

Введение в ОО анализ и проектирование

Содержание

1. Предметная область
2. Модель предметной области
3. Базовые понятия
4. Методология процедурно-ориентированного программирования
5. Методология объектно-ориентированного программирования
6. Методология объектно-ориентированного анализа и проектирования
7. Методология системного анализа и системного моделирования
8. Классы и объекты
9. Модель «сущность - связь»
10. Unified Modeling Language
11. Требования к UML
12. Виды UML диаграмм

Предметная область

- это часть реального мира, данные о которой мы хотим отразить в базе данных, реализовать в информационной системе. Предметная область бесконечна и содержит как существенно важные понятия и данные, так и малозначащие или вообще не значащие данные.

Модель предметной области

- это наши знания о предметной области. Знания могут быть как в виде неформальных знаний в мозгу эксперта, так и выражены формально при помощи каких-либо средств. В качестве таких средств могут выступать текстовые описания предметной области, наборы должностных инструкций, правила ведения дел в компании и т. п. Имеется большое количество методик описания предметной области. Из наиболее известных можно назвать методику IDEF0, диаграммы потоков данных, UML.

Базовые понятия

Декомпозиция - создание модулей, которые в свою очередь представляют собой небольшие программы, взаимодействующие друг с другом по хорошо определенным и простым правилам.

Абстрагирование – процесс выделения абстракций в предметной области задачи. Абстракция – совокупность существенных характеристик некоторого объекта, которые отличают его от всех других видов объектов и, таким образом, четко определяют особенности данного объекта с точки зрения дальнейшего рассмотрения и анализа.

Модульность – принцип разработки программной системы, предполагающий реализацию ее в виде отдельных частей (модулей). Принцип унаследован от модульного программирования, следование ему упрощает проектирование и отладку программы.

Иерархия – ранжированная или упорядоченная система абстракций. Принцип иерархичности предполагает использование иерархий при разработке программных систем.

Методология процедурно-ориентированного программирования

Основой такой методологии разработки программ являлась процедурная, или алгоритмическая, организация структуры программного кода.

Исходным в данной методологии было понятие алгоритма. Алгоритм - это способ решения вычислительных и других задач, точно описывающий определенную последовательность действий, которые необходимо выполнить для достижения заданной цели.

Процедурно-ориентированные языки:

Си, C++, Basic, Фортран, Pascal, и т.д.

Минус: проблема управления и сохранения данных

Методология объектно-ориентированного программирования

Инкапсуляция – позволяет скрывать внутреннюю реализацию. В классе могут быть реализованы внутренние вспомогательные методы, поля, к которым доступ для пользователя необходимо запретить, тут и используется инкапсуляция.

Наследование – позволяет создавать новый класс на базе другого. Класс, на базе которого создается новый класс, называется базовым, а базирующийся новый класс – наследником.

Полиморфизм – это способность объектов с одним интерфейсом иметь различную реализацию.

Абстракция – позволяет выделять из некоторой сущности только необходимые характеристики и методы, которые в полной мере (для поставленной задачи) описывают объект.

Объектно-ориентированные языки программирования: C#, C++, Java, JavaScript, PHP, VB.NET, Python

Методология объектно-ориентированного анализа и проектирования

Циклы разработки ПО с использованием ОО проектирования:

Основной этап: ОО анализ, больше всего времени. Задача ООА – построение модели системы.

На основе построенной модели выполняется проектирование.

ОО анализ

В структурном программировании используется принцип черного ящика – мы не думаем о конкретных действиях и данных, а о том что нужно сделать.

В ООП – белый ящик, первое всего – данные. Мы выделяем характеристики объектов, и потом уже приходим к тому, как их обрабатывать.

Методология системного анализа и системного моделирования

Центральным понятием системного анализа является понятие системы, под которой принято понимать совокупность объектов, компонентов или элементов произвольной природы, образующих некоторую целостность.

Важнейшими характеристиками любой системы являются ее структура и процесс функционирования. Под структурой системы понимают устойчивую во времени совокупность взаимосвязей между ее элементами или компонентами.

Процесс функционирования системы тесно связан с изменением ее свойств или поведения во времени.

Классы и объекты

Класс – это абстрактный тип данных. С помощью класса описывается некоторая сущность (ее характеристики и возможные действия). Описав класс, мы можем создать его экземпляр – объект.

Объект – это уже конкретный представитель класса.

- Состояние объектов
- Поведение объектов
- Объекты как автоматы

- Интерфейс класса
- Реализация класса
- Отношения между классами
- Наследование
- Инстанционирование

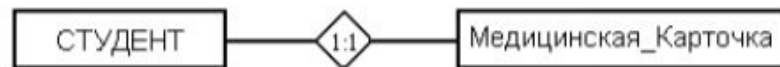
Модель «сущность - связь»

Сущность - объект, который может быть классифицирован

Атрибуты - информация, описывающая сущность

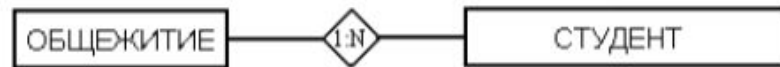
Идентификаторы - атрибуты для идентификации экземпляра сущности

Связи - множественность, которая описывает тип связи между сущностями
(1:1, 1:N, M:N)



СТУДЕНТ-Медицинская_Карточка

а



ОБЩЕЖИТИЕ-СТУДЕНТ

б



СТУДЕНТ-КЛУБ

в

Unified Modeling Language - унифицированный язык моделирования

UML - язык графического описания для объектного моделирования в области разработки программного обеспечения, моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

- это специальный язык, описывающий объектно-ориентированные модели, которым в будущем суждено стать программным кодом.

1. Метод Буча (Booch'93), ориентированный, в первую очередь, на моделирование программного обеспечения сложных систем.
2. Метод Рамбо (OMT-2), ориентированный на анализ процессов обработки данных в информационных системах.
3. Метод Джекобсона (метод OOSE), ориентированный на анализ требований к бизнес-приложениям.

Требования к UML

1. Позволяет моделировать как программное обеспечение сложных систем, так и широкие классы самих систем и бизнес-приложений, с использованием объектно-ориентированных понятий и методов.
2. Обеспечивает взаимосвязь между базовыми понятиями моделей концептуального, программного и физического уровней.
3. Понятен системным аналитикам и программистам.
4. Поддерживается специальными инструментальными программными средствами, реализованными на различных компьютерных платформах.

Виды UML диаграмм

Диаграмма вариантов использования (use case diagram)

Диаграмма классов (class diagram)

Диаграмма состояний (statechart diagram)

Диаграмма деятельности (activity diagram)

Диаграмма последовательности (sequence diagram)

Диаграмма кооперации (collaboration diagram)

Диаграмма компонентов (component diagram)

Диаграмма развертывания (deployment diagram)