**Отчет по занятию №6.**

**Разработка программы обработки массива**

**Контрольные вопросы по теме: «Массивы»**

1. Что называется массивом? Как он объявляется? Какой параметр указывается в скобках при объявлении и использовании массива?

**Ответ:** массив - это структура данных, содержащая элементы одного типа, расположенные в памяти друг за другом. Массив объявляется с помощью указания типа элементов и их количества в квадратных скобках, например: int arr[5]. Параметр, указываемый в скобках, указывает на размер массива.

1. Что называется размером массива? Чем ограничен максимальный размер массива?

**Ответ:** размер массива - это количество элементов, которые могут быть хранены в массиве. Максимальный размер массива ограничен доступной памятью на компьютере.

1. Что называется индексом? Какие данные могут быть использованы в качестве индексов при обозначении элементов массива?

**Ответ:** индекс - это номер элемента в массиве. В качестве индексов могут использоваться целые числа, начиная с нуля и заканчивая размером массива минус один.

1. Сколько размерностей максимально может иметь многомерный массив?

**Ответ:** многомерный массив может иметь неограниченное количество размерностей, но обычно до трех размерностей используются в программах.

1. Какие массивы называются статическими, динамическими?

**Ответ:** статический массив имеет фиксированный размер, который указывается при объявлении и не может быть изменен в процессе выполнения программы. Динамический массив создается во время выполнения программы и его размер может быть изменен динамически.

1. Каковы особенности объявления и использования статических и динамических массивов? Как можно изменить размер динамического массива?

**Ответ:** статический массив объявляется с фиксированным размером при компиляции программы, в то время как динамический массив создается во время выполнения программы с помощью оператора new и его размер может быть изменен с помощью оператора realloc.

1. Может ли быть динамический массив многомерным и почему?

**Ответ:** динамический массив может быть многомерным, но каждая размерность должна быть выделена отдельно, поэтому это может быть неудобно.

1. Как массивы могут передаваться в методы?

**Ответ:** массивы могут быть переданы в методы как указатели на первый элемент массива или как ссылки на массив. При передаче массива в метод передается только указатель на его первый элемент, что позволяет работать с элементами массива в функции.

**Контрольные вопросы по теме: «Подпрограммы как методы»**

1. Что называется методом-процедурой и методом-функцией?

**Ответ:** метод-процедура - это метод, который не возвращает значение (void), а метод-функция - это метод, который возвращает значение определенного типа.

1. Что задается при определении метода?

**Ответ:** при определении метода задается его сигнатура, включающая имя метода, тип возвращаемого значения, список параметров и их типы.

1. Как организуется информационная связь между программой и методом?

**Ответ:** информационная связь между программой и методом организуется через вызов метода из программы.

1. Какие соответствия должны соблюдаться между формальными и фактическими параметрами метода?

**Ответ:** фактические параметры метода должны соответствовать по количеству, типу и порядку формальным параметрам метода.

1. Как выглядит оператор вызова метода-процедуры?

**Ответ:** оператор вызова метода-процедуры выглядит как имя метода, за которым следуют круглые скобки с передаваемыми параметрами, если они есть.

1. Какие разновидности (статусы) формальных параметров возможны при определении метода?

**Ответ:** формальные параметры могут быть обычными, константными, указателями или ссылками.

1. Каким образом могут передаваться параметры в метод? Какой способ передачи параметров в метод используется по умолчанию?

**Ответ:** параметры могут передаваться в методы по значению или по ссылке. По умолчанию используется передача параметров по значению.

1. Чем могут быть выражены фактические параметры, передаваемые в методы по значению и по ссылке?

**Ответ:** фактические параметры, передаваемые по значению, выражаются как константные выражения или переменные, а по ссылке - как ссылки на переменные.

1. Какую роль (входной и/или выходной параметр) могут выполнять параметры, передаваемые в методы по значению и по ссылке?

**Ответ:** параметры могут выполнять роль входных, выходных или входно-выходных параметров, в зависимости от их объявления в методе.

1. Как могут передаваться в методы массивы?

**Ответ:** массивы могут передаваться в методы по значению, по указателю или по ссылке.

1. Как можно задать произвольное количество параметров метода при его определении?

**Ответ:** произвольное количество параметров метода можно задать с помощью механизма переменного числа аргументов (variadic arguments).

