


Использование принципов ООП в программной инженерии

Подготовил: Искаков Д.А.



Этапы развития программирования

- Технология программирования – это система методов, способов и приемов обработки и выдачи информации. Написание программ в машинных кодах. Появление языков низкого уровня. Языки высокого уровня. Технология нисходящего структурного программирования
Появление ООП
- 

Объектно-ориентированные языки программирования

Абстракция

Инкапсуляция

Полиморфизм

Наследование

Синтаксис

Структура программы

Отладка

Классы/ объекты
(Simula-67)

Контейнерные классы
(Smalltalk)

Параметризованные классы и функции
(Шаблоны)
C++

Абстрактные классы/
Интерфейсы
(C++,Java)

Роли (типажи)
(Perl 6)

Соккрытие информации без привязки к конкретным классам
(Smalltalk)

Привязка к конкретным классам
(Eiffel)

Переопределение функций
(Simula-67)

Перегрузка операторов
(Smalltalk)

Одиночное наследование
(Simula-67)

Множественное наследование
(C++)

Множественное наследование интерфейсов
(Java)

Наследование ролей
(Perl 6)

Упрощение синтаксических конструкций
(Smalltalk)

Самодокументируемость
(Eiffel)

Отказ от управляющих конструкций
(Python)

Общий текст программы
(Simula-67)

Объектная декомпозиция
(Smalltalk)

Разделение на модули
(C++)

Среда разработки
(Smalltalk)


Шаблоны
(C++)


Обработка исключений
(C++)

Проектирование по контракту
(Eiffel)



Что такое программная инженерия

- Программная инженерия – это область компьютерной науки и технологии, которая занимается созданием программных систем, настолько больших и сложных, что для этого требуется участие слаженных команд разработчиков различных специальностей и квалификаций. Обычно такие системы существуют и применяются долгие годы, развиваясь от версии к версии, претерпевая на своем «жизненном пути» множество изменений: улучшение существующих функций, добавление новых и удаление устаревших возможностей для работы в новой среде, устранение дефектов и ошибок.
- 



Главная цель программной инженерии

- Главная цель программной инженерии - сокращение стоимости ПО
Сформировались основные принципы и методы проектирования ПО:
Жизненный цикл ПО Модульное программирование Структурное проектирование и программирование. Объектно-ориентированное проектирование и программирование

Выработка системных требований

Выработка требований к ПО

Анализ

Проектирование

Кодирование

Тестирование

Эксплуатация

Определение задач, альтернатив, ограничений

Совокупная стоимость

Ход работ

Оценка альтернатив, выделение рисков и способов борьбы с ними


Фиксация результатов

Экспертизы




Планирование следующих итераций

Разработка и верификация очередной части продукта



Программная инженерия (Software Engineering) - это направление промышленного программирования, предназначенное для решения следующих задач:

- установление и использование инженерных принципов и методов для экономичного получения надежного программного обеспечения;
- применение принципов математики и информатики для получения рентабельных решений в области программного обеспечения вычислительной техники;
- применение систематического подхода к разработке, эксплуатации и сопровождению программного обеспечения.



Что такое ООП в программной инженерии?

- Объектно-ориентированное программирование (ООП) — это способ организации кода в программе путем его группировки в виде объектов, то есть отдельных элементов, которые включают информацию (значения данных) и функции. Объектно-ориентированный подход к организации программы позволяет группировать определенные фрагменты информации (например, музыкальную информацию, такую как название альбома, название трека или имя исполнителя) вместе с часто используемыми функциями или действиями, связанными с этой информацией (например, «добавить трек в список воспроизведения» или «воспроизвести все песни этого исполнителя»). Эти элементы объединяются в один элемент, так называемый объект (например, «Album» или «MusicTrack»). Возможность связать вместе эти значения и функции дает несколько преимуществ. Например, нужно отслеживать только одну переменную, а не несколько; можно сгруппировать вместе связанные между собой функции; можно организовать структуру программы так, чтобы она больше соответствовала реальным потребностям.



Основополагающие концепции ООП

- ▣ **Абстракция данных, Абстрагирование** означает выделение значимой информации и исключение из рассмотрения незначимой. В ООП рассматривают лишь абстракцию данных (нередко называя её просто «абстракцией»), подразумевая набор значимых характеристик объекта, доступный остальной программе.
- ▣ **Инкапсуляция** — свойство системы, позволяющее объединить данные и методы, работающие с ними, в классе. Некоторые языки (например, C++) отождествляют инкапсуляцию с сокрытием, но большинство (Smalltalk, Eiffel, OCaml) различают эти понятия.
- ▣ **Объект** Сущность в адресном пространстве вычислительной системы, появляющаяся при создании экземпляра класса (например, после запуска результатов компиляции и связывания исходного кода на выполнение).



