

Лабораторная работа №1

Тема работы: Введение в C++.

Цель работы: Создание консольной программы на языке C++ с целью изучения общей структуры консольной программы, знакомства с основами системы ввода-вывода и повторения изученного ранее материала по работе со строками вида `char*` в C++.

Общие требования к выполнению работы

1. Изучить особенности создания консольных программ на языке C++ (структура, встроенные типы данных, операторы ввода-вывода, библиотеки ввода-вывода и работы со строками вида `char*`).
2. Создать проект консольной программы.
3. Реализовать программу, которая выполняет консольный ввод-вывод согласно заданию.
4. В программе нельзя использовать старую систему ввода-вывода данных языка C (`printf()` / `fprintf()` и т.п.). Ввод и вывод информации вести только с использованием потоков ввода-вывода и соответствующих операторов.
5. Для упрощения работы со строчными данными считать, что строка не может быть более 80 символов.
6. Предусмотреть в программе реакцию на ошибочные действия пользователя при вводе данных (например, организация повтора ввода).
7. Отладить и выполнить полученную программу.
8. Создать отчет, включающий задание, листинг программы и результаты ее выполнения.

Контрольные вопросы

1. Структура консольной программы в C++.
2. Библиотека ввода-вывода в C++.
3. Потоки для консольного ввода-вывода в C++.
4. Операторы для консольного ввода-вывода в C++.
5. Работа со строками вида `char*` в C++.

Варианты заданий

1. Написать программу «Hello, world!».
2. Написать программу для ввода с клавиатуры строки символов и вывода на экран числа введенных символов.
3. Написать программу для ввода с клавиатуры строки символов и вывода на экран этой строки в обратном порядке.
4. Написать программу для ввода с клавиатуры строки символов и вывода на экран этой строки в зашифрованном виде.
5. Написать программу для ввода с клавиатуры двух строк символов и вывода на экран большей строки по длине.
6. Написать программу для ввода с клавиатуры двух строк символов и вывода на экран большей строки по сравнению значений.
7. Написать программу для ввода с клавиатуры трех строк символов и вывода на экран номера строки, в которой есть заданный символ.
8. Написать программу для вывода на экран аргументов командной строки, номер аргумента для вывода пользователь указывает в диалоге.
9. Написать программу простейшего калькулятора, в котором числа и знак операции вводятся в форме диалога.
10. Написать программу с реализацией определенных действий через консольное меню (не менее 3 пунктов).
11. Написать программу, имитирующую простой диалог с компьютером (например, запрос имени пользователя и 3 вопроса к пользователю с использованием этого имени в обращении).
12. Написать программу, реализующую пошаговую игру в которой информация предоставляется в форме диалога (например, движение по лабиринту).

Лабораторная работа №2

Тема работы: Создание базовых объектов.

Цель работы: Создание консольной программы включающей объекты и реализующей члены объектов с использованием спецификаторов доступа, конструкторы и деструктор, статические члены классов, особую инициализацию членов класса, дружественные функции и классы.

Общие требования к выполнению работы

1. Рассмотреть теоретические сведения по теме лабораторной работы.
2. Создать проект консольной программы.
3. Создать базовый класс согласно заданию. Все данные класса описать в закрытой секции, все методы класса реализовать как открытые. Создать конструктор по умолчанию и конструктор инициализации. Создать деструктор.
4. Вне класса должна быть создана дружественная функция, предназначенная для вывода данных объекта на экран.
5. Для задания значений уникальных номеров использовать или статические члены класса или особую инициализацию членов класса.
6. Установка значений переменных должна иметь проверку на реальный диапазон значений.
7. В головной функции создать несколько одиночных экземпляров базового класса. Обработать все экземпляры базового класса с помощью заданного метода дружественного класса. Вывести содержимое переменных экземпляров классов на экран до и после этой обработки.
8. В головной функции создать массив экземпляров базового класса. Обработать все элементы массива с помощью заданного метода дружественного класса. Вывести содержимое переменных элементов массива на экран до и после этой обработки.
9. Отладить и выполнить полученную программу.
10. Создать отчет, включающий задание, листинг программы и результаты ее выполнения.

Контрольные вопросы

1. Структура базового класса и спецификаторы доступа.
2. Конструкторы и деструктор.
3. Статические члены класса.
4. Особая инициализация членов класса.
5. Дружественные функции и классы.

