**Лаба3.1**

В процессе выполнения данного задания предполагается создание проекта "Классическое приложение Windows" с применением объектно-ориентированного подхода к работе с графикой. Основная задача заключается в реализации вспомогательного класса Vector2D, предназначенного для описания точек и двумерных векторов, с последующим использованием его в визуализации геометрических фигур.

Класс Vector2D представляет собой универсальный инструмент для работы с координатами на плоскости. Он инкапсулирует координаты x и y и предоставляет дружественные функции-операторы для выполнения арифметических операций, таких как сложение, вычитание, масштабирование и присваивание. Это позволяет удобно использовать данный класс при вычислениях позиций объектов на экране. Особое внимание уделено реализации оператора присваивания и конструкторам копирования, что обеспечивает корректную работу при передаче и возвращении объектов этого класса.

В рамках обработки события WM\_PAINT с помощью объекта HDC, предоставляемого функцией BeginPaint, происходит отрисовка девяти окружностей, размещённых в виде сетки 3x3. Для задания координат каждой окружности используются объекты класса Vector2D, которые задают как начальную точку, так и векторы смещения по горизонтали и вертикали. Таким образом, построение точек для функции Ellipse реализуется через арифметику над объектами Vector2D, что значительно упрощает логику и делает код более читаемым и масштабируемым.

Такой подход позволяет легко модифицировать расположение и количество фигур, а также переиспользовать класс Vector2D в других графических задачах, связанных с трансформациями и позиционированием элементов.

**Лаба3.2**

В данной работе реализован пример объектно-ориентированного программирования на языке C++ с использованием наследования, виртуальных функций и шаблонов. Цель программы — продемонстрировать создание объектов разных типов на основе общего базового класса, а также правильное управление памятью.

Ключевыми классами являются Detail и производный от него класс Assembly. Оба класса содержат поле name, которое инициализируется через конструкторы. Метод show() реализован как виртуальный, что позволяет полиморфно выводить информацию об объекте вне зависимости от его конкретного типа.

Особенность реализации заключается в использовании шаблонной функции createObject, которая принимает вектор указателей на базовый класс Detail и создает объект заданного типа (например, Detail или Assembly) по введенному пользователем имени. Таким образом достигается универсальность и повторное использование кода при создании различных объектов.

Объекты сохраняются в векторе arr, который затем используется для вывода информации о каждом объекте с помощью полиморфного вызова метода show(). После использования, все объекты корректно удаляются через delete, что предотвращает утечки памяти.

Программа демонстрирует важные принципы ООП: инкапсуляцию, наследование и полиморфизм. Использование шаблонной функции также показывает, как можно комбинировать обобщённое программирование с классической иерархией классов для создания гибкого и расширяемого кода.