Лекция 3
Методология разработки ПО

О.В. Федорова, доцент каф. ПМИ

Методологический подход

- *Конкретная методология (методологический подход) представляет собой набор методов, объединённых некоторым общим принципом и поддерживаемых соответствующим набором концепций.*
- В узком смысле методология представляет собой определённый методологический подход. Именно **методология определяет, какие языки и системы программирования будут применяться для разработки ПО и какой технологический подход будет при этом использован.**

Методология разработки ПО

- *Методология разработки ПО* (методологический подход) – это объединённая единым философским подходом определённая совокупность методов, применяемых в процессе разработки ПО.

С каждой методологией можно связать характерные для неё атрибуты:

1. ***Философский подход (основной принцип),*** являющийся простым для формулирования и определяющий основной источник эффективности методологии.
2. Согласованное, связное ***множество методов,*** через которые реализуется данная методология.
3. ***Концепции*** (понятия, идеи), поддерживающие методы и позволяющие более точно их определить.

В рамках заданной методологии разработки разрабатываются методики, которые применяются в подходах разработки.

Парадигма программирования

- Любая методология создаётся на основе уже накопленных в ПО эмпирических фактов и практических результатов. Для методологий разработки ПО такими фактами и результатами являются уже существующие языки программирования.
- Когда методология применяется во время кодирования, очень часто её называют *парадигмой программирования* – способом мышления и программирования, не связанным с конкретным языком программирования.

Ядро методологии

- Один из подходов к классификации методологий заключается в том, что существует некоторое *ядро методологии со своими методами, которое уточняется некоторыми дополнительными особенностями – спецификами*.
- Этот подход напоминает принцип словообразования в русском языке – есть корень, к которому добавляются аффиксы (приставки, суффиксы и окончания), уточняющие смысл слова.

Ядро методологий определяется способом описания алгоритмов

- Выделяют следующие основные ядра методологий: методология императивного программирования,
- методология объектно-ориентированного программирования,
- методология функционального программирования,
- методология логического программирования,
- методология ограничительного программирования.

Все перечисленные методологии находятся в диапазоне между двумя понятиями информатики – алгоритма и модели. В данном случае они перечислены в порядке уменьшения связи методологии с понятием алгоритм и

Топология

- Каждое из «корней»-ядер может получить «приставку», определяемую некоторой *топологией* – структуру (информационных, управляющих или логических) узлов и связей программ, которая может быть хорошей или плохой.
- *Топологии определяются совокупностью многочисленных факторов, связанных с абстракциями данных, управления и модульности.* Например, к хорошей топологии приводит отказ от использования глобальных данных и оператора безусловного перехода, правильная модульная организация.

Пример

- Если в императивной методологии придерживаться методов структурного представления (дающих хорошую топологию), то мы получим хорошо известную методологию: структурного императивного программирования, которая более известна под её кратким именем – методология структурного программирования.
- Следует отметить, что успех методологии объектно-ориентированного программирования изначально определила её хорошая топология, базирующаяся на абстрактных типах данных.

Реализация

Каждое из «корней»-ядер может получить «суффикс», определяющий некоторую *реализацию* – организацию аппаратной поддержки – данной методологии.

На данный момент известными являются две реализации – **централизованная и распределённая**, на основе которых строятся соответственно **последовательные и параллельная методологии**.

К последовательным относят обычно фактически все ядра методологий, а к параллельным – соответствующие им модификации.

Примеры

Примеры параллельных методологий:

- методология императивного параллельного программирования, её краткое название – методология параллельного программирования,
- методология логического параллельного программирования.
- Для методологии объектно-ориентированного программирования параллельность неявно используется уже на уровне методов и концепций.
- Смешанные методологии основываются на объединении ряда методов нескольких (обычно родственных) методологий.

Наиболее часто объединяются методологии функционального и логического программирования,

рассматриваемые как *методология декларативного программирования* в противоположность *методологии директивного программирования*, под которым понимают методологию императивного программирования и его модификации, а также методологию объектно-ориентированного программирования.

- Кроме того, в рамках декларативного программирования часто выделяют методологию сентенциального программирования как самостоятельный подход.
- Проводятся исследования в области объединения других методологий, в частности объектно-ориентированного и логического программирования. Ряд исследований посвящён вопросам унификации методологий программирования.
- В эту классификацию не вошли многие существующие методологии.