Варианты заданий

1. Создать класс «Студент» со следующими данными: id – уникальный номер, name – ФИО, rate – рейтинг. Создать дружественный класс «Декан», в котором реализовать метод установки рейтинга студента.
2. Создать класс «Деталь» со следующими данными: id – уникальный номер, name – наименование, mass – вес. Создать дружественный класс «Контроль», в котором реализовать метод установки веса детали.
3. Создать класс «Сотрудник» со следующими данными: id – уникальный номер, name – ФИО, salary – оклад. Создать дружественный класс «Бухгалтер», в котором реализовать метод установки оклада сотрудника.
4. Создать класс «Автомобиль» со следующими данными: id – уникальный номер, name – ФИО владельца, mileage – пробег. Создать дружественный класс «СТО», в котором реализовать метод установки пробега автомобиля.
5. Создать класс «Товар» со следующими данными: id – уникальный номер, name – название, cost – стоимость. Создать дружественный класс «Продавец», в котором реализовать метод установки стоимости товара.
6. Создать класс «Задание» со следующими данными: id – уникальный номер, name – название, result – процент выполнения. Создать дружественный класс «Проверяющий», в котором реализовать метод установки процента выполнения задания.

Лабораторная работа №3

Тема работы: Перегрузка функций и операторов.

Цель работы: Создание консольной программы реализующей перегрузку функции (в том числе с параметрами по умолчанию), перегрузку унарных и бинарных операторов, (в том числе и с помощью дружественных функций), реализацию функции преобразования типов.

Общие требования к выполнению работы

1. Рассмотреть теоретические сведения по теме лабораторной работы.
2. Создать проект консольной программы.
3. Создать базовый класс согласно заданию. Все данные класса описать в закрытой секции, все методы класса реализовать как открытые.
4. Создать конструктор инициализации с параметрами по умолчанию. Остальные конструкторы и деструктор создать по необходимости.
5. Перегрузить операторы-члены класса:
 - присвоение =;
 - ~~-- индексация элемента [];~~
 - инициализация ();
 - унарные арифметические (++);
 - бинарные арифметические (+ или -) и сравнения (> или < или ==), причем только для однотипных операндов;
6. Перегрузить дружественные функции-операторы:
 - унарные арифметические (--);
 - бинарные арифметические (+ или -) и сравнения (> или < или ==), причем для разных типов операндов;
 - вывода в поток (<<).
7. Создать функции преобразования:
 - во встроенный тип (int, double, ...) (например, для подсчета характеристик массива (max, min, avg, count));
 - в пользовательский тип (например, класс описывающий размерность массива).
8. В головной функции создать несколько экземпляров базового класса. Провести вызовы перегруженных функций и операторов. Вывести результаты на экран.
9. Отладить и выполнить полученную программу.
10. Создать отчет, включающий задание, листинг программы и результаты ее выполнения.

Контрольные вопросы

1. Перегрузка функции.
2. Параметры функции по умолчанию.
3. Перегрузка унарных операторов.
4. Перегрузка бинарных операторов.
5. Функции преобразования типов.

Варианты заданий

1. «Целое число».
2. «Целое положительное число».
3. «Дробное число».
4. «Комплексное число».
5. «Строка».
6. «Массив целых чисел».
7. «Массив дробных чисел».

8. «Массив комплексных чисел».
9. «Матрица целых чисел».
10. «Матрица дробных чисел».
11. «Матрица комплексных чисел».

~~Лабораторная работа №4~~

Тема работы: Динамическое создание объектов и массивов объектов.

Цель работы: Создание консольной программы с использованием классов, внутри которых память под данные выделяется динамически, рассмотрение перегрузки операторов new и delete, динамическое создание одиночных экземпляров объектов и массивов экземпляров объектов в программе.

Общие требования к выполнению работы

1. Рассмотреть теоретические сведения по теме лабораторной работы.
2. Создать проект консольной программы.
3. Создать базовый класс согласно заданию. Все данные класса описать в закрытой секции, все методы класса реализовать как открытые.
4. Создать конструктор по умолчанию, конструктор инициализации и конструктор копирования. Конструкторы должны выделять память для данных динамически с использованием оператора new.
5. Создать деструктор, который будет освобождать динамически выделенную память с помощью оператора delete.
6. Создать в классе методы доступа к одиночному элементу данных объекта (по чтению и записи), отображения данных на экран, преобразования данных объекта (например, сортировки).
7. Перегрузить операторы-члены класса для выделения памяти (new и delete).
8. В головной функции динамически создать и проинициализировать экземпляр базового класса. Динамически создать его копию. Выполнить преобразование данных экземпляра базового класса. Вывести содержимое обоих экземпляров на экран. Удалить экземпляры.
9. В головной функции динамически создать массив экземпляров класса. Каждый из них проинициализировать. Выполнить доступ к элементам данных экземпляров классов. Удалить массивы.
10. Отладить и выполнить полученную программу.
11. Создать отчет, включающий задание, листинг программы и результаты ее выполнения.

