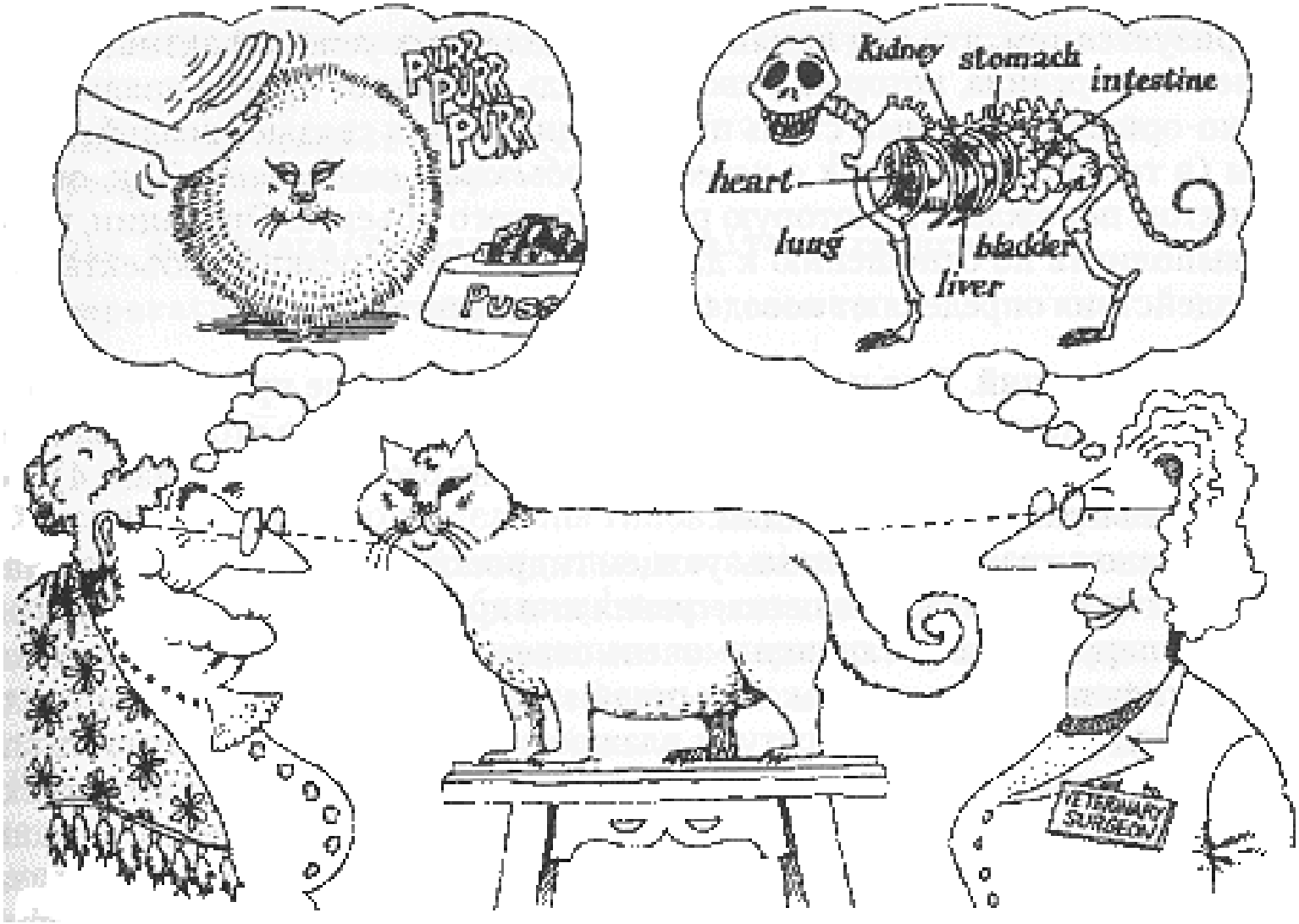
Концепции объектной модели

- Основные:
  - Абстрагирование
  - Инкапсуляция
  - Модульность
  - *Иерархия*
- Дополнительные:
  - Типизация
  - Параллелизм
  - Сохраняемость

Абстрагирование – 1/7

- Определения:
  - Ч. Хоар: Нахождение сходств между определенными объектами, ситуациями или процессами реального мира, и в принятие решений на основе этих сходств, отвлекаясь на время от имеющихся различий
  - М. Шоу: Упрощенное описание или изложение системы, при котором одни свойства и детали выделяются, а другие опускаются. Хорошей является такая абстракция, которая подчеркивает детали, существенные для рассмотрения и использования, и опускает те, которые на данный момент несущественны
- Резюмируя:
  - Выделяются существенные характеристики группы объектов, отличающие их от других объектов и/или групп (классификация)
  - Определяются границы группы (область применимости)

