

Лекция 4

Методология разработки ПО (продолжение)

О.В. Федорова, доцент каф.
ПМИ

Методология императивного программирования

Методология императивного программирования – подход, характеризующийся принципом последовательного изменения состояния вычислителя пошаговым образом. При этом управление изменениями полностью определено и полностью контролируется.

Происхождение. Это исторически первая поддерживаемая аппаратно методология. Она ориентирована на фон-неймановскую модель, остававшуюся долгое время аппаратной архитектурой, получившей широкое практическое применение.

Методы и концепции. Метод изменения состояний – заключается в последовательном изменении состояний. Метод поддерживается концепцией алгоритма. Метод управления потоком исполнения – заключается в пошаговом контроле управления. Метод поддерживается концепцией потока исполнения.

Методология императивного программирования

Вычислительная модель. Императивное программирование основано на описании последовательного изменения состояний вычислителя. В качестве математической модели императивное программирование использует машину Тьюринга – Поста – абстрактное вычислительное устройство, предложенное на заре компьютерной эры для описания алгоритмов.

Класс задач. Методология наиболее пригодна для решения задач, в которых последовательное исполнение каких-либо команд является естественным. С ростом сложности задачи императивные программы становятся всё менее и менее читаемыми. Программирование и отладка больших программ, написанных на основе данной методологии, может затянуться на долгие годы.

Методология объектно-ориентированного программирования

Методология объектно-ориентированного программирования – подход, использующий объектную декомпозицию, при которой статическая структура системы описывается в терминах объектов и связей между ними, а поведение системы – в терминах обмена сообщениями между объектами.

Происхождение. На возникновение объектного мышления оказали влияние моделирование и представление данных, графические пользовательские интерфейсы и системное программирование (с понятием «процесс»). Моделирование реальных систем потребовало естественного описания сущностей – объектов и событий.

Методы и концепции. Метод объектной декомпозиции – заключается в выделении объектов и связей между ними. Метод поддерживается концепциями инкапсуляции, наследования и полиморфизма. Метод абстрактных типов данных – метод, лежащий в основе инкапсуляции. Метод поддерживается концепцией абстрагирования. Метод пересылки сообщений – заключается в описании поведения системы в терминах обмена сообщениями между объектами. Метод поддерживается концепцией сообщения.

Методология объектно-ориентированного программирования

Вычислительная модель. Модель чистого ООП поддерживает явно только одну операцию – посылка сообщения объекту. Сообщения могут иметь параметры, являющиеся объектами. Само сообщение также является объектом.

Класс задач. Методология является мощным средством для моделирования отношений между объектами практически в любой ПрО. Особенно удобно и легко в объектах выразить взаимодействие между различными элементами графического интерфейса пользователя.

Методология функционального программирования

Методология функционального программирования – подход, согласно которому в программах единственным действием является вызов функции, единственным способом разделения программы на части – введение имени для функции и задание для этого имени выражения, вычисляющего значение функции, а единственным правилом композиции – оператор суперпозиции функции.

Происхождение. Методология является одной из старейших. По происхождению она тесно связана с лямбда-исчислением, изобретенным ещё в начале 30-х гг. XX в. логиком А. Чёрчем. Для многих эта методология стала ассоциироваться с языком Lisp, созданным Дж. Маккарти в конце 50-х гг. XX в.

Методы и концепции. Метод аппликативности – заключается в том, что программа есть выражение, составленное из применения функций к аргументам. Программа состоит из совокупности определений функций, представляющих собой вызовы других функций и предложений, управляющих последовательностью вызовов. Метод поддерживается концепцией функции. Метод рекурсивного поведения – заключается в самоповторяющемся поведении, возвращающемся к самому себе. Метод поддерживается концепцией рекурсии..

Методология функционального программирования

Вычислительная модель. Функциональное программирование не содержит понятия времени: программы являются выражениями, а исполнение программ заключается в вычислении этих выражений. В качестве математической модели функциональное программирование использует лямбда-исчисление Чёрча.

Класс задач. Методология обычно применяется для решения тех задач, которых трудно сформулировать в терминах последовательных операций. В эту категорию попадают практически все задачи, связанные с искусственным интеллектом. Это такие задачи, как обработка естественного языка, экспертные консультирующие системы, проблемы зрительного восприятия, и многие другие.

Методология логического программирования

Методология логического программирования – подход, согласно которому программа содержит описание проблемы в терминах фактов и логических формул, а решение проблемы находится с помощью механизмов логического вывода.

Происхождение. Методология начинает свой отсчёт времени с конца 60-х гг. XX в., когда К. Грин предложил использовать резолюцию как основу логического программирования. А. Колмероз создал язык логического программирования Prolog в 1971 г. В основе логических языков обычно лежит какое-либо логическое исчисление с крупноблочными правилами вывода.

Методы и концепции. Метод единообразного применения механизма логического доказательства к программе. Метод поддерживается концепцией доказательства. Метод унификации – механизм сопоставления с образцом для перестройки структур данных. Метод поддерживается концепцией подстановки.

Методология логического программирования

Вычислительная модель. Логическое программирование – это программирование в терминах фактов и правил вывода.

