4 лабораторная работа Наследование: расширение, спецификация, специализация, конструирование

Спецификация – можно реализовать двумя методами – с помощью интерфейса или абстрактного класса – те суперкласс описывает поведение которое будет реализовано в подклассе

В интерфейсе  
  
между классом А и классом С, можно взять интерфейс и присвоить ему объект нижестоящего класса

С точки зрения подстановки, мы подставили и произошло замещение

Спецификация через абстрактный класс

Отличие интерфейса от абстрактного класса   
в интерфейсе прописываются только сигнатуры методов, то есть нельзя описывать сами методы и атрибуты, а в абстрактном классе можно описывать, но при этом и там и там нельзя создать объекты, можно только наследоваться от них

Расширение, при расширении в подкласс добавляются новые методы а также могут замещать методы супер класса, принцип подстановки выполняется не полность, не выполняется для добавленных методов

Специализация – 3 лабораторная иерархия общее частное

Конструировани – в супер классе добавляются функция заглушка, мы делаем его более общим, закрываем методы, он будет общий, убрать некоторые методы

Цель: чтобы унаследовать функцинальность другого класса

4\_2

Комбинирование -проблемы – при комбинировании, ограничения- нельзя наследоваться от двух классов, можно наследоваться от интерфейсов, но от более 2 классов нельзя,

Проблема множественного – может возникнуть конфликт имен, можно избежать, если прописать в подклассе написать new или использовать разные модификаторы доступа

комбинирование – реализация множественного наследования, при котором можно скомбинировать разные категория наследования, может быть спецификация расширение то что было до этого

5

Комбинирование через общих предков

Наследование есть интерфейс А, от него наследуется интерфейс В и класс С, от которых наследуется класс К

Проблемы – те же самые, нельзя наследоваться от 2 и более классов, конфликт имен может быть

Конструкиция для атрибутов base

Это вызов конструктора базового класса

В конструкторе К мы можем прописать base? И когда мы записываем Передаем какой то параметр в конструктор он передается в суперпкласс

Если забыли base написать – будет логическая ошибка, но компиляцию пройдет, не будет передан параметр в конструктор суперкласса, то атрибуту С не будет присовоено занчение, вызовется контруктор без параметров

Комбинирование через общих предков позволяет реализовать множественное наследование через общий суперкласс; комбинирует в себе два класса с общим суперклассом, который, в свою очередь, сам присутствует только в единственном экземпляре благодаря virtual наследованию, позволяющее избежать дублирование этого суперкласса.

Она же наследуется по 1 ветке и она же определяет ту функцию которая пришла

При наследовании через общих предком мы избегаем конфликта имен

**Лаба 6**

Ассоциация 1 к одному 1 к многим

Ассоциация отношение взаимосвязи в которой не прослеживается четкая иерархия

В ассоциации 1 к 1 происходит ассоциация между 2 объектами разных классов

Реализуется с помощью атрибута который имеет тип данных другого класса, создаем ссылку на него, и с помощью этого атрибута внутри класса будут доступны функции атрибуты и операции другого класса

Мы создаем объекты двух классов и в качестве параметра в конструкторы передаем объект второго класса для того чтобы присвоить ссылку на него в другой атрибут

**Один к многим**

Связываем объект одного класса сразу с несколькими объектами другого класса

Либо сделать атрибут публичным

Либо сделать через атрибут доступа

Ассоциация бинарного отношения

В классе А есть массив класса И который cодержит ссылки на объекты класса И

Инициализация массива

Конкретизация переход от от общего к частному мы используем шаблон

Хэш код объекта

Ref взятие адреса объектов для работы с глобальной памятью

(в 1 лабе мы брали атрибуты и записывали ссылки в атрибуты)

Копии адресов

Конркетизация параметров

**Лаба 7 использование клиент сервер, клиент утилита**

Клиент (использует ресурсы) – сервер (ресурсы – атрибуты, функции, методы)

Иерархи подчинения

Клиент-утилита применяется

Static – можно вызвать от класса утилиты, а не от объекта класса – методы который не зависит от обхекта класса,

Он общий для всего класса, методы для которого не нужно создавать объект

Не зависит от объекта класса

Для статичного метода не нужн осоздавать объекты

Нельзя пронаследоваться от него

Оно устанавлиает

Клиент оращается к операции FD в которой в качестве параметра передается обхект класса сервера