Абстрагирование – 2/7



Абстрагирование – 3/7

Абстрагирование – 4/7

- Абстрагирование концентрирует внимание на внешних особенностях объекта и позволяет отделить самые существенные особенности поведения от несущественных
- Принципы построения абстракций:
  - Барьер абстракции
    - Разделение смысла и реализации, основанное на принципе минимизации связей, когда интерфейс объекта содержит только существенные аспекты поведения и ничего больше
  - «Принцип наименьшего удивления»
    - Абстракция должна охватывать все поведение объекта, но не больше и не меньше, и не привносить сюрпризов или побочных эффектов, лежащих вне ее сферы применимости.

Абстрагирование: примеры абстракций – 5/7

- Абстракция сущности
  - Объект представляет собой полезную модель некой сущности в предметной области
- Абстракция поведения
  - Объект состоит из обобщенного множества операций
- Абстракция виртуальной машины
  - Объект группирует операции, которые либо вместе используются более высоким уровнем управления, либо сами используют некоторый набор операций более низкого уровня
- Произвольная абстракция
  - Объект включает в себя набор операций, не имеющих друг с другом ничего общего

Абстрагирование: понятия – 6/7

- Клиент – любой объект, использующий ресурсы другого объекта (сервера)
- Поведение объекта описывается:
  - Услугами, оказываемыми другим объектам (сервер)
  - Операциями, которые выполняются над другими объектами (клиент)
- Обычно объект одновременно и клиент и сервер
- Протокол
  - полный набор операций, которые клиент может осуществлять над другим объектом
  - правильный порядок, в котором эти операции вызываются

