Лабораторная работа №5
по теме: "ИССЛЕДОВАНИЕ МЕХАНИЗМА ЕДИНИЧНОГО НАСЛЕДОВАНИЯ"

Цели работы:

1. создание простой иерархии классов и изучение методов инициализации объектов производных классов;
2. исследование конструкции объектов производных классов;
3. исследование уровней защищенности унаследованных компонент базового класса в объектах производного класса;
4. исследование методов доступа к одноименным функциям базового и производных классов.

ЗАДАНИЕ

1. Создайте базовый класс BASE, в котором опишите

- в разделе public поле int i ;
- в разделе protected поле long l;
- в разделе private поле double d.

Напишите конструктор, инициализирующий поля i, l и d тремя задаваемыми значениями.

Объясните различия и сходства между закрытыми (private) и защищенными (protected) членами класса.

2. Создате класс DERIVED, производный от класса Base (наследование типа public), в котором в разделе private опишите поле float f.

Напишите конструкторы класса DERIVED:

- конструктор без параметров;
- конструктор с 4-мя параметрами для инициализации всех полей объекта.

3. В функции main описать неинициализированный объект класса DERIVED и откомпилировать программу. Если возникнут проблемы, устранить их. Вывести размеры типов BASE и DERIVED и объяснить результаты.

4. Описать инициализированный объект класса Derived. Продемонстрировать, инициализацию каких полей, унаследованных от класса Base, можно выполнять с помощью присваивания непосредственно в конструкторе класса Derived. Какие поля класса Derived обязательно нужно инициализировать с помощью конструктора класса Base? Для исследования можно вносить необходимые изменения в конструкторы классов Base и Derived.

---

5. Перегрузить операцию вставки в поток для объектов класса Derived таким образом, чтобы выводились адреса и значения всех полей объекта. К каким полям, унаследованным от класса Base, нет доступа? Для снятия проблемы добавить в классе Base необходимые интерфейсные функции. Создав объект класса Derived, исследовать размещение полей в памяти. Привести схематическую структуру объекта.

6. Описать класс Derived_1, производный (public) от класса Derived и не имеющий новых полей. В классе описать конструктор со всеми необходимыми параметрами (сколько их нужно?).

Класс имеет общедоступную функцию void foo(), которая модифицирует значения полей, унаследованных от базового класса (i++; l+=1;). Откомпилировать программу. Заменить тип наследования Derived от Base на private и вновь откомпилировать программу. Какая возникла проблема? Для ее решения использовать возможность восстановления уровня доступа к компонентам базового класса.

7. Вернуть для Derived тип наследования public. На глобальном уровне и в классах Base и Derived описать функции void ff(), которые сообщают о своей принадлежности к классу или глобальному уровню. В функции foo класса Derived_1 добавить вызовы всех трех функций ff. В каких разделах классов Base и Derived нужно описать функции ff, чтобы они были доступны в Derived_1? Проверить работу программы, вызвав функцию foo для какого-либо объекта класса Derived_1.

8. Оставить в функции Derived::foo только один вызов в виде ff(); и проверить работу программы в следующих вариантах. Вначале функция ff определена в классах Derived, Base и на глобальном уровне. Затем ее описание убираем вначале из класса Derived, а затем из классов Derived и Base. Как в каждом случае это отражается на работе программы?