

Лабораторная работа №4

Тема: «Наследование в C++»

Цель работы: получить навыки разработки иерархической системы классов для оптимизации проектирования структур данных при решении прикладных задач

Выполнение работы:

1. Анализ предметной области: определить чему посвящена предметная область, какие в ней есть термины и понятия, субъекты и объекты, способы взаимодействия субъектов, способы использования объектов, закономерности. Например, если речь идёт о графических примитивах в трёхмерном пространстве, то следует выявить список возможных примитивов (точка, линия, прямоугольник, параллелепипед, шар и т.п.), способы их описания (так, для точки достаточно указать её координаты, а для шара необходимо знать координаты центра и радиус), возможные способы преобразования (перемещение, масштабирование, поворот и т.п.). Отдельно следует сформировать краткий «словарь» предметной области, выделив список существительных (объекты и их характеристики) и список глаголов (методов). Темы / предметные области приведены в приложении 1.

2. Проектирование структуры классов для предметной области: определение набора классов, их свойств и методов, иерархии наследования или ассоциативных связей:

- a. определить, какой из классов какие свойства и методы содержит;
- b. выделить общие/ подобные / «пересекающиеся» свойства и методы, которые присущи разным классам;
- c. выявить отношения между классами, где требуется реализация наследования, а где включения;
- d. проверить, есть ли предпосылки для разработки базового абстрактного класса.

Результаты проектирования *можно* оформить в виде диаграммы классов. Предпочтительным является использование нотации языка UML со следующими правилами:

- класс обозначается прямоугольником;
- прямоугольник делится на три части, в каждой из которых, соответственно, указываются: имя класса, список свойств, список методов;
- имена классов, свойств и методов могут быть записаны на русском языке, но в соответствии с нормами написания стандартных идентификаторов (одно слово, включающее буквы, цифры, символ подчёркивания и не начинающееся с цифры);
- имена классов записываются с заглавной буквы, имена свойств и

методов – со строчной;

- перед именем свойства или метода ставится символ, указывающий на режим доступности: закрытый (-), защищённый (#), открытый (+);
- после имени метода ставятся круглые скобки, в которых могут быть перечислены параметры метода;
- наследование классов обозначается стрелкой с треугольным не закрашенным наконечником; стрелка при наследовании направляется от класса-наследника к родительскому классу.

Пример диаграммы классов приведён на рисунке 1.

