Лабораторная работа №3
Наследование, полиморфизм и шаблоны проектирования

Часть 1
(C++, C#, Java, кроме Python)

Цель: познакомится с шаблоном проектирования Strategy, изучит возможности использования наследования и полиморфизма в обектно-ориентированной парадигме.

Сначала к работе идут общие теоретические сведения, далее варианты для самостоятельной работы.

Не забываем, что если вам неинтересны данные предложенные варианты, вы можете придумать свой 😉

Наследование

Наследование служит для возможности повторного использования кода. Класс, от которого выполняется наследование, называется базовым. Производному классу становятся доступны члены базового класса с модификаторами `public` и `protected`. Для объявления производного класса после его имени ставится двоеточие и далее указывается имя базового.

Базовый класс представляет наиболее общие характеристики и поведение описываемого объекта. В производном классе определена более специфическая разновидность предмета описания базового класса. Примером иерархии наследования может служить базовый класс «Человек», с такими атрибутами как имя, возраст, адрес и производный класс «Работник», с атрибутами зарплата, должность.

Полиморфизм

Полиморфизм позволяет классам с одинаковой спецификацией иметь различную реализацию, которая может быть изменена в процессе наследования. Один из способов добавления полиморфного поведения в программу – это описание виртуальных методов. Виртуальным называется такой метод, который объявляется с модификатором `virtual` в базовом классе. Виртуальный метод отличается тем, что он может быть переопределен в одном или нескольких производных классах и у каждого производного класса может быть свой вариант реализации виртуального метода. При переопределении виртуального метода в производном классе указывается ключевое слово `override`. В ряде случаев уровень абстрагирования, предоставляемый базовым классом, не предполагает какую-либо практическую реализацию некоторых методов класса. Такие методы объявляются с ключевым словом `abstract` и не содержат тело метода. Класс с абстрактными методами считается абстрактным, при этом становится запрещено создавать объекты этого класса. Как и в случае виртуального метода, абстрактный метод в производном классе переопределяется при помощи метода `override`.

Развитием концепции абстрактных методов в C# является интерфейс. Для объявления интерфейса указывается ключевое слово `interface`, далее идет имя интерфейса и в фигурных скобках перечисляются имена методов без реализации.

Класс может быть унаследован от одного или нескольких интерфейсов. Методы, реализующие интерфейс, должны быть объявлены как `public`. Имя интерфейса в C++ не обязательно должно начинаться с буквы `I`.

Интерфейсы

Парадигма существующая не только в C++, но здесь более подробно поговорим о реализации именно в C++. Интерфейсы реализуются через абстрактные классы, содержащие чисто виртуальные функции. Интерфейсы предоставляют абстрактный набор методов, которые классы должны реализовать. Это обеспечивает структурную однородность в приложении, позволяя разным классам предоставлять общий интерфейс.

Шаблон проектирования Strategy

Шаблон проектирования Strategy использует полиморфизм, для того чтобы определить семейство алгоритмов и сделать их взаимозаменяемыми. Для реализации шаблона Strategy можно объявить интерфейс, содержащий метод, предполагающий множество реализаций. Для каждого алгоритма реализации следует объявить класс, унаследованный от интерфейса и предоставляющий реализацию алгоритма.

Полезные ссылки:
Наследование и полиморфизм - Основы C++ (yandex.ru)
Обработка исключений - Основы C++ (yandex.ru)
Тут Оглавление | Patterns и Паттерны проектирования на C++ тут есть и другие паттерны, если вам не нравится `Strategy`, можете выбрать другой поведенческий паттерн.
Стратегия (шаблон проектирования) — Википедия (wikipedia.org) тут есть примеры для различных языков
Практическое использование шаблона Стратегия / Хабр (habr.com)
Разработка интерфейсных классов на C++ / Хабр (habr.com) -- объемно, но достаточно полно
Interfaces in C++ (Abstract Class) - Explore How Pure Virtual Function Works - DataFlair (data-flair.training) -- кратко, но на английском языке.

Задание
Общая часть

В данной работе необходимо использовать один из шаблонов проектирования, например описанный выше `Strategy,` реализовать наследование, полиморфизм и парадигму интерфейсных классов. В работе для обработки ошибок должны быть реализованы исключения (часть проверок можно сделать классическим способом).

Сначала необходимо создать диаграмму классов, а потом уже начинать реализацию.

