ФТК

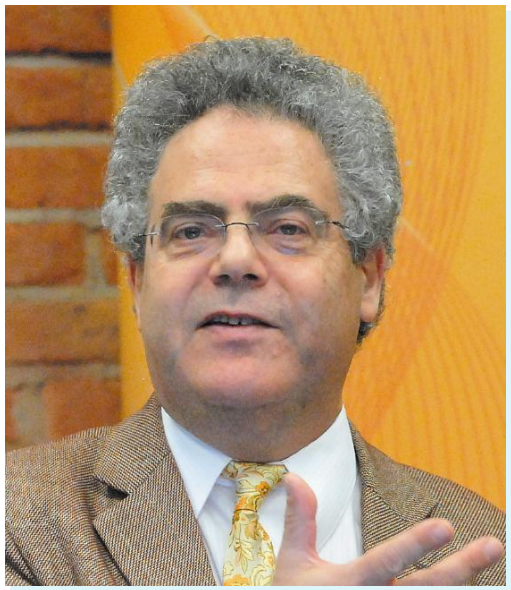
КСПТ
КОМПЬЮТЕРНЫЕ СИСТЕМЫ И ПРОГРАММНЫЕ ТЕХНОЛОГИИ

Технологии разработки программного обеспечения

Контрактное программирование

Контрактное программирование

- Контрактное программирование (программирование по контракту, Design by Contracts, DbC) – подход к созданию программ высокого качества
- Автор подхода – проф. Бертран Мейер
- Впервые введено в языке программирования Eiffel
- 1985 год



Контрактное программирование

- Основная идея – объединить программный код и спецификации
- Спецификации (контракты) встраиваются в программу
- В основе лежит логика Хоара
- Тройка Хоара: {P}C{Q}
  - P и Q – утверждения
  - C – часть программы

Контрактное программирование

- В терминах контрактного программирования метод (или функция) обязуется выполнить контракт:
  - Если на вход поступили данные удовлетворяющие входному условию контракта, то метод гарантирует соблюдение выходного условия
  - Если входные данные НЕ удовлетворяют первому условию, то ничего не гарантируется
  - При этом соблюдаются некоторые обобщенные условия

Контрактное программирование

- Входное условие называется предусловием (precondition)
- Выходное условие – постусловием (postcondition)
- Дополнительно обеспечивается поддержка инвариантов.
- Инвариант (invariant) – условие, которое не должно нарушаться из-за выполнения метода, т.е. должно гарантироваться выполнение инварианта до и после выполнения метода
  - Во время выполнения инвариант может быть временно нарушен!

Контрактное программирование

- Пример 1. Метод, вычисляющий корни квадратного уравнения $\mathbf{a \cdot x^2 + b \cdot x + c = 0}$
- Предусловие:
  - $a \neq 0$
- Постусловие:
  - Вариант 1:
    - $a \cdot x_1^2 + b \cdot x_1 + c = 0$;
    - $a \cdot x_2^2 + b \cdot x_2 + c = 0$
  - Вариант 2:
    - $x_1 + x_2 = -b/a$;
    - $x_1 \cdot x_2 = c/a$.

Контрактное программирование

- Пример 2. Снятие денег в банкомате
  - Balance – сумма на счете клиента
  - Cash – сумма, которую клиент хочет снять
  - Amount – денежный запас банкомата
- Предусловия:
  - Cash > 0;
  - Cash ≤ 10000;
- Постусловия:
  - Balance = old(Balance) – Cash;
  - Amount = old(Amount) – Cash;
- Инвариант:
  - Amount ≥ 0;
  - Balance ≥ - 1000;

Контракты и ООП

- В ООП языках программирования контракты могут являться часть объектной модели.
- При наследовании:
  - Предусловия у наследников могут быть ослаблены
  - Постусловия у наследников могут быть усилены
  - Инварианты у наследников могут быть усилены

Использование контрактов

- Проверка всех условий во время исполнения
  - В случае нарушения – останов
  - Возможность отключения проверок в релизах
- Статические проверки
  - Вывод с помощью дедуктивных методов
  - Проверка контрактов с помощью методов статического анализа

Использование контрактов

- Документирование
  - Предусловия + постусловия + интерфейс класса – документирование методов
  - Инварианты + интерфейс класса – документирование классов
- Тестирование
  - Контракты могут служить основой для автоматизированного тестирования:
    - Предусловия и инварианты – ограничения на генерируемые тесты
    - Постусловия и инварианты – основа для построения тестовых оракулов.

Мощность контрактов

- Контракты задаются с помощью выражений логики первого порядка с использованием:
  - целочисленной арифметики
  - вещественной арифметики
  - И.т.п
- Контракты – сугубо декларативное описание требовний

Языки программирования, поддерживающие контракты

- Языки со встроенной поддержкой
  - Eiffel
  - SPEC#
  - Fortress
  - D
  - …
- Языки с добавленной поддержкой
  - C# (Code Contracts)
  - Java (JML, Modern Jass, CoFoJa и т.п.)
  - ADA
  - C/C++ (DbC средствами препроцессора)
  - …