К известным, но редко выделяемым явно, относят следующие методологии:

- методология событийного программирования – подход, использующий взаимодействия через события при функционировании системы.
- методология автоматного программирования – подход, представляющий функционирование системы в виде конечного автомата.

К мало известным (в настоящее время) относят следующие методологии:

- методология программирования, управляемого потоком данных, – подход, заключающийся в том, что операции срабатывают не последовательно, а в зависимости от готовности данных;
- методология доступ-ориентированного программирования – подход, в котором функции связываются с переменными таким образом, что при доступе к переменной автоматически будет вызываться соответствующая функция;
- методология нейросетевого программирования – подход, заключающийся в том, что на основе знаний от экспертов создаётся программа на нейронном языке программирования, которая затем компилируется в эквивалентную нейронную сеть из аналоговых нейронов.

Новые методологии

Из приведённого далеко неполного списка методологий видно, что новые методологии основываются на применении к разработке ПО идей из самых разных областей научно-технической деятельности:

- операционные системы (прерывания),
- автоматическое управление (теория конечных автоматов),
- вычислительные системы (системы, управляемые потоком данных),
- программирование (доступ-ориентированное программирование),
- оптимизация (нейронные сети) и т.д.

Три точки зрения на происхождение методологий

- В настоящее время выделяют три точки зрения на происхождение методологий: *практическая, алгоритмическая и структурно-языковая.*
- Первая точка зрения – ***независимое происхождение на основе практического опыта программистов*****.** Очень многие методологии имеют автора-создателя или основателя научной школы, исследующей данную методологию.
- В этом случае методологию можно рассматривать как концентрацию практического опыта программирования. Эта точка зрения позволяет оценить сложность создания достаточно подробной классификации методологий, которая охватывала бы большую часть существующих и развивающихся методологий, чрезвычайно разнородных по используемым в них методам и принципам.

Алгоритмическое происхождение вытекает из следующего утверждения:

Теория алгоритмов и логика – родители программирования.

Выделяют следующие ***4 главные модели алгоритма:***

1. Абстрактные вычислительные машины Тьюринга и Поста. Они определяют методологии императивного, автоматного и событийного программирования.
2. Рекурсивные функции Гильберта и Аккермана. От них унаследовала свои идеи и конструкции методология структурного программирования.
3. Комбинаторная логика Шейнфинкеля и Карри и её современное представление – лямбда-исчисление Чёрча. Эти идеи активно развиваются в методологии функционального программирования.
4. Нормальные алгорифмы Маркова. Модель послужила основой методологий логического программирования и сентенциального программирования.

Отображение структур языка

Ещё одно объяснение берёт за основу понятие отображения структур языка. Согласно этой точке зрения сущность языка определяют три его составные части:

1. *Структура данных* (Д) – представление данных (и результатов).
2. *Структура управления* (У) – преобразование исходных данных в результат.
3. *Логическая структура* (Л) – определение преобразования задачи в алгоритм.

Каждая из трёх структур языка моделирования (ПрО) может быть отображена на любую из структур языка программирования. При этом каждое такое отображение определяет либо некоторую методологию, либо, по крайней мере, достаточно серьёзный метод.

В итоге получаются следующие 9 отображений:

1. **Д → Д:** представляет процесс укрупнения данных и операций над ними и приводит к методам модульности и абстрактных типов данных.
2. **У → У:** связано с понижением уровня структуры управления языка моделирования, ведёт к идее методологии структурного программирования.
3. **Л → Л:** лежит в основе методологии логического программирования.
4. **Д → У:** активизирует пассивные данные, преобразуя их в активные процессы; лежит в основе методологий функционального и сентенциального программирования; в значительной степени определяет методологию объектно-ориентированного программирования.
5. **Д → Л:** даёт возможность по совокупности операций построить логическую структуру и определяет методологию ограничительного программирования.
6. **У → Д:** лежит в основе методов интерпретации; определяет методологию доступ-ориентированного программирования.
7. **У → Л:** лежит в основе методов расшифровки смысла задачи.
8. **Л → Д:** может быть связано с типизацией данных и определяет метод развитой системы типов и приведений; также может быть связано с интерпретаторами, реализующими языки с развитой логической структурой.
9. **Л → У:** может быть использовано в системах структурного синтеза.