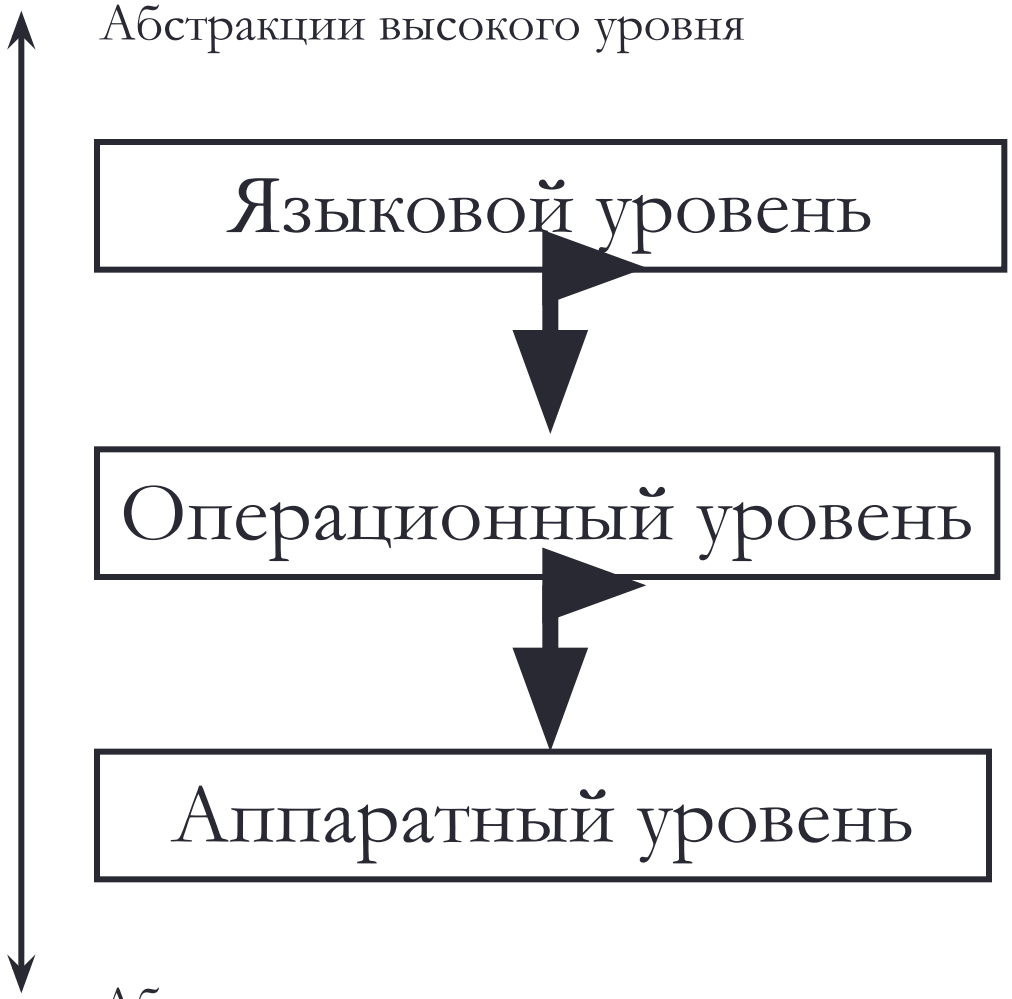
ЛЕКЦИИ 3 – 4

Методологии программирования

Трехуровневая организация ЭВМ



Трехуровневая организация ЭВМ

Платформа – видимые программисту средства поддержки программного продукта каждым из этих уровней абстракции.

Типы платформ:

- Аппаратная платформа – тип аппаратной архитектуры, на котором может быть установлен программный продукт.
- Операционная платформа – тип операционной системы и системного интерфейса, с которым может работать программный продукт.
- Языковая платформа – язык программирования и интерфейс прикладного программирования (библиотеки), на котором может быть реализован программный продукт.

Основные понятия и определения

Методология – совокупность методов, применяемых в жизненном цикле и объединенных общим философским подходом.

С каждой методологией можно связать некоторые характерные для нее атрибуты:

- Философский подход (или основной принцип), являющийся простым для формулирования и определяющий основной источник эффективности методологии.
- Согласованное, связанное множество моделей методов, через которые реализуется данная методология.
- Концепции (понятия, замыслы), поддерживающие методы и позволяющие более точно их определить.

Основные понятия и определения

Синтаксис – правила образования текстов.

Для описания синтаксиса наиболее часто используется система обозначений Бэкуса-Наура:

- Метапеременные представляют собой слова или группы слов, заключенных в угловые скобки ‘< >’.
- Под значением метапеременной понимается некоторая конечная последовательность основных символов языка, из которых, в конечном счете состоят программы.
- Символ ‘::=’ означает «определяется как»
- Символ ‘|’ означает «или»
- Произвольное количество повторений (в том числе и ноль) некоторой конструкции заключается в символы ‘{ }’.
- Символы, указанные в ‘[ ]’, являются необязательными.

