

ЛАБОРАТОРНАЯ РАБОТА №1

Работа с проектами в среде Borland C.

Цель работы

Изучить среду программирования, освоить навыки работы с проектами. Восстановить навыки работы со структурами, освоить приемы использования функций - членов структур.

Порядок выполнения работы

- 4.1. Разработать программу для работы со структурами согласно варианту.
- 4.2. Разработать демонстрационную программу, перенеся объявление структурных переменных и вызов соответствующих функций в отдельный модуль, объединить модули в проекте.
- 4.3. Модифицировать программу, используя функции-члены.

Варианты заданий

Вариант 1.

Построить программу для работы со структурой Дата. Программа должна обеспечивать простейшие функции для работы с данными структурами: увеличение/уменьшение на 1 день, ввод значений, вывод значений.

Вариант 2.

Построить программу для работы со структурой Время. Программа должна обеспечивать простейшие функции для работы с данными структурами: увеличение/уменьшение на 1 час, минуту, секунду, ввод значений, вывод значений.

Вариант 3.

Построить программу для работы со структурами - строками. Структура должна включать следующие поля: массив для хранения строки, его длину, время создания строки. Программа должна обеспечивать простейшие функции для работы с данными структурами: изменение строки, вывод строки, нахождение подстроки в строке.

Вариант 4.

Построить программу для работы со структурами - окнами. Структура должна включать соответствующие поля: размер окна, его положение на экране, цвет. Программа должна обеспечивать простейшие функции для работы с данными структурами: отображение окна, удаление окна, изменение цветов.

Вариант 5.

Построить программу для работы со структурами - многочленами. Структура должна включать соответствующие поля: порядок, набор коэффициентов. Программа должна обеспечивать простейшие функции для работы с данными структурами: вычисление значения многочлена для данного параметра, вывод многочлена в удобной форме.

Вариант 6.

Построить программу для работы со структурами - квадратными матрицами. Структура должна включать соответствующие поля: порядок, набор коэффициентов. Программа должна обеспечивать простейшие функции для работы с данными структурами: ввод матрицы, транспонирование матрицы, вывод матрицы в удобной форме.

Вариант 7.

Построить программу для работы со структурами - правильными дробями. Структура должна включать соответствующие поля: числитель, знаменатель. Программа должна обеспечивать простейшие функции для работы с данными структурами: сложение, вычитание, умножение, деление, вывод дроби в удобной форме.

Вариант 8.

Построить программу для работы со структурами - комплексными числами. Структура должна включать соответствующие поля: вещественную и мнимую часть числа. Программа должна обеспечивать простейшие функции для работы с данными структурами: сложение, вычитание, умножение, деление, вывод числа в удобной форме.

Контрольные вопросы

- 6.1. Классы и объекты
- 6.2. Методы
- 6.3. Описание и вызов методов класса

ЛАБОРАТОРНАЯ РАБОТА №2

Использование классов в C++.

Цель работы

Изучить способы инкапсуляции данных, особенности различных типов конструкторов, деструкторы. Освоить операции new и delete для работы с динамической памятью.

Порядок выполнения работы

- 4.1. Оформить программу п.4.3 из лабораторной работы 1 с использованием классов, способов разграничения доступа к данным.
- 4.2. Создать перегружаемые конструкторы, конструктор по умолчанию.
- 4.3. Создать конструктор копирования с динамическим выделением памяти, деструктор.
- 4.4. Добавить в демонстрационную программу инициализацию объектов с помощью различных конструкторов.
- 4.5. Выполнить программу в пошаговом режиме для контроля вызова деструктора.

Контрольные вопросы

- 5.1. Чем отличается class от struct?
- 5.2. Что такое инкапсуляция?
- 5.3. Что такое конструктор, зачем он нужен, когда вызывается?
- 5.4. Что такое деструктор, зачем он нужен, когда вызывается?
- 5.5. В чем особенность и когда нужен конструктор копирования?
- 5.6. Операции new и delete.
- 5.7. This, его назначение и применение

ЛАБОРАТОРНАЯ РАБОТА №3

Переопределение операций.

Цель работы

Освоить приемы написания переопределенных операций - членов и друзей класса, оперирующих объектами класса и переменными базовых типов данных.

Порядок выполнения работы

- 4.1. Переписать программу из лабораторной работы 2, включив в нее функции (члены класса и его друзья) для выполнения операций сложения переменной базового типа с объектом, сложения двух объектов класса, вычитания из объекта класса переменной базового типа.
- 4.2. Добавить в программу переопределенные операции префиксного и постфиксного инкремента.
- 4.3. Изменить демонстрационную программу, включив в нее перегруженные операции.