Класс задач. Класс задач логического программирования практически совпадает с классом задач методологии функционального программирования.

Методология сентенциального программирования

Методология сентенциального программирования – подход, согласно которому суть программы состоит в перестройке чёткой и достаточно сложной структуры данных в рамках заданных условий.

Происхождение. Идея и основа подхода – нормальные алгоритмы Маркова. На основе подхода построены две модели, основанные на разных вариантах отождествления – конкретизации и унификации. Метод конкретизации считается основой методологии сентенциального программирования и языка Рефал, созданного В.Ф. Турчиным в 1986 г. Метод унификации совместно с формализмом математической логики стал основой методологии логического программирования и языка Prolog. Содержательно этот язык следует относить к сентенциальному программированию, так как целью его разработки послужили задачи математической лингвистики. Использование рекурсии вместо отождествления приводит уже к методологии функционального программирования.

Методы и концепции. Основной метод – метод отождествления. Метод поддерживается концепцией подстановки.

Методология сентенциального программирования

Вычислительная модель. Сентенциальное программирование – это программирование в форме обработки метавыражений.

Класс задач. Класс задач сентенциального программирования практически совпадает с классом задач методологии функционального и логического программирования, хотя в большей степени ориентирован на символьные преобразования.

Методология ограничительного программирования

Методология ограничительного программирования – подход, согласно которому в программе определяется тип данных решения, ПрО решения и ограничения на значение искомого решения; решение находится системой. В ряде работ встречаются следующие названия методологии: программирование в ограничениях и постановочное программирование.

Происхождение. Методология возникла в начале 80-х гг. XX в. как перспективная область исследований на пересечении символьных вычислений, искусственного интеллекта, исследования операций и интервальной арифметики.

Методы и концепции. Метод описательной модели вычислений заключается в том, что программа на языке программирования содержит описание понятий и задач. Метод поддерживается концепцией модели.

Методология ограничительного программирования

Вычислительная модель. Программирование в ограничениях – это программирование в терминах постановок задач.

Класс задач. Класс задач ограничительного программирования – задачи исследования операций и искусственного интеллекта. В таких задачах используется некоторое пространство решений, сужением которого достигается необходимый результат. Сужения можно естественным образом представить как ограничения.

Методология структурного [императивного] программирования

Методология структурного [императивного] программирования – подход, заключающийся в задании хорошей топологии императивных программ, в том числе отказе от использования глобальных данных и оператора безусловного перехода, разработке модулей с сильной связностью и слабым сцеплением. Подход базируется на двух основных принципах: Последовательная декомпозиция алгоритма решения задачи сверху вниз и Использование структурного кодирования. В противоположность методологии ООП данная методология известна под названием *методология процедурно-ориентированного программирования*.

Происхождение. Методология является важнейшим развитием императивной методологии. Создателем структурного подхода считается Эдсгер Вайб Дейкстра. Ему также принадлежит попытка соединить структурное программирование с методами доказательства программ.

Методология структурного [императивного] программирования

Методы и концепции. Метод алгоритмической декомпозиции сверху вниз – заключается в пошаговой детализации постановки задачи, начиная с наиболее общей задачи. Метод поддерживается концепцией алгоритма. Метод модульной организации частей программы – заключается в разбиении программы на специальные компоненты, называемые модулями. Метод поддерживается концепцией модуля. Метод структурного кодирования – заключается в использовании при кодировании трёх основных управляющих конструкций (следование, разветвление, повторение). Метод поддерживается концепцией управления.

Класс задач. Класс задач для данной методологии соответствует классу задач для императивной методологии. При этом удаётся разрабатывать более сложные программы, поскольку их легко воспринимать и анализировать.

Методология [императивного] параллельного программирования

Методология [императивного] параллельного программирования – подход, в котором предлагается использование явных конструкций для параллельного исполнения выбранных фрагментов программ.

Происхождение. Анализ эффективности решений сложных вычислительных задач показал, что скорость вычислений можно значительно улучшить, если использовать не одно, а несколько устройств одновременно. Создание многопроцессорных систем привело к крупным исследованиям в этой области. Ещё одна причина возникновения этой методологии связана с появлением сложных программ, требующих поддержки явного параллелизма (например, операционных систем).

Методология [императивного] параллельного программирования

Методы и концепции. Метод синхронизации исполняемого кода – заключается в использовании специальных атомических операций для осуществления взаимодействия между одновременно исполняемыми фрагментами кода. Метод поддерживается концепцией примитивов синхронизации.

Класс задач. Методология может очень эффективно применяться для обработки больших однородных массивов данных. Такие массивы часто встречаются в реализации вычислительных и статистических методов. Кроме этого, данная методология успешно применяется при моделировании, в операционных системах и системах реального времени.

Методология логического параллельного программирования

Методология логического параллельного программирования – подход, обобщающий логическую методологию на параллельное выполнение доказательств.

Происхождение. Язык Concurrent Prolog, предложенный Эхудом Шапиро развивает логический подход к абстрактным спецификациям и включает несколько идей Дейкстры (в том числе, идею охраняемого предложения). При этом ПрО описывается как формальная теория в некотором логико-математическом языке.