Основополагающие концепции ООП

- ▣ **Наследование** — свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью. Класс, от которого производится наследование, называется базовым, родительским или суперклассом. Новый класс — потомком, наследником, дочерним или производным классом.
- ▣ **Класс** является описываемой на языке терминологии исходного кода моделью ещё не существующей сущности (объекта). Фактически он описывает устройство объекта, являясь своего рода чертежом. Говорят, что объект — это экземпляр класса. При этом в некоторых исполняющих системах класс также может представляться некоторым объектом при выполнении программы посредством динамической идентификации типа данных. Обычно классы разрабатывают таким образом, чтобы их объекты соответствовали объектам предметной области.



Этапы разработки программ с использованием ООП

- ООП представляет собой систематизированный подход к алгоритмической формализации сложных предметных областей. ООП предполагает этапы разработки программ: Первый этап: абстрагирование Второй этап: инкапсуляция Третий этап: модульность Четвертый этап: иерархия



Этапы разработки программ с использованием ООП Абстрагирование

- Выделение абстракций. Это - анализ предметной области, для которой составляется программа, с целью определения основных объектов этой предметной области, их свойств, отношений между объектами, возможных операций над объектами или их составляющими. Различие: Процедурное программирование нацелено на моделирование действий, выполняемых компьютером. ООП нацелено на моделирование предметной области решаемой задачи.



Этапы разработки программ с использованием ООП Инкапсуляция

- Типизация объектов и синтез абстрактных типов данных. Определение новых типов данных и наборов специфических функций и операций, применяемых к этим типам данных. Модульность Объектная декомпозиция. Выделение подтипов (подобъектов) для каждого из типов и их составляющих. Иерархия Композиционная иерархизация объектов. Т.е. выделение родовитых и композиционных отношений над объектами.



Инкапсуляция

- Инкапсуляция (encapsulation) - это объединение производного типа данных с набором функций, используемых при работе с этим типом данных, в единый класс. Функции, включенные в класс, называют методами класса. Данные – элементами или полями класса, Конкретные представители класса – объекты, экземпляры. Класс (объект) - это то, что поддерживает инкапсуляцию. Инкапсуляция позволяет сделать класс «самодостаточным» для решения конкретной задачи. Класс всегда несет в себе некоторую функциональность. Это мощное средство обмена готовыми к работе программными заготовками.



Инкапсуляция и ограничение доступа к данным

- Инкапсуляция предполагает возможность ограничения доступа к данным (полям) класса. Это позволяет упростить интерфейс класса, показав наиболее существенные для внешнего пользователя данные и методы. обеспечить возможность внесения изменений в реализацию класса без изменения других классов (важно для дальнейшего сопровождения и модернизации программного кода). При сокрытии полей объекта доступ к ним осуществляется только посредством методов класса. Это защищает данные от внешнего вмешательства или неправильного использования




Наследование

- Наследование (inheritance) - это возможность определять новые классы посредством добавления полей, свойств и методов к уже существующим классам. Такой механизм получения новых классов называется порождением. При этом новый, порожденный, класс (потомок) наследует все поля, методы и свойства своего базового, родительского класса. Наследование поддерживает концепцию иерархии классов (hierarchical classification). Применение иерархии классов делает управляемыми большие потоки информации. Наследование обеспечивает поэтапное создание сложных классов и разработку собственных библиотек классов



Полиморфизм

- Полиморфизм (polymorphism) Это возможность заменить в классе потомке метод класса родителя, сохранив при этом имя метода. Это свойство классов решать схожие по смыслу проблемы разными способами. Цель полиморфизма - использование одного имени для задания общих для класса действий. Для изменения метода необходимо перекрыть его в потомке, т.е. объявить в потомке одноименный метод и реализовать в нем нужные действия. В результате объекте-родителе и объекте-потомке будут действовать два одноименных метода, имеющие разную алгоритмическую основу. Концепция полиморфизма - идея «один интерфейс - множество методов». Полиморфизм позволяет манипулировать объектами различной степени сложности путем создания общего для них стандартного интерфейса для реализации похожих действий.



Общие операции в объектно-ориентированном программировании

На практике объектно-ориентированное программирование делится на две части. Одна часть включает стратегии и приемы для разработки программ (или объектно-ориентированного дизайна). Вторая часть ООП — это собственно структуры программирования, доступные в данном языке программирования для создания программы с использованием объектно-ориентированного подхода. Существуют следующие общие операции в ООП:

- определение классов;
- создание свойств, методов, а также методов доступа get и set;
- управление доступом к классам, свойствам, методам и методам доступа;
- создание статических свойств и методов;
- создание структур с перечислением;
- определение и использование интерфейсов;
- работа с наследованием, включая переопределение элементов класса.

Спасибо за внимание)

ggwp!!!