Контрольные вопросы

1. Конструктор копирования.
2. Оператор new и его перегрузка.
3. Оператор delete и его перегрузка.
4. Динамическое создание экземпляров классов.
5. Динамическое создание массивов экземпляров классов.

Варианты заданий

1. Одномерный массив с элементами типа char (строка).
2. Одномерный массив с элементами типа int.
3. Одномерный массив с элементами типа double.
4. Одномерный массив с элементами пользовательского типа (например, complex).
5. Двумерный массив (матрица) с элементами типа char.
6. Двумерный массив (матрица) с элементами типа int.
7. Двумерный массив (матрица) с элементами типа double.
8. Двумерный массив (матрица) с элементами пользовательского типа (complex).
9. Трехмерный массив с элементами типа char.
10. Трехмерный массив с элементами типа int.
11. Трехмерный массив с элементами типа double.
12. Трехмерный массив с элементами пользовательского типа (например, complex).

Лабораторная работа №5

Тема работы: Наследование.

Цель работы: Создание консольной программы для реализации иерархии классов с использованием простого и множественного наследования, исследование спецификаций наследования, выполнение передачи параметров конструктору базового класса, разрешение с помощью виртуального наследования неоднозначностей при множественном наследовании.

Общие требования к выполнению работы

1. Рассмотреть теоретические сведения по теме лабораторной работы.
2. Создать проект консольной программы.
3. По полученному базовому классу создать классы наследников по двум разным ветвям наследования ($B \leftarrow P1 \leftarrow P11$ и $B \leftarrow P2 \leftarrow P21$):
 - во всех классах должны быть свои данные (характеристики объектов), расположенные в `private` или `protected` секциях;
 - во всех классах создать конструкторы инициализации объектов для всех классов (не забыть про передачу параметров в конструкции базовых классов);
 - во всех классах создать деструкторы;
 - во всех классах создать метод вывода объекта на экран (всех его параметров);
 - остальные методы создавать по необходимости.
4. Создать класс с множественным наследованием ($P11 \leftarrow P3 \rightarrow P21$). Создать в этом классе выше описанные функции (в коде функций не использовать доступ к данным через оператор разрешения зоны видимости ::).
5. В головной функции создать экземпляр класса `P3`. Вывести содержимое экземпляра класса `P3` на экран. Проверить, что получится, если не будет использовано виртуальное наследование.
6. Отладить и выполнить полученную программу.
7. Создать отчет, включающий задание, листинг программы и результаты ее выполнения.

Контрольные вопросы

1. Множественное наследование.
2. Неоднозначности при множественном наследовании.
3. Передача параметров базовому классу.
4. Виртуальное наследование.
5. Оператор разрешения зоны видимости.

Варианты заданий

1. «Человек».
2. «Учащийся».
3. «Работник».
4. «Транспортное средство».
5. «Инструмент».
6. «Продукт».
7. «Дом».
8. «Книга».
9. «Телевизор».
10. «Компьютер».
11. «Фигура».
12. «Завод».
13. «Животное».

14. «Растение».
15. «Гриб».
16. «Вирус».

Лабораторная работа №6

Тема работы: Виртуальные функции и абстрактные классы.

Цель работы: Создание консольной программы для реализации абстрактных классов и чисто виртуальных функций, создания и наследования виртуальных функций, изучения статического и динамического полиморфизма на основе виртуальных функций, включая виртуальные деструкторы.

