Введение в объекты

ООП как развитие абстракции

- Все является объектом
- Программа - набор объектов, указывающих друг другу, что делать, по средствам сообщений
- Каждый объект имеет собственную память(ресурсы), состоящую из других объектов
- У каждого объекта есть тип
- Все объекты определенного типа могут получать одинаковые сообщения

- Объект обладает состоянием (полями), поведением (методами) и уникальностью (своей областью в памяти)

Интерфейс класса

- Определяет какие поля и методы будут реализованы у объекта

Реализация класса

- Код выполняющий эти запросы

Объект предоставляет услуги

- Важно не перенасыщать методами
- Объект как поставщик услуг должен выполнять свои функции, не пытась сделать большее

Скрытая реализация

- Создатель класса скрывает реализацию от клиента
- Сокрытие может осуществляться за счет ключевых слов private public protected

Повторное использование реализации

- Композиция (отношения между классами - А содержит Б, Б не существует без А)
- Агрегация - не так сильно-связный вариант композиции (А имеет Б)

Наследование

•Класс родитель - имеет основную логику, при изменении логики меняется логика наследников(тип объекта - базовый)
•Класс наследник - расширяет логику родителя (тип объекта - произовдный, частный случай базового)
•Может дополнять класс новыми методами (связь - похоже на)
•Может переопределять старые (связь является)

Взаимозаменяемые объекты и полиморфизм

- Ранее связывание - компилятор вызывает функцию с соответствующим именем, компоновщик привязывает ее к определенному месту кода
- Позднее связывание - компилятор убеждается в наличии вызываемой функции, вызываемый код не известен до времени выполнения

- Восходящее преобразование типов - объект трактуется как его родитель

Однокорневая иерархия

- В основе - единственный класс Object
- Все объекты обладают общей функциональностью
- Реализация уборки мусора

Контейнеры

•Можно хранить объекты не задавая размер
•List,Map,Set
•Контейнер подойдет для любых элементов - при получение элемента срабатывает нисходящее преобразование
•Параметризация - для исключения ошибок в случае нисходящего преобразование (List <MyObject>)

Создание и время жизни объектов

- Хранятся в куче, время жизни, тип и количество объектов не известно до момента запуска программы (тк все объекты большие и сложные) - для построения динамического экземпляра - ключевое слово new
- Время жизни не известно из-за хранения в куче, но уборщику известно

Обработка исключений

- Исключения генерируются на месте ошибок. Исключения требуют их обработки

Параллельное выполнение

- Для осуществления параллельного выполнения используются потоки. Проблемы с потоками наступают при необходимости получения одного ресурса несколькими потоками
- Реализация различных блокировок (примитивов синхронизации)

веб

- Клиент-Сервер - ресурсы и вычисления на сервере, получает/отправляет клиент, проблемы с доступом с несколькими клиентами сразу, необходимость оптимизации вычислений => распределение вычислений

Программирование на стороне клиента

Изначально поддержка интерактивности только на стороне сервера. HTML поддерживал только простейшие средства представления.
Разработка модулей расширения позволила расширить работу со стороны браузера
Для расширения графической и интерактивной составляющей визуальной части клиента - языки сценариев
Разработка апплетов - на джава (мини-программа исполняемая только внутри браузера)

Сервер

- Интрасеть - информационная сеть внутри компании
- Задача программирования на сервере - обеспечить совместимость браузеров
- + Обращение к базе данных

Резюме

- Процедурные прогр - определение данных и вызов функций
- ООП повышает читабельность кода (если все хорошо)