

Практическая работа 22 Разработка иерархии классов

Цель занятия: Получить практические навыки анализа предметной области

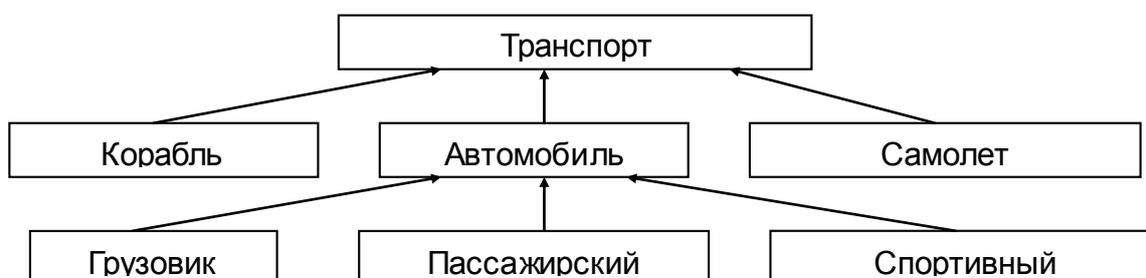
Перечень оборудования и программного обеспечения

Персональный компьютер
Microsoft Office (Word, Visio)
Microsoft Visual Studio

Краткие теоретические сведения

Наследование

Наследование — это свойство, с помощью которого один объект может приобретать свойства другого. При этом поддерживается концепция иерархической классификации, имеющей направление сверху вниз.



Используя наследование, объект должен определить только те качества, которые делают его уникальным в пределах своего класса. Он может наследовать общие атрибуты от своих родительских классов.

Синтаксис:

```
class имя_класса : имя_родительского_класса  
{тело_класса}
```

Пример:

```
class Predator:Animal  
{  
private int Speed;  
}
```

С помощью наследования создается иерархия классов (отношение ‘являться’). Кроме того, можно построить еще одну структуру – иерархию объектов (тогда, когда один объект является частью другого – отношение ‘часть-целое’).

Диаграммы классов показывают набор классов, интерфейсов, а также их связи. Диаграммы этого вида чаще всего используются для моделирования объектно-ориентированных систем. Они предназначены для статического представления системы.

Класс – это описание набора объектов с одинаковыми атрибутами, операциями, связями и семантикой.

Графически класс изображается в виде прямоугольника, разделенного на 3 блока горизонтальными линиями:

- имя класса
- атрибуты (свойства) класса
- операции (методы) класса.

Для атрибутов и операций может быть указан один из трех типов видимости:

- — private (частный)
- # — protected (защищенный)
- + — public (общий)

Видимость для полей и методов указывается в виде левого символа в строке с именем соответствующего элемента.

Каждый класс должен обладать именем, отличающим его от других классов. Имя – это текстовая строка. Имя класса может состоять из любого числа букв, цифр и знаков препинания (за исключением двоеточия и точки) и может записываться в несколько строк.

На практике обычно используются краткие имена классов, взятые из словаря моделируемой системы. Каждое слово в имени класса традиционно пишут с заглавной буквы (верблюжья конвенция).

Атрибут (свойство) – это именованное свойство класса, описывающее диапазон значений, которые может принимать экземпляр атрибута. Класс может иметь любое число атрибутов или не иметь ни одного. В последнем случае блок атрибутов оставляют пустым.

Атрибут представляет некоторое свойство моделируемой сущности, которым обладают все объекты данного класса. Имя атрибута, как и имя класса, может представлять собой текст. На практике для именования атрибута используются одно или несколько коротких существительных, выражающих некое свойство класса, к которому относится атрибут.

Можно уточнить спецификацию атрибута, указав его тип, кратность (если атрибут представляет собой массив некоторых значений) и начальное значение по умолчанию.

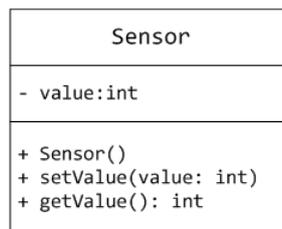
Статические атрибуты класса обозначаются подчеркиванием.