Общие требования к выполнению работы

1. Рассмотреть теоретические сведения по теме лабораторной работы.
2. Создать проект консольной программы.
3. По полученному базовому классу создать классы наследников по двум разным ветвям наследования ($B \leftarrow P1 \leftarrow P11$ и $B \leftarrow P2$):
 - во всех классах должны быть свои данные (характеристики объектов), расположенные в `private` или `protected` секциях;
 - во всех классах создать конструкторы инициализации объектов для всех классов (не забыть про передачу параметров в конструкции базовых классов);
 - во всех классах создать деструкторы;
 - остальные методы создавать по необходимости.
4. Создать в базовом классе чисто виртуальные функции расчета (например, расчет площади фигуры и т.п.) и вывода объекта на экран (всех его параметров). Выполнить реализацию этих виртуальных функций в классах наследниках.
5. Задать в базовом классе деструктор как виртуальный.
6. В головной функции динамически создать массив указателей на базовый класс. Заполнить массив указателями на динамически создаваемые объекты производных классов ($P1, P11, P2$). Для каждого элемента массива проверить работу виртуальных функций. Удалить из памяти динамически выделенные объекты.
7. Отладить и выполнить полученную программу. Проверить, что будет, если выше описанные методы не будут виртуальными.
8. Создать отчет, включающий задание, листинг программы и результаты ее выполнения.

Контрольные вопросы

1. Виртуальные функции.
2. Абстрактные классы.
3. Наследование виртуальных функций.
4. Виртуальный деструктор.
5. Статический и динамический полиморфизм.

Варианты заданий

1. «Человек».
2. «Учащийся».
3. «Работник».
4. «Транспортное средство».
5. «Инструмент».
6. «Продукт».
7. «Дом».
8. «Книга».
9. «Телевизор».
10. «Компьютер».
11. «Фигура».
12. «Завод».
13. «Животное».

14. «Растение».
15. «Гриб».
16. «Вирус».

Лабораторная работа №7

Тема работы: Шаблоны функций и классов.

Цель работы: Создание консольной программы для реализации шаблонов функций и классов, включая: явные специализации шаблонных функций и классов, стандартные типы в шаблонных классах.

Общие требования к выполнению работы

1. Рассмотреть теоретические сведения по теме лабораторной работы.
2. Создать проект консольной программы.
3. Создать шаблонную функцию ~~или класс~~ согласно заданию.
4. Реализовать явную специализацию шаблонной функции ~~или класса~~ для строк символов (char *).
5. ~~В шаблонных классах использовать стандартные типы и аргументы по умолчанию. Данные в шаблонных классах хранить в статическом массиве. Набор методов класса должен обеспечить основные операции с элементами объекта (добавление, удаление, поиск, просмотр, упорядочивание и т.п.) и операции над однотипными объектами (сравнение, сложение и т.п.).~~
6. ~~Дополнительно рассмотреть реализацию наследования шаблонных классов (например, от простого базового класса).~~
7. В функции main выполнить действия с шаблонной функцией ~~или классом~~, которые продемонстрируют работу методов.
8. Отладить и выполнить полученную программу. Проверить использование специализаций, стандартных типов и аргументов по умолчанию.
9. Создать отчет, включающий задание, листинг программы и результаты ее выполнения.

Контрольные вопросы

1. Шаблоны функций.
2. Шаблоны классов.
3. Явные специализации шаблонных функций.
4. Явные специализации шаблонных классов.
5. Стандартные типы в шаблонных классах.

Варианты заданий

1. Функция сортировки элементов массива.
2. Функция поиска максимального элемента в массиве.
3. Функция перестановки наоборот элементов в массиве.
4. Функция поиска первого элемента в массиве по заданному ключу.
5. Функция замены всех подобных элементов в массиве по заданному ключу.
6. Функция бинарного поиска первого элемента в отсортированном массиве по заданному ключу.
7. Класс «Массив».
8. Класс «Матрица»
9. Класс «Стек».
10. Класс «Очередь».
11. Класс «Список».
12. Класс «Множество».

Лабораторная работа №8

Тема работы: Обработка исключительных ситуаций.

Цель работы: Создание консольной программы для реализации обработки ошибок с использованием механизма обработки исключительных ситуаций, создание специальных классов обработки исключительных ситуаций, рассмотрение замены обработчиков функций `terminate()` и `unexpected()` собственными функциями.

