Объектно-ориентированное программирование

Лекций: 34 часа

Лаб. работ: 36 часов (8 работ)

Курсовая работа

Самостоятельная работа

Седьмой семестр

Восьмой семестр

Сайт дисциплины: http://vt.cs.nstu.ru/~malyavko/OOP/index.html

E-mail: translab@ngs.ru

Малявко Александр Антонович

Литература

1. Васильев А.Н. Java. Объектно-ориентированное программирование. – СПб.: Питер, 2011.
2. Монахов В. В. Язык программирования Java и среда Netbeans. – СПб.: БХВ-Петербург, 2011.
3. Хорев П. Б. Объектно-ориентированное программирование: учебное пособие по направлению "Информатика и вычислительная техника". М. : Академия , 2011.
4. Хабибуллин И.Ш. Самоучитель JAVA. 3-е изд. перераб. и доп. – СПб.: БХВ-Петербург, 2008.
5. Шилдт Г. Java. Полное руководство. М.: Вильямс, 2012
6. Флэнаган Д. Java. Справочник. СПб.-М.: Символ, 2004.
7. Эккель Б. Философия Java. Библ. программиста. СПб.: Питер, 2011.

Парадигмы программирования

Парадигма программирования — это система идей, понятий и средств, определяющих стиль написания компьютерных программ, т. е. по сути — это стиль мышления программиста.

Перечень известных парадигм программирования (1)

- Агентно-ориентированная (ООП с ограничениями)
- Компонентно-ориентированная (– "" –)
- Конкатенативная (PostScript, Forth, Joy, …)
- Декларативная (контрастирует с Императивной)
  - Ограничениями (Jaffar, Lassez, GNU Prolog, …)
  - Функциональностью (Lisp, Erlang, ML, Haskell, …)
  - Потоком данных
    - Таблично-ориентированная (электронные таблицы: Excel, …)
    - Реактивная (AWK, *HDL, Verylog, …)
  - Логическим выводом (Prolog, Mercury, Oz, …)

Список парадигмы программирования (2)

- Событийно-ориентированная (OS Windows, пользовательские интерфейсы, сервера, …)
  - Сервис-ориентированная (технологии SOAP, CORBA, DCOM, Web-services, …)
- Комбина́торная (разновидность функциональной)
- Императивная (контрастирует с Декларативной)
  - Процедурная
- Предметно-ориентированная (*DSL, XML, XSL, …)
- Метапрограммирование (программы, порождающие другие программы)
  - Автоматизация разработки ПО или решение задач без программирования (человеко-машинные языки, ИИ)
  - Обобщённое программирование (шаблоны, generics, …)
  - Рефлексивно-ориентированная (расширения ООП: самоверификация, самомодификация, …; Objective-C, MOO, Common Lisp, …)

Список парадигм программирования (3)

- Структурная (иерархия блоков, запрет GOTO)
  - Модульная (процедуры/функции, библиотеки)
  - Рекурсивная (частный случай модульности)
  - Объектно-ориентированная (стандарт написания хорошо структурированных модульных программ)
    - Класс-ориентированная (статическое, определяемое полностью в процессе разработки, наследование: C++, Java, C#, …)
    - Прототип-ориентированная (динамическое наследование от прототипа, изменяющегося в процессе функционирования приложения: Self, JavaScript, Lua, …)
- Параллельная (масса параллельных версий популярных языков всех прочих парадигм)

Основные модели программирования

- Функциональное программирование
- Логическое программирование
- Императивное программирование
  - Структурное программирование
    - Объектно-ориентированное программирование

Предшественники ООП

- Процедурное программирование
- Структурное программирование
- Модульное программирование
- Событийно-ориентированное программирование
- Компонентное программирование

Первое применение принципов ООП: язык Симула (1967 год)

Первый действительно и полностью объектно-ориентированный язык: Smalltalk (1971 год)

Что такое ООП?

