**Список лабораторных работ**

по дисциплине

«Объектно-ориентированный анализ и проектирование»

осенний семестр 2024/2025 уч. года

**ЛР №1: Основы языка C++. Исследование модульного подхода к созданию программ**

*Цель работы:* актуализировать знания о разработке программ на языке C++. Изучить модульный подход к разработке программ. Работа в пакетах Qt Creator и Visual Studio.

*Задание на работу:* создать комплексное решение, объединяющее несколько проектов с помощью IDE Visual Studio и Qt Creator. Реализовать три консольных приложения, в соответствии с вариантом задания. Для первого задания использовать жесткое кодирование входных параметров, для второго задания использовать ввод через консоль, для третьего – ввод исходных параметров и вывод результатов работы в текстовые файлы.

**ЛР №2: Создание пользовательского класса на языке C++. Исследование механизмов инкапсуляции**

*Цель работы:* разработать пользовательский класс для выбранной предметной области (или, заданной по номера варианта). Изучить механизм инкапсуляции.

*Задание на работу:* реализовать заданную структуру в виде программного кода. Атрибуты класса должны быть закрытыми, доступ к атрибутам осуществляется с помощью *get* и *set* методов. Атрибуты класса должны использовать максимально возможное количество типов данных языка C++ (включая контейнеры или массивы). В основной программе необходимо создать несколько экземпляров разработанного класса, инициализировать заданные атрибуты, вывести информацию об экземпляре класса на экран. Обосновать использования инкапсуляции для разработки пользовательского класса.

**ЛР №3: Динамическое создание объектов пользовательских классов с помощью операторов new и delete. Использование работы конструкторов и деструкторов. Работа с умными указателями.**

*Цель работы:* изучить особенности работы с экземплярами классов с помощью динамического выделения памяти. Изучить механизм работы конструкторов и деструкторов объектов. Изучить работу с умными указателями.

*Задание на работу:* доработать класс из ЛР № 2, добавив в него конструкторы (по умолчанию, с параметрами, копирования) и деструктор. Создание и удаление объекта класса выполнить с помощью операторов new и delete соответственно. Проанализировать преимущества работы с динамической памятью. Изучить механизм использования умных указателей. Сравнить работу умных указателей с работой обычных указателей.

**ЛР №4: Создание иерархии классов. Исследование механизмов наследования.**

*Цель работы:* научится создавать базовые классы и классы наследники. Исследовать механизм наследования.

*Задание на работу:* для классов, созданных в ЛР № 2–3 продумать иерархию объектов. Разработать базовый класс и несколько классов наследников. В базовом классе выделить общие свойства и методы для объектов исследуемой предметной области, в классах наследниках сосредоточить уникальные свойства, описывающие объекты, порождённые от базового класса. Исследовать свойства декомпозиции на примере разработанной иерархии, доказать ее целесообразность. При реализации методов базового класса и классов наследников предусмотреть различные варианты обеспечении доступа к методам и атрибутам базового класса. На примере продемонстрировать механизм наследования методов базового класса, механизм наследования конструкторов. Проиллюстрировать как работает передача параметров конструктора в конструктор класса наследника и базового класса.

**ЛР №5: Создание полиморфных функций. Исследование механизмов полиморфизма**

*Цель работы:* научиться создавать полиморфные функции. Изучить механизм полиморфизма.

*Задание на работу:* для классов, созданных в ЛР № 2–4 модифицировать базовый класс таким образом, чтобы в нем была реализована полиморфная функция. Реализацию выполнить несколькими способами: с помощью виртуальных функций и с помощью чисто виртуальных функций. Создать абстрактный класс. Обосновать необходимость использования механизма полиморфизма для разработанной иерархии классов. Объяснить, что такое динамический полиморфизм.

**ЛР №6: Перегрузка операторов**

*Цель работы:* изучить механизм перегрузки операторов для объектов пользовательского класса.

*Задание на работу:* для классов, разработанных в ЛР № 2–5 реализовать перегрузку операторов (до двух-трех операторов). Проиллюстрировать работу перегруженных операторов на примере.