**Лаба2.1**

В данной программе на языке C++ реализована система копирования объектов с использованием механизма виртуальных функций и полиморфизма. Основной целью стало правильное копирование объектов базового и производного классов при помощи общей функции и предотвращение утечек памяти.

Ключевыми элементами являются классы Class1 и производный от него Class2. Каждый класс реализует метод copy(), возвращающий указатель на копию текущего объекта. В Class1 этот метод возвращает копию типа Class1, а в Class2 переопределяется и возвращает копию типа Class2. Это позволяет достичь полиморфного поведения, что является хорошей практикой при работе с иерархией классов.

Функция COPY() принимает ссылку на объект и вызывает его метод copy(), помещая полученный указатель в вектор database. Это обеспечивает добавление в коллекцию копий, а не оригиналов, что предотвращает проблемы с совместным владением ресурсами.

Вывод в консоль реализован с использованием typeid, что позволяет продемонстрировать, что копии сохраняют правильный тип во время выполнения. Это подтверждает корректность работы виртуальной функции copy() и демонстрирует возможности RTTI (Run-Time Type Information).

Также особое внимание уделено управлению памятью. Для отслеживания утечек используется макрос \_CRTDBG\_MAP\_ALLOC и структура Leaks, вызывающая \_CrtDumpMemoryLeaks() в деструкторе. Это гарантирует, что при завершении программы будет выведен отчет о возможных утечках памяти.

В финале программы вручную освобождаются все выделенные объекты с помощью delete, после чего вектор очищается.

**Лаба2.2**

В рамках данного задания требуется реализовать объектно-ориентированный подход при создании и отображении графических фигур в классическом приложении Windows. Задача делится на два ключевых этапа: создание объектов и их визуализация.

На первом этапе, в обработчике события WM\_CREATE функции WndProc, необходимо создать две фигуры: окружность и прямоугольник. Для этого проектируется иерархия классов, где базовый класс, например Shape, содержит виртуальный метод Draw(HDC hdc). Производные классы Circle и Rectangle переопределяют данный метод для отрисовки соответствующей фигуры. Каждая фигура создаётся с использованием конструктора, принимающего параметры (координаты, размеры и пр.), и сохраняется в общий глобальный контейнер std::vector<Shape\*>. Это позволяет сохранять объекты разных типов в одной коллекции, используя полиморфизм.

На втором этапе, при получении сообщения WM\_PAINT, между вызовами BeginPaint и EndPaint происходит обход глобального вектора и вызов метода Draw(hdc) для каждого объекта. Таким образом, каждая фигура отвечает за собственную отрисовку, что соответствует принципам инкапсуляции и разделения ответственности.

Такой подход обеспечивает высокую расширяемость программы. Для добавления новых типов фигур достаточно унаследовать их от базового класса и реализовать метод отрисовки. Также программа сохраняет чистоту архитектуры, соблюдая принципы ООП.