ООП имеет уже более чем сорокалетнюю историю, но, несмотря на это, до сих пор не существует чёткого общепринятого определения данной технологии. Основные принципы, заложенные в первые объектные языки и системы, подверглись существенному изменению (или искажению) и дополнению при многочисленных реализациях последующего времени.

Кроме того, примерно с середины 1980-х годов термин «объектно-ориентированный» стал модным, в результате с ним произошло то же самое, что несколько раньше произошло с термином «структурный» — его стали искусственно «прикреплять» к любым новым разработкам, чтобы обеспечить им привлекательность.

Так что же такое ООП?

По мнению Алана Кея, создателя языка Smalltalk, которого считают одним из «отцов-основателей» ООП, объектно-ориентированный подход заключается в согласованном применении следующего набора основных принципов:

- Всё является объектом.
- Вычисления осуществляются путём взаимодействия (обмена данными) между объектами, при котором один объект требует, чтобы другой объект выполнил некоторое действие.
- Объекты взаимодействуют, посылая и получая сообщения. Сообщение — это запрос на выполнение действия, дополненный набором аргументов, которые могут понадобиться при выполнении действия.

Объекты

- Каждый объект имеет независимую память, которая содержит данные примитивных типов и другие объекты.
- Каждый объект является представителем (экземпляром) класса, который определяет общие свойства объектов.
- В классе задаётся поведение (функциональность) объекта.

Объекты и классы:

- Тем самым все объекты, которые являются экземплярами одного класса, могут выполнять похожие действия.
- Классы организованы в единую древовидную структуру с общим корнем, называемую иерархией наследования.
- Память и поведение, связанное с экземплярами определённого класса, автоматически доступны любому классу, расположенному ниже в иерархическом дереве.

ОО-программа – это

набор объектов, имеющих состояние и поведение.

Объекты взаимодействуют посредством посылки сообщений или путем вызова методов.

Программа в целом — это тоже объект, для выполнения своих функций она в определенной последовательности обращается к входящим в неё объектам.

Каждый из внутренних объектов программы , в свою очередь, при получении запроса извне выполняет требуемое действие либо путём обращения к другим объектам (возможно – входящим в его состав), либо возвращая сформированный им самим результат.

Основные термины ООП

- Абстракция — это способ выделить набор значимых характеристик объекта, исключая из рассмотрения незначимые (для данного уровня абстракции).
- Класс — описываемая на языке исходного кода модель ещё не существующей сущности (объекта). Фактически он описывает устройство объекта, являясь своего рода чертежом.

  Говорят, что объект — это **экземпляр** класса. Обычно классы разрабатывают таким образом, чтобы их объекты соответствовали реальным объектам предметной области.
- Прототип — это объект-образец, по образу и подобию которого создаются другие объекты. Объекты-копии могут сохранять связь с прототипом, автоматически наследуя изменения в нем – динамическое наследование.
- Объект — сущность, появляющаяся при создании экземпляра класса или при копировании прототипа.

Основные термины ООП

- Инкапсуляция — это свойство системы программирования, позволяющее объединить некоторые данные класса и методы, работающие с ними, и скрыть детали реализации от пользователя.
- Наследование — это свойство системы программирования, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствованной функциональностью. Класс, от которого производится наследование, называется базовым, родительским или суперклассом. Новый класс — потомком, наследником или производным классом.
- Полиморфизм — это свойство системы программирования, позволяющее одинаковым образом использовать объекты разных классов, имеющие одинаковый (полностью или частично) интерфейс.

Цели ООП

- уменьшение сложности разработки составных частей программной системы, тем самым – увеличение сложности создаваемых приложений;
- повышение надежности, устойчивости и управляемости программного обеспечения;
- взаимная независимость, т.е. обеспечение возможности модификации некоторых компонентов программного обеспечения без изменения остальных его компонентов;
- обеспечение возможности повторного использования отдельных компонентов программного обеспечения.