Контрольные вопросы

- 5.1. Какова цель перегрузки функций и операций?
- 5.2. Что такое дружественные функции, как они описываются и вызываются?
- 5.3. Какие операции должны быть функциями - друзьями, а какие - только членами?
- 5.4. Чем отличаются операции «объект + переменная» от «переменная + объект»?
- 5.5. Чем отличаются операции префиксного и постфиксного инкремента
- 5.6. Особенности постфиксного инкремента и декремента.
- 5.7. Способы возврата результата переопределенной операции.
- 5.8. Какие операции нельзя переопределять?
- 5.9. Операции преобразования типов.

ЛАБОРАТОРНАЯ РАБОТА №4

Производные классы.

Цель работы

Освоить приемы построения производных классов, создания и применения виртуальных функций.

Варианты заданий

Вариант 1 - Класс «записная книжка», включающий ФИО, телефон, дату рождения и функцию вычисления количества дней до дня рождения;

Вариант 2 - Класс «расписание», включающий дисциплину, аудиторию, время начала и функцию вычисления времени до начала занятия;

Вариант 3 - Класс «строка», включающий строку, время создания, текущее время и функцию вычисления времени существования строки;

Вариант 4 - Класс «окно», включающий координаты и цвет окна, текст в окне, цвет текста и функцию изменения текста в окне;

Вариант 5 - Класс «передаточная функция», включающий полином в числителе, полином в знаменателе и функцию деления полиномов;

Вариант 6 - Класс «матрица», включающий матрицу, ее определитель и функцию вычисления определителя;

Вариант 7 - Класс «дробное комплексное число», включающий дробную вещественную часть, дробную мнимую часть и арифметические операции над комплексными дробями;

Вариант 8 - Класс «комплексное число», включающий комплексное число в стандартной и экспоненциальной форме и функцию вычисления экспоненциальной формы числа.

Контрольные вопросы

- 6.1. Как наследуются компоненты с правами доступа `public`, `private` и `protected` в производный класс и структуру?
- 6.2. Как устранить неоднозначности при множественном наследовании?
- 6.3. Наследование и порядок вызова конструкторов и деструкторов в производном классе.
- 6.4. Что такое виртуальные функции, как они описываются и вызываются?
- 6.5. Что такое чистые виртуальные функции и абстрактные классы?
- 6.6. Какие функции могут и не могут быть виртуальными?
- 6.7. Как подавить виртуальность функции?

ЛАБОРАТОРНАЯ РАБОТА №5

Ввод-вывод в потоки.

Цель работы

Освоить применение операций ввода - вывода в потоки, перегрузку операций ввода - вывода, применение статических функций - членов класса.

Порядок выполнения работы

- 4.1. Для заданий, выполненных в предыдущих работах, перегрузить операции ввода - вывода объектов класса, разработать функции, позволяющие осуществлять ввод и вывод отдельного объекта и всего списка объектов класса с/на консоль и в/из текстового файла в удобной форме.
- 4.2. Изменить демонстрационную программу, включив в нее вызов разработанных функций, конструкторов и деструкторов для изменения количества объектов.

Контрольные вопросы

- 5.1. Стандартные потоки в С.
- 5.2. Иерархия потоковых классов.
- 5.3. Операции помещения в поток и извлечения из потока, их использование.
- 5.4. Как проконтролировать состояние потока?
- 5.5. Файловые потоки
- 5.6. Режимы открытия файлов
- 5.7. Функции ввода - вывода и позиционирования в файлах.
- 5.8. Статические члены и методы класса.

ЛАБОРАТОРНАЯ РАБОТА №6

Обработка исключительных ситуаций на C++.

Цель работы

Освоить приемы обработки исключений при возникновении математических ошибок и ошибок недостатка памяти, прерываний операционной системы.

Порядок выполнения работы

- 4.1. Для заданий, выполненных в предыдущих работах, перегрузить операции ввода - вывода объектов класса, разработать функции, позволяющие осуществлять ввод и вывод отдельного объекта и всего списка объектов класса с/на консоль и в/из текстового файла в удобной форме.
- 4.2. Изменить демонстрационную программу, включив в нее вызов разработанных функций, конструкторов и деструкторов для изменения количества объектов.

Контрольные вопросы

- 5.1. Стандартные потоки в C.
- 5.2. Иерархия потоковых классов.
- 5.3. Операции помещения в поток и извлечения из потока, их использование.
- 5.4. Как проконтролировать состояние потока?
- 5.5. Файловые потоки
- 5.6. Режимы открытия файлов
- 5.7. Функции ввода - вывода и позиционирования в файлах.
- 5.8. Статические члены и методы класса.

Для задач из предыдущей лабораторной работы добавить обработку исключений при возникновении ошибок: недостатка памяти, выход за пределы диапазона допустимых значений и т.д. Обработку исключений реализовать на уровне методов и демонстрационной программы. Изменить демонстрационную программу так, чтобы она демонстрировала механизм исключений.