**Контрольные вопросы по теме: «Указатели»**

1. Что называется указателем? Как он объявляется, инициализируется и используется? Для чего используются оператор получения адреса и оператор косвенного доступа?

**Ответ:** указатель - это переменная, которая содержит адрес другой переменной. Он объявляется с указанием типа данных, на который он указывает, например: int \*ptr; - указатель на целое число. Для инициализации указателя используется оператор присваивания, например: ptr = &x; - где x это переменная, адрес которой нужно сохранить в указателе. Оператор получения адреса (&) используется для получения адреса переменной, а оператор косвенного доступа (\*) используется для доступа к значению переменной, на которую указывает указатель.

1. Особенности указателей на тип char. Особенности массивов указателей. Их объявление, инициализация и использование.

**Ответ:** указатель на тип char используется для работы со строками. Массив указателей - это массив, элементами которого являются указатели. Они объявляются, как массив указателей на определенный тип данных, например: int \*ptrArray[10]; - массив из 10 указателей на целые числа. Инициализация происходит путем присваивания адресов переменных или выделенных участков памяти. Использование массива указателей позволяет работать с несколькими указателями одновременно.

1. В каких выражениях могут использоваться указатели? Операторы, используемые с указателями, примеры использования. Использование указателей для работы с массивами.

**Ответ:** указатели могут использоваться в выражениях для работы с адресами памяти. Операторы, используемые с указателями, включают в себя операторы присваивания, арифметические операторы (+, -, ++, --), оператор разыменования (\*), оператор взятия адреса (&) и другие. Пример использования указателей для работы с массивами: int arr[5]; int \*ptr = arr; - указатель ptr указывает на начало массива arr.

1. Что называется ссылкой? Как она объявляется, инициализируется и используется. Назначение и особенности использования оператора sizeof.

**Ответ:** ссылка - это псевдоним переменной. Она объявляется с использованием символа &, например: int x = 10; int &ref = x; - переменная ref является ссылкой на переменную x. Оператор sizeof возвращает размер переменной или типа данных в байтах. Он может использоваться для вычисления размера массива или структуры.

1. Особенности динамического выделения памяти. Какие операторы используются для выделения и возврата памяти? Форматы и примеры их использования.

**Ответ:** для динамического выделения памяти в C++ используются операторы new и delete. Оператор new выделяет участок памяти нужного размера и возвращает указатель на этот участок. Например: int \*ptr = new int; - выделение памяти под целое число. Оператор delete освобождает выделенную память: delete ptr; - освобождение памяти, выделенной для переменной ptr.

**Контрольные вопросы по теме: «Структуры»**

1. Что называется структурой? Как объявляется структура и переменная типа структуры? Как можно использовать структуры?

**Ответ:** структура в C++ - это пользовательский тип данных, который позволяет объединить несколько переменных разных типов под одним именем. Структура объявляется с использованием ключевого слова "struct", после которого идет название структуры и список переменных в фигурных скобках.

1. Как объявляются и используются битовые поля в структурах?

**Ответ:** битовые поля в структурах позволяют экономить память, задавая размер каждого поля в битах. Для объявления битовых полей в структуре используется двоеточие и число битов после названия переменной.

**Контрольные вопросы по теме: «Перечисления»**

1. Что называется перечислением? Как оно объявляется, инициализируется и используется?

**Ответ:** перечисление (enum) в C++ - это пользовательский тип данных, который представляет собой список именованных целочисленных констант.

Перечисление объявляется с помощью ключевого слова enum, например: ```cpp enum Color {RED, GREEN, BLUE}; ``` Инициализировать переменные типа перечисления можно следующим образом: ```cpp Color color = RED; ```

Использовать перечисление можно для определения переменных, аргументов функций, возвращаемых значений и т.д.

1. Какой тип могут иметь перечисления? На что он указывает?

**Ответ:** перечисления в C++ могут иметь целочисленный тип (int по умолчанию), но также могут иметь явно определенный тип, например: ```cpp enum class Status : char {OK, ERROR}; ``` В данном случае перечисление Status имеет тип char. Тип перечисления указывает на тип используемых значений.

1. Какие операции могут применяться к значениям перечислений?

**Ответ:** к значениям перечислений можно применять операции сравнения (==, !=) и присваивания. Также можно выполнять преобразование перечислений в целочисленные типы и наоборот.