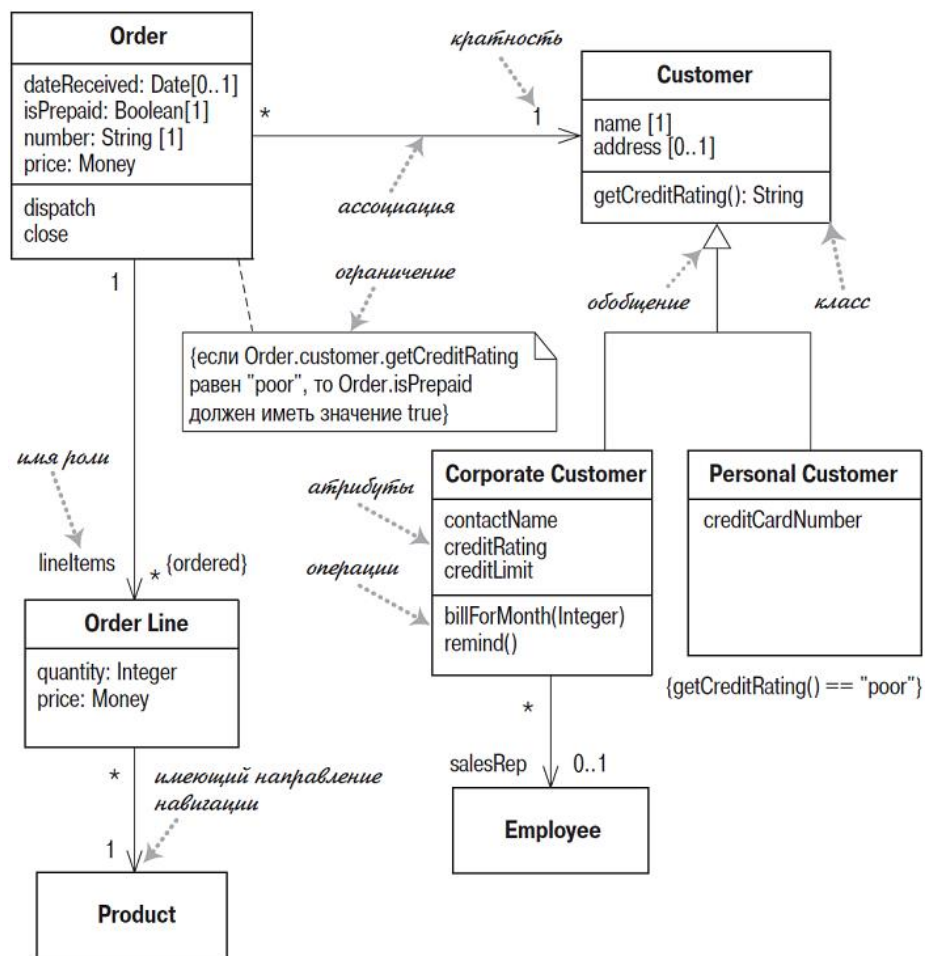


Рисунок 1 – Диаграмма классов трёхмерных графических примитивов

3. Формулирование тестовой задачи для проверки работы объектов, например, сформировать список объектов разных типов, но имеющих один базовый класс и для некоторых объектов, обладающих заданным свойством, преобразовать их или перенести в новый список. Также можно рассмотреть сценарии обработки каких-то запросов, в зависимости от системы и предметной области. Задача должна быть поставлена так,

чтобы проиллюстрировать работу каждого неабстрактного класса из разработанной системы классов.

4. Разработка упрощенной библиотеки классов, достаточной для демонстрации работы программы;

5. Разработка консольного приложения, тестирующего систему (библиотеку) классов, спроектированную и реализованную для решения конкретной задачи из некоторой предметной области.

6. Оформление отчета по результатам выполнения работы.

Приложение 1

1. Библиотека классов, реализующих графические примитивы на плоскости с возможностью аффинных преобразований (поворот, растяжение, сдвиг).
2. Библиотека классов, реализующих графические примитивы на плоскости с реализацией операций над множествами (объединение, пересечение, разность, симметрическая разность).
3. Классы для описания оконного графического интерфейса, аналогичного графическим интерфейсу современных операционных систем (например, окно, диалоговое окно, кнопка, поле ввода, рисунок, переключатель и т.д.). Должны быть реализованы обработчики событий с использованием виртуальных функций.
4. Библиотека классов, реализующих графические примитивы в трёхмерном пространстве с возможностью аффинных преобразований (поворот, растяжение, сдвиг).
5. Библиотека классов, реализующих комплексную арифметику (комплексное число, вещественное число, целое число, рациональное число, корень n -ой степени из числа).
6. Система классов для обеспечения работы с абонентами сотовой телефонной компании.
7. Система классов для обеспечения работы отдела кадров предприятия.
8. Система классов для обеспечения работы деканата.
9. Система классов, описывающих различные транспортные средства и обеспечивающая процесс их регистрации.
10. Моделирование работы туристического агентства.
11. Моделирование дорожного движения на заданной карте дорог.
12. Моделирование стационарной компьютерной сети.
13. Моделирование работы супермаркета.
14. Система классов для обеспечения работы менеджера банка по открытию вкладов.
15. Моделирование работы мессенджера
16. Система классов для обеспечения работы менеджера банка по выдаче кредитов.
17. Библиотека классов для реализации работы паспортного стола.