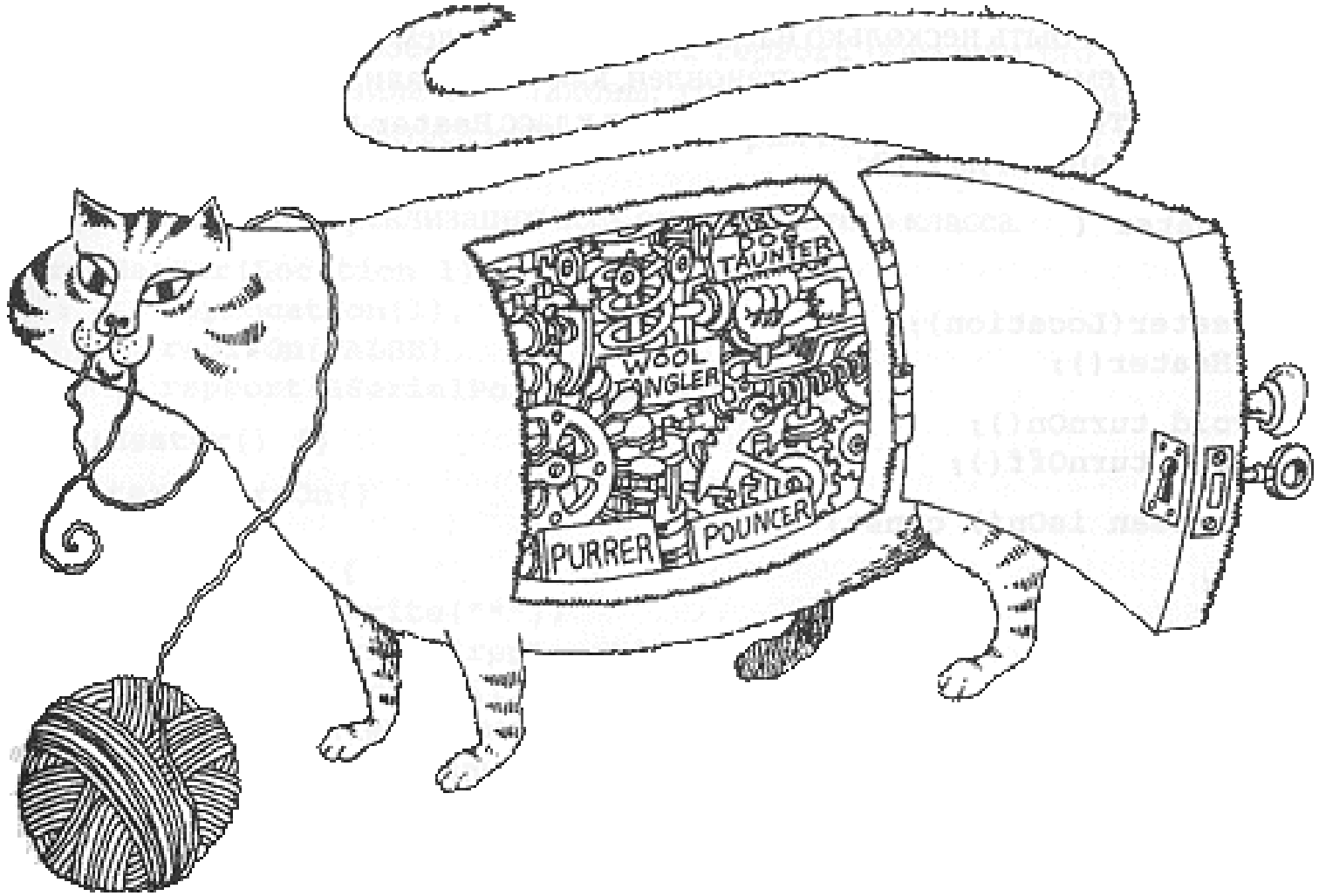
Абстрагирование: понятия – 7/7

- Операция описывается:
  - Входными данными
  - Выходными данными
  - Результатом ее выполнения
  - Условиями выполнения: предусловиями и постусловиями
- Операция может быть представлена как:
  - Вызов метода
  - Получение сообщения
  - Наступление условий

Инкапсуляция – 1/2

- Определения:
  - Упаковка данных и/или функций в единый компонент
  - Скрытие «подробностей» реализации
  - Разделение объекта на интерфейс и реализацию
- Резюмируя:
  - Отделение друг от друга элементов объекта, определяющих его устройство и поведение
  - Служит для того, чтобы изолировать поведение от реализации

Инкапсуляция – 2/2



Модульность – 1/4

- Определения:
  - Разделение на компоненты, которые компилируются отдельно, но связаны между собой
  - Мельчайший модуль – класс, обычно несколько классов
- Резюмируя:
  - Модульность это разбиение системы на цельные, слабо связанные между собой компоненты

Модульность – 2/4

- Интерфейсы модулей образуют более высокий уровень абстракции, чем интерфейсы классов
- Модуль обычно является единицей документирования, что вызывает стремление укрупнить их

Модульность – 3/4

- Типичные проблемы разбиения на модули:
  - Слишком крупные модули
  - Слишком мелкие модули
  - Слишком зависимые модули
- Полезные принципы разбиения:
  - Конечная цель декомпозиции – снижение затрат на разработку (= проектирование + кодирование + тестирование + …)
  - Перенос из реализации в интерфейс менее затратен, чем уменьшение интерфейса

Модульность – 4/4

Иерархия (повтор)

- Ранжирование, или упорядочение абстракций
- Упорядочивают модули (и классы)
- Наиболее важные виды иерархии:
  - Общее/частное (is-a)
    - Структура классов
  - Целое/часть (part-of)
    - Структура объектов
- Наследование может быть одиночным и множественным