1. Как создается перечисление с битовыми флагами?

**Ответ:** для создания перечисления с битовыми флагами (enum class с флагами) используется следующий синтаксис: ```cpp enum class Permissions : unsigned int { Read = 1, Write = 2, Execute = 4 }; ```

В данном случае каждая константа имеет значение, которое представляет собой степени двойки, что позволяет использовать их в качестве флагов и комбинировать их с помощью побитовых операторов.

**Контрольные вопросы по теме: «Организация работы с файлами»**

1. На какие типы делятся файлы по организации хранения информации в них и способам доступа к этой информации?

**Ответ:** файлы по организации хранения информации делятся на текстовые и бинарные, а по способам доступа - последовательного и произвольного.

1. Какие файлы являются файлами последовательного доступа, произвольного доступа, двоичными (бинарными)?

**Ответ:** файлы последовательного доступа обрабатываются по порядку, файлы произвольного доступа позволяют обращаться к данным в любом порядке, бинарные файлы содержат данные в двоичном формате.

1. Какие типовые операции используются при работе с файлами?

**Ответ:** типовые операции при работе с файлами включают открытие, чтение, запись, закрытие, перемещение указателя и проверку наличия файла.

1. Для чего предназначены управляющие элементы: OpenFileDialog, SaveFileDialog, FolderBrowserDialog? Каковы особенности их использования?

**Ответ:** управляющие элементы OpenFileDialog, SaveFileDialog и FolderBrowserDialog предназначены для выбора файлов/папок пользователем. OpenFileDialog используется для открытия файлов, SaveFileDialog - для сохранения, а FolderBrowserDialog - для выбора папки. Особенностью их использования является предоставление пользователю удобного интерфейса для работы с файлами и папками.

1. Методы каких классов (для работы с файловой системой) из пространства имен System.IO (File, FileInfo, Directory, DirectoryInfo, Path, DriveInfo, FileStream, StreamReader, StreamWriter) являются статическими? В чем особенность статических методов классов?

**Ответ:** статические методы классов из пространства имен System.IO не требуют создания экземпляра класса для их вызова. Они могут быть использованы напрямую через имя класса, что упрощает работу с файловой системой.

**Контрольные вопросы по теме: «Разработка пользовательского интерфейса»**

1. Что называется интерфейсом? Принципы разработки интерфейса, их характеристика.

**Ответ:** интерфейс в программировании - это способ взаимодействия пользователя с программой. Принципы разработки интерфейса включают в себя простоту использования, интуитивность, эргономичность, консистентность, адаптивность к различным устройствам. Характеристики интерфейса могут включать в себя элементы управления, структуру, цветовую схему, шрифты и т.д.

1. Типы интерфейсов, их характеристика и типовой состав элементов.

**Ответ:** типы интерфейсов включают в себя текстовые, графические, веб-интерфейсы и др. Типовой состав элементов интерфейса может включать кнопки, поля ввода, списки, меню и т.д.

1. Характеристика основных элементов пользовательского интерфейса, особенности их реализации.

**Ответ:** основные элементы пользовательского интерфейса включают в себя кнопки, текстовые поля, метки, списки, меню и др. Особенности их реализации включают в себя обработку событий, стилизацию, позиционирование на форме и т.д.

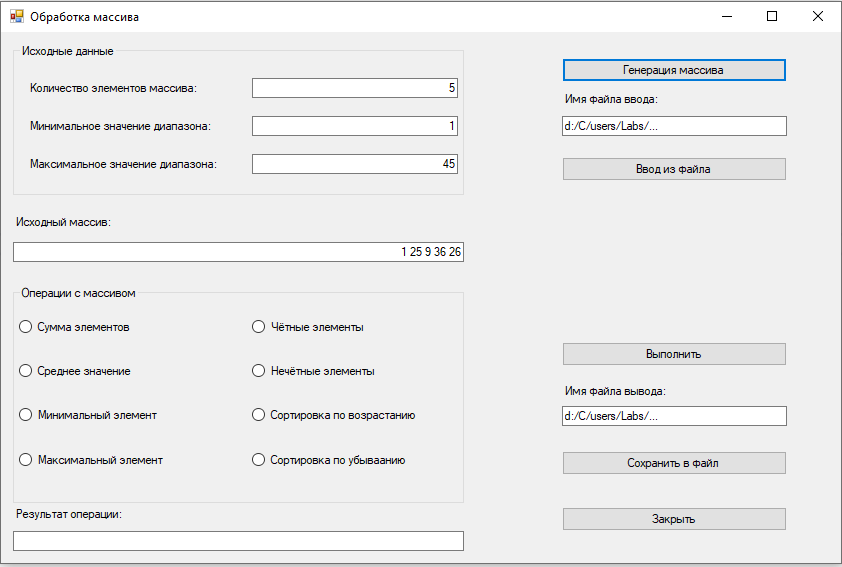
1. Диалоговые окна, особенности их реализации. Модальность диалоговых окон.