Общие требования к выполнению работы

1. Рассмотреть теоретические сведения по теме лабораторной работы.
2. Создать проект консольной программы.
3. Создать обычный или шаблонный класс согласно заданию. Данные в классе хранить в статическом массиве. Набор методов класса должен обеспечить основные операции с элементами объекта (добавление, удаление, поиск, просмотр и т.п.) и операции над однотипными объектами (сравнение, присвоение и т.п.).
4. Обработку всех ошибочных ситуаций реализовать с использованием схемы обработки исключительных ситуаций. Реализовать собственный класс обработки исключительной ситуации.
5. Рассмотреть замену обработчиков функций `terminate()` и `unexpected()` собственными функциями.
6. В функции `main` выполнить действия с экземпляром класса, которые продемонстрируют работу методов, в том числе и реакцию на возникновение ошибок.
7. Отладить и выполнить полученную программу. Проверить работу обработки исключительных ситуаций.
8. Создать отчет, включающий задание, листинг программы и результаты ее выполнения.

Контрольные вопросы

1. Схема обработки исключительных ситуаций в C++.
2. Оператор `throw`.
3. Реализация обработчиков исключительной ситуации `catch()`.
4. Классы исключительных ситуаций.
5. Функции `terminate()` и `unexpected()`.

Варианты заданий

1. Класс «Массив».
2. Класс «Матрица».
3. Класс «Стек».
4. Класс «Очередь».
5. Класс «Список».
6. Класс «Множество».

Лабораторная работа №9

Тема работы: Система ввода-вывода и работа с файлами.

Цель работы: Создание консольной программы для реализации работы с данными, которые хранятся в текстовых и бинарных файлах с использованием: перегрузки операторов ввода-вывода, управления курсорами чтения и записи, с обработкой ошибок состояния потока ввода-вывода, а также вывод данных на консоль с использованием возможностей управления форматом флагами и манипуляторами, создание манипуляторов.

Общие требования к выполнению работы

1. Рассмотреть теоретические сведения по теме лабораторной работы.
2. Создать проект консольной программы.
3. Создать класс, согласно заданию, который будет описывать некоторый объект (задать не менее трех характеристик) и базовые методы для работы с ним.
4. Создать в этом классе операторы работы с потоками ввода-вывода, ориентированными на работу с текстовыми файлами.
5. Создать в этом классе методы (чтение, запись, ...) для работы с текстовыми файлами в бинарном режиме (на основе методов потока для работы с отдельными символами).
6. Создать в этом классе методы (чтение, запись, ...) для работы с бинарными файлами в бинарном режиме (на основе методов потока для работы с блоками памяти).
7. Для обработки всех ошибочных ситуаций использовать конструкцию try...catch().
8. В головной функции выполнить обработку данных в трех режимах:
 - работа с текстовым файлом в текстовом режиме;
 - работа с текстовым файлом в бинарном режиме;
 - работа с бинарным файлом в бинарном режиме;Для каждого выше указанного варианта выполнить следующие действия:
 - сформировать файл данных (не менее 5 строк или записей);
 - вывести записи данных в прямом и обратном порядке не считывая все записи в память (при вводе-выводе данных применить знания флагов вывода и манипуляторов);
 - найти записи по условию (по шаблону названия) и вывести их на экран;
 - удалить записи по условию в файле, удаляемые записи вывести на экран.
9. Отладить и выполнить полученную программу. Проверить обработку исключительных ситуаций.
10. Создать отчет, включающий задание, листинг программы и результаты ее выполнения.

Контрольные вопросы

1. Потоки ввода-вывода.
2. Флаги и манипуляторы формата.
3. Файловые потоки.
4. Работа с текстовыми файлами.
5. Работа с бинарными файлами.

Варианты заданий

1. «Студент».
2. «Сотрудник».
3. «Турист».
4. «Организация».
5. «Блюдо».
6. «Заметка».
7. «Деталь».

8. «Автомобиль».
9. «Аэропорт».
10. «Город».
11. «Остановка».
12. «Растение».

Лабораторная работа №10

Тема работы: Контейнеры объектов.

Цель работы: Создание консольной программы с реализацией объектов для хранения данных в динамической памяти на основе шаблонных классов, изучение указателей на объекты.