Операция (метод) – это реализация метода класса. Класс может иметь любое число операций либо не иметь ни одной. Часто вызов операции объекта изменяет его атрибуты.

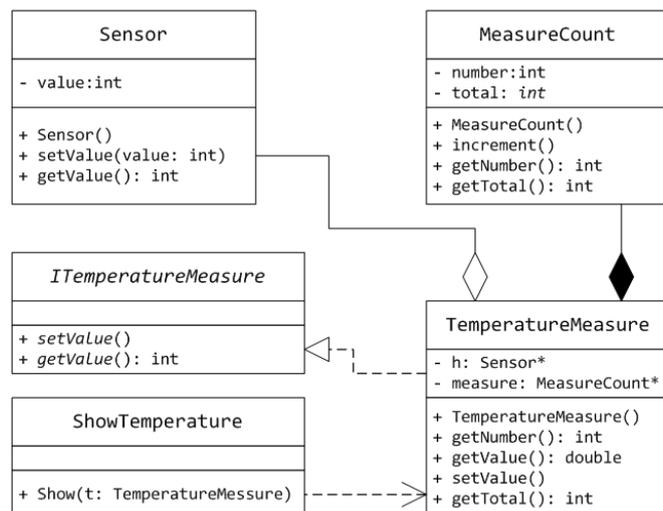
Графически операции представлены в нижнем блоке описания класса. Допускается указание только имен операций. Имя операции, как и имя класса, должно представлять собой текст. На практике для именования операции используются короткие глагольные конструкции, описывающие некое поведение класса, которому принадлежит операция. Обычно каждое слово в имени операции пишется с заглавной буквы, за исключением первого, например move (переместить) или isEmpty (проверка на пустоту). Можно специфицировать операцию, устанавливая ее сигнатуру, включающую имя, тип и значение по умолчанию всех параметров, а применительно к функциям – тип возвращаемого значения.

Абстрактные методы класса обозначаются курсивным шрифтом. Статические методы класса обозначаются подчеркиванием.

Изображая класс, не обязательно показывать сразу все его атрибуты и операции. Для конкретного представления, как правило, существенна только часть атрибутов и операций класса. В силу этих причин допускается упрощенное представление класса, то есть для графического представления выбираются только некоторые из его атрибутов. Если помимо указанных, существуют другие атрибуты и операции, вы даете это понять, завершая каждый список многоточием.



После организации связей диаграмма приобретает вид:



Задания

- 1 Изучить теоретические сведения и задание к работе
- 2 В соответствии с вариантом задания составить диаграмму иерархии классов.
- 3 В соответствии с вариантом задания составить отлаженную программу.

Порядок выполнения работы

Создадим базовый класс Pets (питомцы), в котором определим поля Name, Age. Создадим производные классы Кот, Собака. Кроме того создадим класс Птица, который будет базовым для класса Попугай:

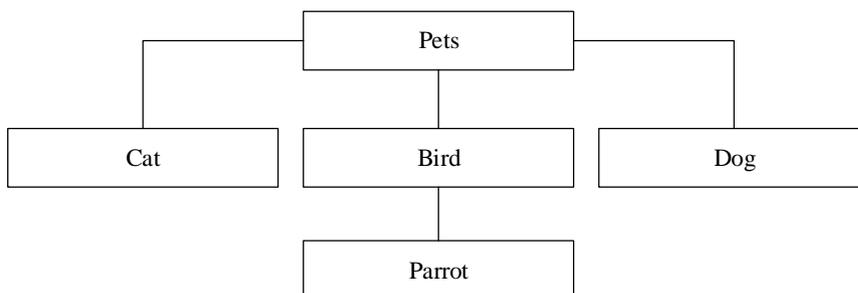
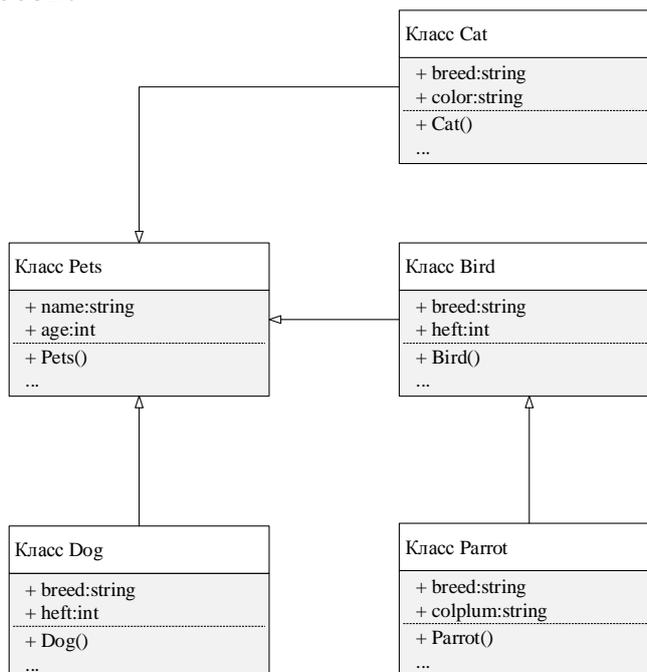