Основные понятия и определения

Семантика – правила истолкования текста тем, кому они адресованы.

В отличии от синтаксиса семантика обычно описывается словесно, на естественном языке.

Одним из формальных подходов к описанию семантики является операционный подход. В нем семантика описывается в терминах некоторого вычислителя (машины), например абстрактной (воображаемой, виртуальной) машины. В процессе работы вычислителя меняется состояние программы, образуемое значениями ее переменных. Существует некоторое начальное состояние, а также результат исполнения программы.

Алгоритмическое происхождение некоторых методологий

Четыре основные методологии:

- Первая модель – абстрактная вычислительная машина Тьюринга. Она определяет методологию императивного программирования.
- Вторая модель – рекурсивные функции Гильберта и Аккермана. От них унаследовала свои идеи и конструкции методология структурного программирования.
- Третья модель – лямбда-исчисление Черча, Шейнфинкеля и Карри. Эти идеи активно развиваются в методологии функционального программирования.
- Четвертая модель – нормальные алгорифмы Маркова. Считается что эта модель послужила основой логического программирования.

Отображение структур языка

Сущность языка определяют три его составные части:

- Структура данных.
- Структура управления.
- Логика.

Отображение структур языка

- Данные → Данные. Отображение представляет собой процесс укрупнения данных и операций над ними и приводит к методам модульности и абстрактных типов данных.
- Управление → Управление. Отображение, связанное с понижением уровня структуры управления языка моделирования, ведет к идее методологии структурного программирования.
- Логика → Логика. Отображение лежит в основе методологии логического программирования.
- Данные → Управление. Отображение активизирует пассивные данные, преобразуя их в активные процессы и лежит в основе методологии функционального программирования, а также определяет методологию объектно-ориентированного программирования.

Отображение структур языка

- Данные → Логика. Отображение дает возможность по совокупности операций построить логическую структуру и определяет методологию программирования на ограничениях.
- Управление → Данные. Отображение лежит в основе методов интерпретации и определяет методологию доступ-ориентированного программирования.
- Управление → Логика. Отображение лежит в основе методов расшифровки смысла задачи.
- Логика → Данные. Отображение может быть связано с типизацией данных и определяет метод развитой системы типов и приведений.
- Логика → Управление. Отображение может быть использовано в системах структурного синтеза.

Виды методологий

- Методологий императивного программирования.
- Методология объектно-ориентированного программирования.
- Методология функционального программирования.
- Методология логического программирования.
- Методология программирования на ограничениях.

Смешанные методологии

- Методология структурного императивного программирования (методология структурного программирования).
- Методология императивного параллельного программирования (методология параллельного программирования).
- Методология логического параллельного программирования.

Методология императивного программирования

Методология императивного программирования – подход, характеризующийся принципом последовательного изменения состояния вычислителя пошаговым образом.

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

Основное понятие – оператор.

Две группы операторов:

1. Атомарные операторы, у которых никакая часть не является самостоятельной.

2. Структурные операторы, объединяющие другие операторы в новый, более крупный оператор (составной, выбора, цикла и т.п.)

Методология императивного программирования

<оператор> ::= <простой оператор> | <структурный оператор>
<простой оператор> ::= <оператор присваивания> |
<оператор вызова> | <оператор возврата>
<структурный оператор> ::= <составной оператор> |
<оператор ветвления> | <оператор цикла>
<оператор присваивания> ::= <переменная> := <выражение>
<оператор вызова> ::= <имя подпрограммы> ([<параметры>]);
<оператор возврата> ::= return [<выражение>];
<составной оператор> ::= begin <оператор> {; <оператор>} end
<оператор выбора> ::= if <выражение> then <оператор> |
if <выражение> then <оператор> else <оператор>
<оператор цикла> ::= while <выражение> do <оператор>

Методология объектно-ориентированного программирования

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

Методы и концепции:

- Метод объектно-ориентированной декомпозиции заключается в выделении объектов и связей между ними. Метод поддерживается концепциями инкапсуляции, наследования и полиморфизма.
- Метод абстрактных типов данных – это 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

- Инкапсуляция – это сокрытие информации и комбинирование данных и функций, которые аналогичны абстрактным типам данных.

- Наследование – построение иерархии порожденных объектов с возможностью для каждого такого объекта, относящегося к иерархии, доступа к методам и данным всех порождающих объектов.

- Полиморфизм – присваивание действию одного имени, которое затем разделяется вверх и вниз по иерархии объектов, причем каждый объект иерархии выполняет это действие способом, подходящим именно ему.

Методология объектно-ориентированного программирования

Свойства экторной модели параллельных вычислений Хьюита:

- объектом является процесс, который может иметь различные внутренние состояния. При получении сообщения объект становится активным;
- извне внутренние состояния объекта может быть изменено только посредством передачи ему сообщения, специфицирующего выполняемую объектом операцию;
- во время работы объект может обмениваться сообщениями с другими объектами.