Общие требования к выполнению работы

1. Рассмотреть теоретические сведения по теме лабораторной работы.
2. Создать проект консольной программы.
3. Построить шаблонный класс, который будет описывать элемент хранимых данных, доступ к ним, сравнение элементов и т.п. по необходимости.
4. Для полученного варианта задания, построить контейнерный шаблонный класс операций над элементами данных, включающий операции:
 - добавления;
 - удаления;
 - поиска;
 - просмотра;
 - остальные функции добавлять по необходимости.
5. Для данного контейнерного класса предусмотреть при формировании элемента задание режима уникальных элементов (т.е. проверку на дублирование значений элементов).
6. Для обработки всех ошибочных ситуаций использовать конструкцию `try...catch()`.
7. Дополнительно к контейнеру рекомендуется реализовать класс-итератор.
8. В функции `main` создать три экземпляра шаблонного класса-контейнера для разных типов данных. Работа с этими объектами должна демонстрироваться на следующих операциях: добавить – просмотреть – найти – удалить – найти – просмотреть.
9. Отладить и выполнить полученную программу. Проверить обработку исключительных ситуаций (например, чтение из пустого стека, дублирование объектов и т.п.).
10. Создать отчет, включающий задание, листинг программы и результаты ее выполнения.

Контрольные вопросы

1. Шаблонные классы.
2. Контейнерные структуры: стек, очередь.
3. Контейнерные структуры: список, кольцо.
4. Контейнерные структуры: бинарное дерево.
5. Обработка исключительных ситуаций.

Варианты заданий

1. Стек.
2. Очередь.
3. Список односвязный.
4. Список двусвязный.
5. Бинарное дерево.
6. Кольцо односвязное.
7. Кольцо двусвязное.
8. Матрица.
9. Граф.

Лабораторная работа №11

Тема работы: Пространства имен.

Цель работы: Создание консольной программы с реализацией объектов, которые находятся в собственном пространстве имен, реализация взаимодействия классов путем наследования, композиции и агрегации, рассмотрение особенностей построения диаграммы классов.

Общие требования к выполнению работы

1. Рассмотреть теоретические сведения по теме лабораторной работы.
2. Для полученного списка классов требуется сформировать диаграмму классов, используя принципы наследования, композиции и агрегации.
3. Создать проект консольной программы.
4. Создать собственное пространство имен, в которое будут помещены все создаваемые пользовательские классы.
5. Реализовать классы, согласно диаграмме классов, причем:
 - во всех классах должны быть свои данные в виде 1-2 переменных, представляющие характеристики объектов и расположенные в `private` или `protected` секциях;
 - во всех классах создать конструктор по умолчанию и конструктор инициализации (не забыть про передачу параметров в конструкции базовых классов);
 - во всех классах создать деструктор;
 - во всех классах создать метод просмотра содержимого;
 - остальные методы создавать по необходимости.
6. Описание каждого класса требуется реализовать в отдельной паре файлов (.h и .cpp).
7. В головной функции создать объект класса, расположенного последним в списке, выполнить доступ к его методам и методам его составных частей.
8. Отладить и выполнить полученную программу. Рассмотреть вызовы конструкторов и деструкторов при создании и разрушении объектов. Рассмотреть различные виды организации доступа к элементам собственного пространства имен и пространства имен `std`.
9. Отладить и выполнить полученную программу.
10. Создать отчет, включающий задание, диаграмму классов, листинг программы и результаты ее выполнения.

Контрольные вопросы

1. Объявление пространства имен.
2. Пространство имен `std`.
3. Обращение к элементам пространства имен.
4. Директива `using` и объявление `using`.
5. Диаграмма классов.
6. Наследование, композиция и агрегация.

Варианты заданий

1. Человек, отдел, специальность, рабочее_место, начальник, должность, работник.
2. Человек, страна, турфирма, путевка, адрес, тургид, турист.
3. Человек, родитель, адрес, улица, дом, классный_руководитель, ученик.
4. Человек, ВУЗ, специальность, курс, куратор, группа, студент.
5. Транспортное_средство, вагон, машинист, двигатель, проводник, локомотив, поезд.
6. Транспортное_средство, двигатель, моторное_масло, колесо, фара, лампа, автомобиль.
7. Печатное_издание, автор, издатель, переплет, страница, обложка, книга.
8. Печатное_издание, владелец, колонка, статья, типография, автор, газета.
9. ЭВМ, процессор, память, винчестер, программа, ОЗУ, ПЭВМ.

10. Электронный_прибор, ЖКИ-экран, блок_питания, конденсатор, транзистор, телевизор.
11. Архитектурный_объект, координата, квартира, этаж, архитектор, жилой_дом;.
12. Архитектурный_объект, координата, автор, материал, описание, памятник.
13. Растение, гнездо, ветка, корень, лист, плод, дерево.
14. Фигура, координата, цвет, линия, фон, заголовок, круг.