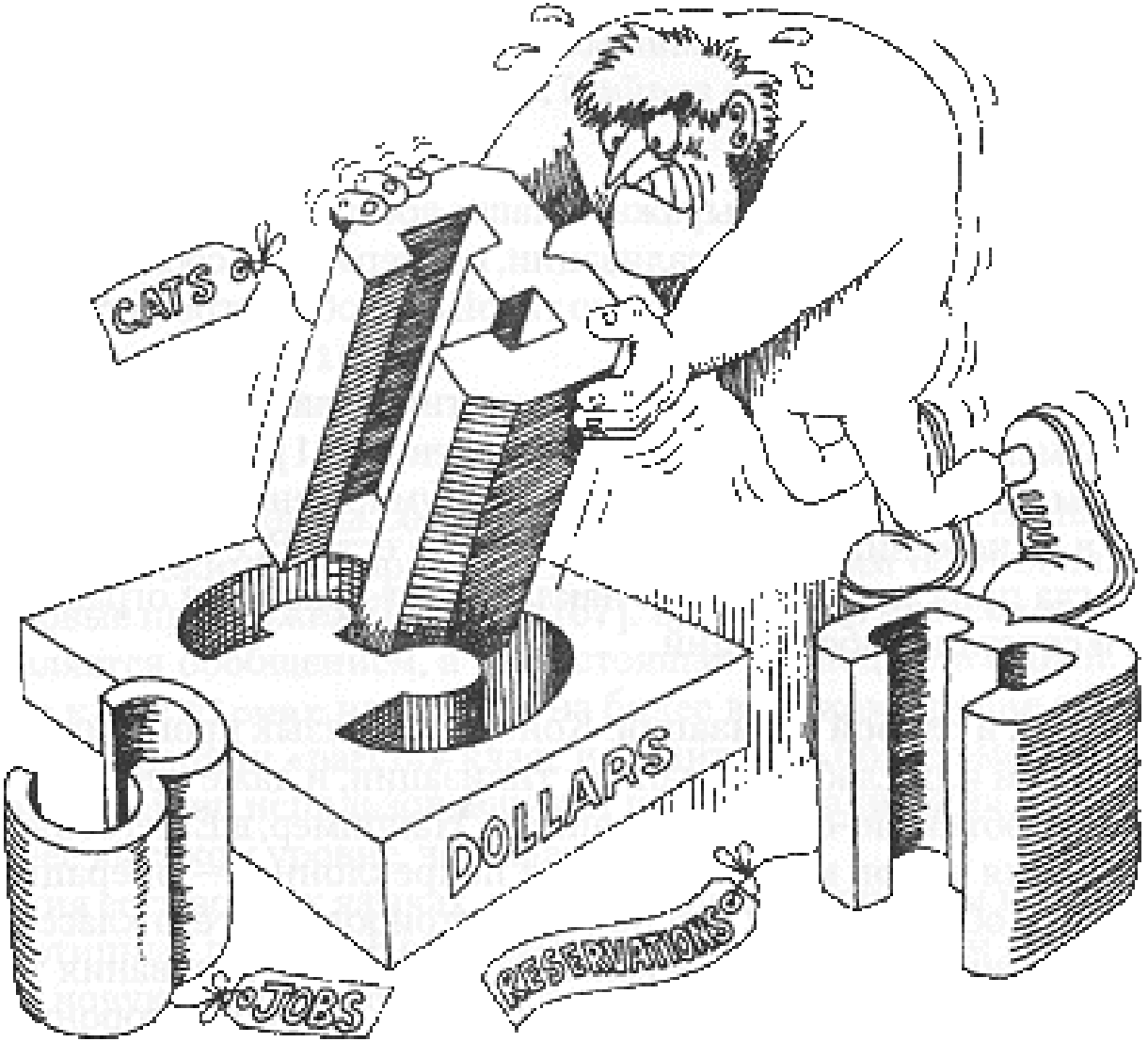
Типизация – 1/3

- Контроль типов – это правила использования объектов, не допускающие или ограничивающие взаимную замену объектов различных классов
- По силе:
  - Сильная (strong) – нет приведения типов
    - Python, Ada
  - Слабая (weak) – есть приведение типов
    - C, C++, Java

Типизация – 2/3

- По времени связывания:
  - Статическая (static, early binding)
    - Переменная связывается с типом в момент компиляции
    - Ada, C++, Pascal, Java
  - Динамическая (dynamic, late binding)
    - Переменная связывается с типом в момент присвоения значения
    - PHP, Python, JavaScript, Object Pascal