Методология объектно-ориентированного программирования

Языки объектно-ориентированного программирования можно разделить на три группы:

- Чистые языки, в наиболее классическом виде, поддерживающие объектно-ориентированную методологию (Simula, Smalltalk).
- Гибридные языки, которые появились в результате внедрения объектно-ориентированных конструкций в популярный императивный язык программирования (C++, Object pascal, Ada 95).
- Урезанные (очищенные) языки, которые появились в результате удаления из гибридных языков наиболее опасных и ненужных с объектно-ориентированной точки зрения конструкций (Java, C#).

Методология функционального программирования

Методология функционального программирования – способ составления программ, в котором единственным действием является вызов функции, единственным способом расчленения программ на части – введение имени функции и задания для этого имени выражения, вычисляющего значения функции, а единственным правилом композиции – оператор суперпозиции функции.

Методы и концепции:

- Метод аппликативности заключается в том, что программа есть выражение, составленное из применения функций к аргументам. Метод поддерживается концепцией функции.
- Метод рекурсивного поведения заключается в самоповторяющемся поведении, возвращающегося к самому себе. Метод поддерживается концепцией рекурсии.
- Метод настраиваемости заключается в том, что можно легко порождать новые программные объекты по образцу, как значения соответствующих выражений.

Методология логического программирования

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

Методы и концепции:

- Метод единообразия заключается в одинаковом применении механизма логического доказательства ко всей программе.
- Метод унификации – это механизм сопоставления с образцом для создания и декомпозиции структур данных.

Методология программирования на ограничениях

Методология программирования на ограничениях – это подход, при котором в программе определяется тип данных решения, предметная область решения и ограничения на значение искомого решения.

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 структурного императивного программирования

Методология структурного императивного программирования – подход, заключающийся в задании хорошей топологии императивных программ, ориентированной на сокращение количества общих затрат на разработку программного обеспечения.

Принципы:

- последовательная декомпозиция алгоритма решения задачи сверху вниз;
- использование структурного кодирования.

Методы и концепции:

- Метод алгоритмической композиции сверху вниз заключается в пошаговой детализации постановки задачи, начиная с наиболее общей задачи. 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 трех основных управляющих конструкций. Метод поддерживается концепцией управления.

Оператор goto

Точка зрения	Заменители goto	Использование goto
Ортодоксальная (Эдсгер Вибе Дейкстра)	Нет	Нет
Только заменители goto	Да	Нет
Критики Дейкстры (только goto)	Нет	Да
И заменители, и goto	Да	Да

Методология императивного параллельного программирования

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

Методы и концепции:

- Метод синхронизации исполняемого кода заключается в использовании специальных атомических (атомарных) операций для осуществления взаимодействия между одновременно исполняемыми фрагментами кода. Метод поддерживается концепцией примитивов синхронизации.

Методология императивного параллельного программирования

Уровни параллелизма:

- параллелизм на уровне микрокоманд;
- параллелизм на уровне операторов (кроме циклов);
- параллелизм на уровне циклов и итераций;
- параллелизм на уровне подпрограмм (процедур и функций);
- параллелизм на уровне потоков управления;
- параллелизм на уровне процессов;
- параллелизм на уровне приложений.

Синтаксис параллельного программирования

<процесс> ::= <простой процесс> | <структурный процесс>
<простой процесс> ::= <послать значение> | <принять значение> | <процесс вычислительной модели>
<структурный процесс> ::= <последовательный процесс> | <параллельный процесс>
<послать значение> ::= <канал связи> << <выражение>;
<принять значение> ::= <канал связи> >> <выражение>;
<процесс вычислительной модели> ::=
<любые вычислительные действия, не зависящие от наличия параллелизации>
<последовательный процесс> ::= seq <процесс> {<процесс>}* end
<параллельный процесс> ::= par <процесс> {<процесс>}* end

Языковые подходы к параллельному программированию

- Программирование на параллельном языке программирования
- Программирование на широко распространенном языке программирования, который расширен языковыми распараллеливающими конструкциями.
- Программирование с использованием дополнительных указаний компилятору на уровне языка прагм.
- Программирование на распространенном языке программирования с использованием коммуникационных библиотек и интерфейсов для организации межпроцессорного взаимодействия.
- Применение средств автоматического распараллеливания последовательных программ такими инструментами, как компиляторы.

Прочие методологии

- Методология автоматного программирования – это подход, предполагающий использование аппарата конечных автоматов.
- Методология программирования, управляемая потоками данных, - это подход, заключающийся в том, что операции срабатывают не последовательно, а в зависимости от готовности данных.
- Методология программирования, управляемого событиями, - подход, заключающийся в структуризации программного кода, основанного на идее наличия предопределенного множества именованных событий.
- Методология доступ-ориентированного программирования – подход, в котором функции с переменными связываются таким образом, что при доступе к переменной процедура будет вызываться автоматически.