Лабораторная работа №12

Тема работы: Использование библиотеки STL для реализации обработки данных.

Цель работы: Создание консольной программы для представления данных с использованием контейнеров и обработки их с использованием алгоритмов и функторов библиотеки STL.

Общие требования к выполнению работы

1. Рассмотреть теоретические сведения по теме лабораторной работы.
2. Создать проект консольной программы.
3. Реализовать программу, которая обрабатывает строки текстового файла согласно заданию, а результаты обработки помещает в строки другого текстового файла. Для хранения и обработки данных использовать контейнеры и алгоритмы библиотеки STL.
4. На экран выводить отладочную информацию: какой файл открыт (или ошибка открытия файла), номер обработанной строки и результат ее обработки и т.п. Для обработки всех исключительных ситуаций использовать конструкцию `try...catch()`.
5. Отладить и выполнить полученную программу.
6. Создать отчет, включающий задание, листинг программы и результаты ее выполнения.

Контрольные вопросы

1. Контейнеры STL.
2. Алгоритмы STL.
3. Библиотека для работы со строками `<string>`.

Варианты заданий

1. Подсчет заданных символов в строке.
2. Подсчет слов в строке.
3. Подсчет слов в строке, которые содержат указанную подстроку.
4. Подсчет слов в строке, которые содержат все символы из указанного набора символов.
5. Подсчет слов в строке, которые содержат хотя бы один символ из указанного набора символов.
6. Подсчет слов в строке, не содержащих ни одного символа из указанного набора символов.
7. Сумма целых положительных чисел в строке (числа – слова, содержащие только цифры).
8. Сумма знаковых чисел в строке (числа – слова, содержащие цифры, а также необязательный ведущий знак числа («+» или «-») и необязательный разделитель дробной части («.» или «,»)).
9. Проверка правильности расстановки скобок («(» и «)») в строке с учетом их вложенности.
10. Проверка правильности расстановки скобок 4 видов: («(» и «)»), («[» и «]»), («{» и «}»), («<» и «>») в строке с учетом их вложенности и парности.
11. Обработка строк параметров вида `имя_параметра=значение_параметра`. Результатом должен быть список параметров в памяти.
12. Обработка файлов инициализации параметров.

Лабораторная работа №13

Тема работы: Динамическая идентификация данных.

Цель работы: Создание консольной программы для обработки разнотипных данных с использованием динамической идентификации и приведения типов.

Общие требования к выполнению работы

1. Рассмотреть теоретические сведения по теме лабораторной работы.
2. Создать проект консольной программы.
3. По полученному базовому классу создать классы наследников по трем разным ветвям наследования ($B \leftarrow P1$, $B \leftarrow P2$ и $B \leftarrow P3$):
 - во всех классах должны быть свои данные (характеристики объектов), расположенные в private или protected секциях;
 - во всех классах создать конструкторы инициализации объектов для всех классов (не забыть про передачу параметров в конструкции базовых классов);
 - во всех классах создать деструкторы;
 - остальные методы создавать по необходимости.
4. Создать в базовом классе виртуальную функцию вывода объекта на экран (всех его параметров). Выполнить реализацию этой виртуальной функции в классах наследниках.
5. Создать в каждом производном классе уникальную неvirtуальную функцию.
6. Задать в базовом классе деструктор как виртуальный.
7. В головной функции динамически создать массив указателей на базовый класс. Заполнить массив указателями на динамически создаваемые объекты базового и производных классов (B, P1, P2, P3). Для каждого элемента массива, который относится к производному классу, перед вызовом виртуальной функции выполнить его уникальную неvirtуальную функцию. Удалить из памяти динамически выделенные объекты.
8. Отладить и выполнить полученную программу.
9. Создать отчет, включающий задание, листинг программы и результаты ее выполнения.

Контрольные вопросы

1. Динамическая идентификация типа.
2. Оператор `dynamic_cast`.
3. Замена оператора `type_id` оператором `dynamic_cast`.

Варианты заданий

1. «Человек».
2. «Учащийся».
3. «Работник».
4. «Транспортное средство».
5. «Инструмент».
6. «Продукт».
7. «Дом».
8. «Книга».
9. «Телевизор».
10. «Компьютер».
11. «Фигура».
12. «Завод».
13. «Животное».
14. «Растение».

15. «Гриб».

16. «Вирус».