Варианты:

1. Предметная область: **АТС**. АТС имеет список тарифов на междугородние разговоры. Есть два типа тарифов: обычный и льготный. В классе «АТС» реализовать

методы добавления обычного тарифа и добавления льготного тарифа. Класс «АТС» должен выполнять вычисление средней стоимости тарифов с учетом скидки.

2. Предметная область: **Вокзал**. Класс «вокзал» имеет список тарифов на различные направления. На некоторые тарифы может быть предоставлена скидка, заданная в процентах. В классе «вокзал» реализовать методы добавления нового тарифа со скидкой и без скидки, поиск направления с минимальной стоимостью.

3. Предметная область: **ЖЭС**. ЖЭС имеет информацию о всех жильцах. Имеются два типа жильцов со льготами и без льгот. В классе «ЖЭС» реализовать метод добавления нового жильца, имеющего и не имеющего льготы, а также метод подсчета стоимости всех оказанных услуг.

4. Предметная область: **Аэропорт.** Касса аэропорта имеет список тарифов на различные направления. Тариф содержит название направления и стоимость перевозки. На некоторые направления предоставляется фиксированная скидка. В классе «аэропорт» реализовать метод добавления нового тарифа и метод поиска направления с максимальной стоимостью.

5. Предметная область: **Банк**. Система хранит информацию о вкладчиках и сделанных ими вкладах. Класс «вкладчик» содержит имя вкладчика и величину вклада. Некоторым вкладчикам при создании вклада на счет может дополнительно перечисляться фиксированная сумма. В классе «банк» реализовать методы добавления нового вкладчика и метод вычисления общей суммы вкладов.

6. Предметная область: **Отдел расчета зарплаты**. Информационная система отдела расчета зарплаты на предприятии хранит данные о величине оплаты за различные виды работ. На некоторые виды работ предоставляется надбавка, заданная в процентах. В классе «отдел расчета зарплаты» реализовать методы добавления нового типа работ и метод вычисления средней величины оплаты.

7. Предметная область: **Фирма грузоперевозок**. Фирма имеет список тарифов по перевозке грузов. Класс «тариф» хранит наименование тарифа и цену. На некоторые тарифы предоставлена скидка, заданная в процентах. В классе фирма реализовать методы добавления нового тарифа и метод поиска тарифа с минимальной стоимостью.

8. Предметная область: **Гостиница.** Информационная система гостиницы хранит информацию о всех номерах и их стоимости. На проживание в некоторых номерах предоставляется скидка, заданная в процентах. В классе «гостиница» реализовать метод добавления информации о номере и метод вычисления средней стоимости.

9. Предметная область: **Интернет-оператор**. Информационная система провайдера хранит данные о клиентах. Некоторым клиентам предоставляется фиксированная скидка. В классе «оператор» реализовать метод добавления нового клиента и метод вычисления суммарной стоимости оказанных услуг.

10. Предметная область: **Интернет-магазин**. В информационной системе хранятся данные о товарах. Класс «товар» содержит стоимость товара и его наименование. На некоторые товары предоставляется скидка, заданная в процентах. В классе «магазин» реализовать метод добавления нового товара, имеющего скидку и не имеющего, также метод поиска товара с минимальной стоимостью.

Если вашего варианта нет в списке, то вы начинаете отсчитывать с первого варианта: например, если вы в списке группы 16, то ваш вариант 6, если вы списке 30, то ваш вариант 1й, если вы 24, то ваш вариант 4.

Откуда вдохновение: Obektno Orientirovanie Programmirovanie.pdf (bntu.by)

Что еще можно:

- Написать свою реализацию если вы чувствуете себя достаточно уверенным программистом и у вас есть идея, которая будет похожа на данное задание.
- Если у вас нет идей как реализовать наследование по вариантам выше, а вы можете реализовывать в классах дополнительный фцкуционал, то можно потренироваться реализовывать наследование в отдельной программе, варианты для реализации могут быть следующими:

Спроектировать иерархию для заданной предметной области:

1) автотранспорт;
2) жилищно-коммунальная сфера;
3) здравоохранение;
4) бытовое обслуживание населения;
5) образование;
6) муниципальное управление;
7) железнодорожный транспорт;
8) авиаперевозки;
9) компьютерная техника;
10) энергетика.

Как и с основной задачей вы можете воплотить свою индивидуальную реализацию.

Часть 3
Отчет

Отчет как и в первой лабораторной работе. В этот раз Python не нужно, располагать код на github -- опционально. Диаграмма классов обязательна!