Типизация – 3/3



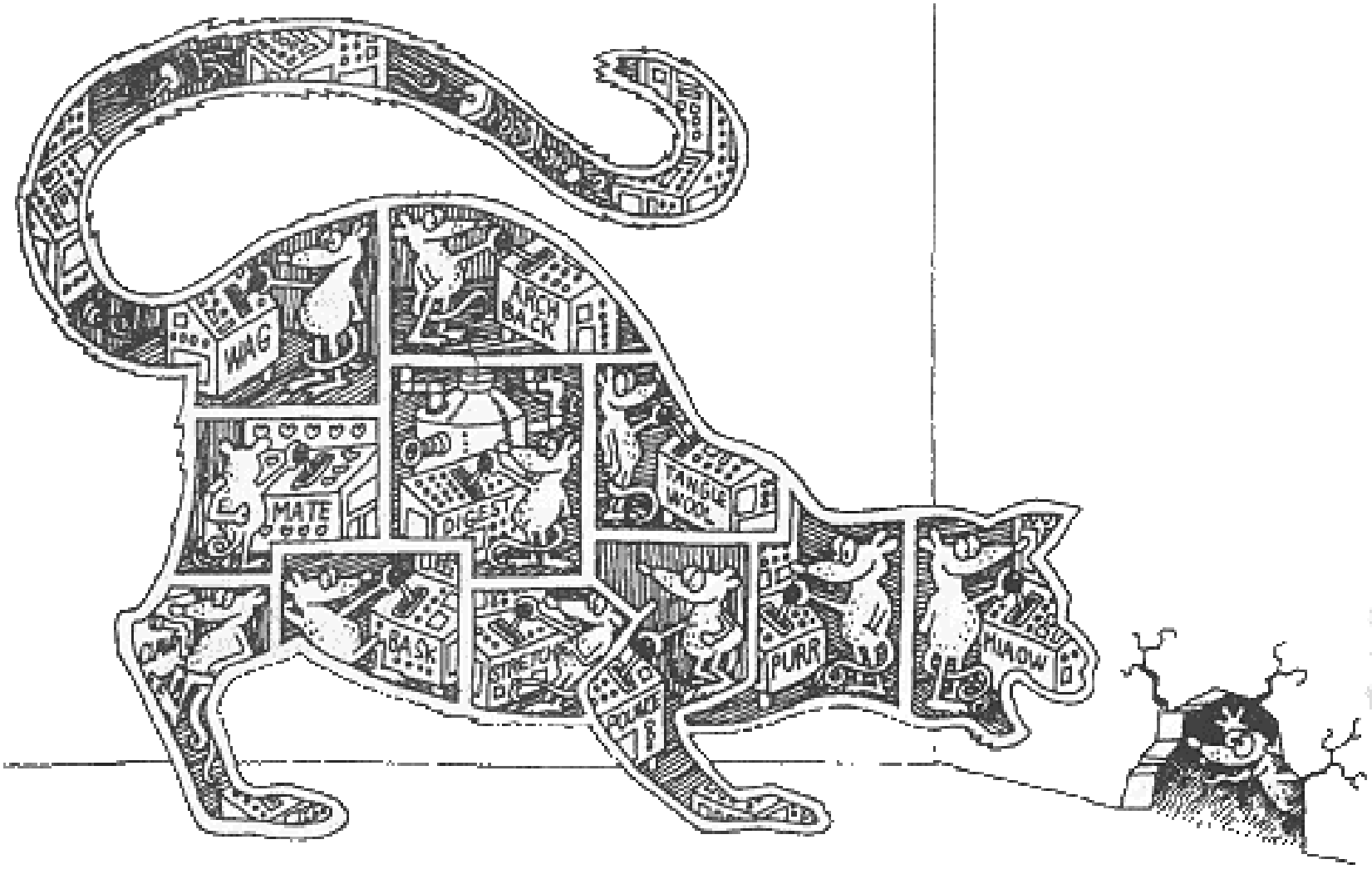
Параллелизм – 1/3

- Способность выполнять параллельно или «параллельно» несколько управляющих цепочек одновременно
- Свойство, отличающее активные объекты от пассивных
- Тяжеловесный (process)
  - Выполняются ОС независимо, могут иметь различное адресное пространство
- Легковесный (thread)
  - Выполняются ОС независимо, имеют общее адресное пространство

Параллелизм – 2/3

- Необходимость и преимущества:
  - Выполнение в сети
  - Более равномерное распределение нагрузки
  - Использование дополнительных мощностей
- Проблемы и сложности:
  - Обеспечение взаимодействия
    - Синхронизация
    - Обмен данными
  - Конкуренция за ресурсы

Параллелизм – 3/3



Существование объектов (persistence) – 1/2

- Способность сохранять состояние и класс объекта во времени (и пространстве)
- В распределенных системах объекты могут перемещаться по хостам (возможно, изменяя представление)

Существование объектов (persistence) – 2/2

- Локальные переменные методов
- Глобальные переменные процесса
- Разделяемые переменные межпроцессного взаимодействия
- Хранящиеся между сеансами выполнения
- Переходящие между версиями
  - Например, конфигурации
- Сохраняющиеся после исчезновения программы
  - Например, содержимое БД

Зачем

Как

ПОСТРОЕНИЕ АРХИТЕКТУРЫ

Архитектура

- Проектирование можно разбить на:
  - Разработка архитектуры
  - Детальная разработка проекта
- Архитектура – представление структуры и поведения системы на самом верхнем уровне
- Цели выбора архитектуры
  - Простота:
    - Понимания
    - Реализации
  - Расширение – добавление функциональности
  - Изменение – смена требований
  - Эффективность:
    - Скорость выполнения
    - Малый размер

Декомпозиция

- Цели
  - Связность внутри модуля – максимизация
  - Сцепление – минимизация
- Рекомендуемое число модулей одного уровня: 7±2
- Последовательное разбиение модулей верхнего уровня на более мелкие – рекурсивное проектирование

Все пропало

Ужас-ужас

ПРАКТИЧЕСКИЕ ЗАДАНИЯ