Диаграмма классов:



Создадим классы, в них объявим поля. Т.к. экземпляры созданных классов планируем создавать в другом классе, используем спецификатор public. После чего можно создавать объекты.

Содержание отчета

- 1 Название работы
- 2 Цель работы
- 3 Технические средства обучения
- 4 Задания (условия задач)
- 5 Порядок выполнения работы
- 6 Ответы на контрольные вопросы
- 7 Вывод

Варианты заданий

Построить иерархию классов используя наследование в соответствии с вариантом задания, в случае необходимости добавить понятия:

1. Студент, преподаватель, персонал, заведующий отделением
2. Служащий, персонал, рабочий, инженер
3. Рабочий, персонал, инженер, администрация
4. Республика, монархия, королевство, государство, империя, парламентская республика, президентская
5. Банк, организация, страховая компания, нефтегазовая компания, перерабатывающий завод
6. Журнал, книга, печатное издание, учебник
7. Рабочий, персонал, инженер, администрация, директор, завхоз
8. Место, область, город, мегаполис
9. Игрушка, продукт, товар, молочный продукт
10. Квитанция, накладная, документ, счет
11. Автомобиль, поезд, транспортное средство, экспресс
12. Деталь, механизм, коробка передач, двигатель
13. Корабль, бриг, пароход, парусник, подводная лодка, фрегат
14. Продукт, бытовая техника, товар, техника, посуда, компьютер
15. Магазин, здание, кафе, жилой дом, супермаркет, бутик
16. Двигатель, двигатель внутреннего сгорания, дизель, реактивный двигатель
17. Звезда, планета, небесное тело, спутник, комета, галактика
18. Млекопитающее, кошка, бизон, пресмыкающееся, птица, животное, павлин, утка
19. Тест, экзамен, выпускной экзамен, испытание, квалификационный экзамен
20. Шкаф, мебель, диван, кресло, мягкая мебель, корпусная мебель, стол
21. Шуба, одежда, майка, верхняя одежда, брюки, куртка, повседневная одежда, свитер

22. Фехтование, спортивная игра, спорт, единоборство, волейбол, командный спорт, водное поло, борьба,

23. Виолончель, саксофон, музыкальный инструмент, духовой музыкальный инструмент, там-там, струнный музыкальный инструмент, ударный музыкальный инструмент, гитара, барабан

Контрольные вопросы

- 1 Что такое наследование классов?
- 2 Как определяется наследование классов?
- 3 Сколько базовых классов может иметь класс?
- 4 С помощью какой программы можно строить диаграммы классов?
- 5 Какие элементы включает диаграмма классов?
- 6 Как на диаграмме классов показать наследование классов?

Используемая литература

1. Гниденко, И. Г. Технология разработки программного обеспечения: учеб. пособие для СПО / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — М.: Издательство Юрайт, 2017.

2. Шарп Джон Ш26 Microsoft Visual C#. Подробное руководство. 8-е изд. — СПб.: Питер, 2017.

3. Васильев А.Н. Программирование на C# для начинающих. Основные сведения. – Москва: Эксмо, 2018.

4. Васильев А.Н. Программирование на C# для начинающих. Особенности языка. – Москва: Эксмо, 2019.

5. <http://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx>.