Лабораторная работа № 6

Часть 1

Тема работы: Исследование механизма множественного наследования

Перед выполнением лабораторной работы изучить тему множественного наследования. Обратить внимание на проблемы множественного наследования и на виртуальные базовые классы. Все задания сопроводить пояснениями. Каждое задание должно быть реализовано в виде одного .cpp файла.

Задание1:

1. Класс Derived является производным (public) от классов Base1 и Base2. Каждый из трех классов имеет по два конструктора, которые (кроме описанных ниже действий) выводят сообщение о вызове типа: "Конструктор Base без параметров".
2. Класс Base1, имеет одно закрытое поле i целого типа. Первый конструктор не имеет параметров и обнуляет i. Второй имеет один параметр типа int, используемый для инициализации i произвольными значениями. Класс имеет две общедоступные интерфейсные функции void put(int) и int get(void), которые позволяют изменить или прочесть значение i.
3. Класс Base2, имеет одно закрытое поле - массив name из 20 элементов. Первый конструктор не имеет параметров и инициализирует поле name словом "Пусто". Второй имеет один параметр типа char*, используемый для инициализации name значениями символьных строк. Класс имеет две общедоступные интерфейсные функции void put(char*) и char* get(void), которые позволяют изменить или прочесть значение name.
4. Класс Derived имеет одно закрытое поле ch типа char. Первый конструктор не имеет параметров и присваивает ch значение 'V' (от void – пустой). Второй конструктор имеет три параметра типов char, char* и int, используемые для инициализации соответственно полей ch, name и i.

Класс имеет две общедоступные интерфейсные функции void put(char) и char get(void), которые позволяют изменить или прочесть значение ch. Кроме того, в нем объявляется как дружественная операция вставки в поток вывода, которая выводит на экран значения i, name и ch.

5. В функции main описать переменную типа Derived без инициализации и вывести ее значение с помощью перегруженной операции вставки в поток. Выяснить порядок вызова конструкторов.
6. Описать другую переменную класса Derived, инициализировав ее явно некоторыми значениями. Вывести значение этой переменной на экран и проанализировать порядок вызова конструкторов.
7. В конструкторе класса Derived с параметрами изменить порядок вызова конструкторов базовых классов. Проверить, как это отразилось на работе программы и почему.
8. Изменить порядок наследования базовых классов в описании класса Derived и проверить, как это отразилось на работе программы.

Задание 2:

1. Задан базовый класс DomesticAnimal (домашнее животное), в котором определены три защищенных поля weight (вес), price (цена) и color (окраска). Класс снабжен конструктором без параметров и конструктором с тремя параметрами для инициализации трех полей класса. Кроме того, определена функция print, выводящая значения полей и сообщение о принадлежности функции к классу DomesticAnimal.
2. Производными от этого класса (public) являются классы Cow (корова) и Buffalo (бык), в которых не определено новых полей.
3. Класс Beefalo (теленок) является производным (public) от Cow и Buffalo. Его конструктор инициализирует поля weight, price и color без передачи параметров своим базовым классам Cow и Buffalo.
4. Классы Cow, Buffalo и Beefalo имеют свои функции print, которые

выводят сообщения о своей принадлежности к конкретному классу и выводят значения трех полей с помощью вызова print из DomesticAnimal.

5. Выявить и объяснить ошибки при компиляции, исправить программу.
6. В функции main описать переменные типа Cow и Beefalo (с инициализацией) и вызвать для них функцию print. Объяснить результаты.

Часть 2

Тема: Механизм виртуальных функций и его применение в программных проектах.

Перед выполнением лабораторной работы изучить тему вртуальных функций. Повторить процесс компиляции. Все задания сопроводить пояснениями. Каждое задание должно быть реализовано в виде одного .cpp файла.

Задание 1:

1. Рассматривается иерархия классов геометрических фигур. В качестве базового используется абстрактный класс Figure, в котором объявлены общие для всех фигур способности (виртуальные функции):
    1. double area(void), вычисляющая и возвращающая площадь соответствующей фигуры;
    2. void show(void), выводящая информацию о типе фигуры (круг, прямоугольник и т.п.), о заданных размерах фигуры (например, радиус для круга, или длины сторон прямоугольника) и величину площади фигуры.
2. Производные от Figure классы кругов (Circle) и прямоугольников (Rectangle). В классе Circle конструктор принимает один аргумент - радиус и проверяет, больше ли он нуля (при ошибке - выход из программы с соответствующим сообщением). В классе Rectangle

конструктор имеет один или два аргумента - длины сторон (квадрат и прямоугольник, причем функция show должна идентифицировать квадрат). Для проведения исследований в классах фигур должны быть описаны public - функции, возвращающие адреса каждого из полей данных (радиуса для круга или каждой из сторон прямоугольника).

3. В функции main:
    1. Создается произвольный набор конкретных фигур.
    2. Для каждого типа фигур вычисляется и выводится на экран размер одного объекта, а также адрес этого объекта и адреса его полей данных. Проанализировать результаты и дать им объяснения.
    3. Создать объект базового класса Figure и откомпилировать программу. Объяснить результат компиляции.
    4. Из адресов построенных фигур создать массив указателей на базовый класс. Организовать цикл, в котором выводится (с помощью виртуальной функции show) информация о каждой фигуре из массива.

Задание 2:

1. На базе программы задания 1 организовать проект figure с раздельно компилируемыми файлами, имеющий следующую структуру:
    1. Объявления классов Figure, Circle и Rectangle разместить в заголовочном файле figure.h.
    2. Определения функций-членов вынести за пределы объявлений классов и собрать в отдельном файле figure.cpp.
    3. Функцию main вынести в отдельный файл fig_main.cpp.
    4. Откомпилировать проект и убедиться в его работоспособности.
    5. Создать класс Triangle (треугольников), производный от класса Figure. Для этого создать новый заголовочный файл new_fig.h, подключающий файл figure.h и содержащий объявление класса Triangle. Конструктор класса Triangle имеет три аргумента (длины сторон). В нем

осуществляется проверка того, что из заданных элементов может быть составлен треугольник: сумма двух сторон должна быть больше третьей. При ошибке - выход из программы с соответствующим сообщением.

6. Реализации функций-членов класса Triangle размещаются в отдельном файле triangle.cpp. Площадь треугольника вычисляем по формуле Герона: S = sqrt(p*(p-a)*(p-b)*(p-c)), где p - полупериметр треугольника.
7. В функцию main вносятся следующие изменения. Изменяем подключаемый файл на new_fig.h, создаем объект класса Triangle и добавляем его в массиву казателей на объекты.
8. Добавляем в проект файл triangle.cpp, компилируем программу и проверяем ее работоспособность. В случае успеха заменяем в проекте файл triangle.cpp на объектный файл triangle.obj и проверяем работу нового проекта.