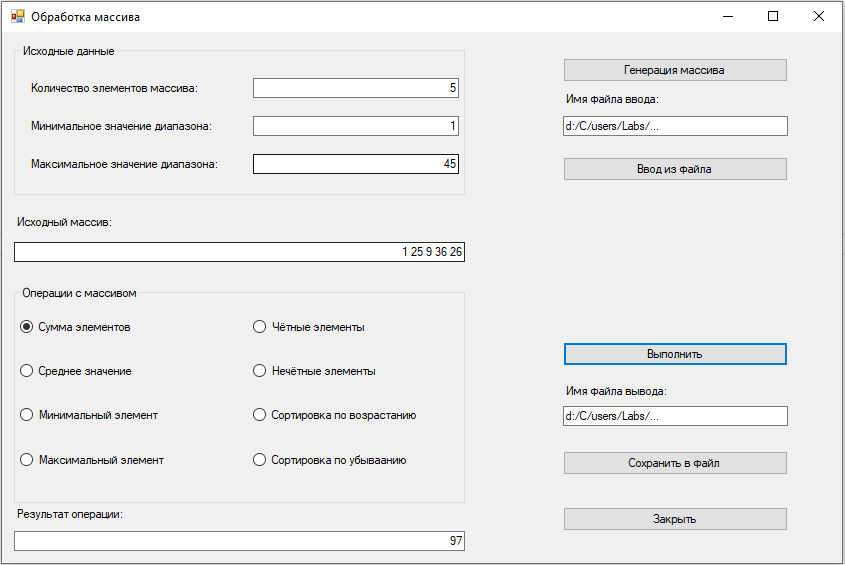
**Ответ:** диалоговые окна - это специальные окна, которые появляются поверх основного окна программы для выполнения определенных действий. Они могут быть модальными, что означает, что пользователь не может взаимодействовать с другими окнами программы, пока диалоговое окно не будет закрыто.

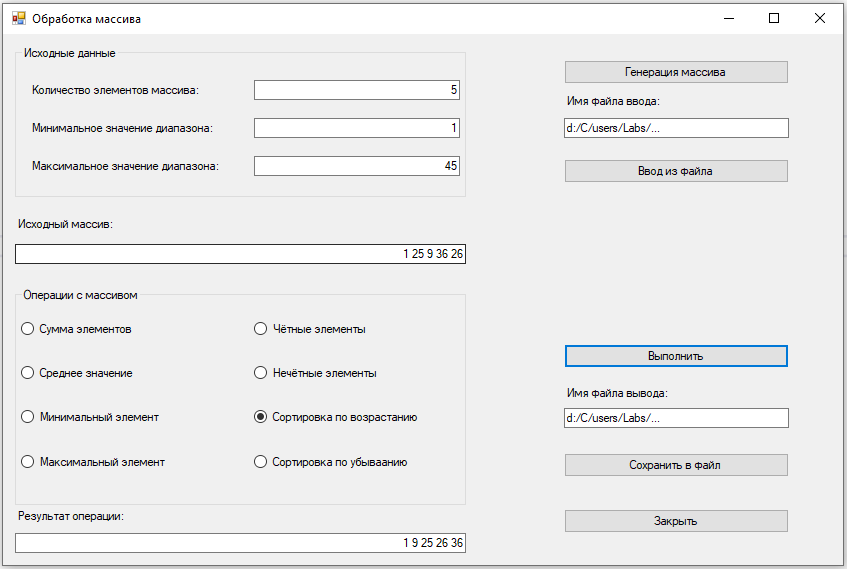
1. Назначение и характеристика стандартных элементов управления, особенности их применения.

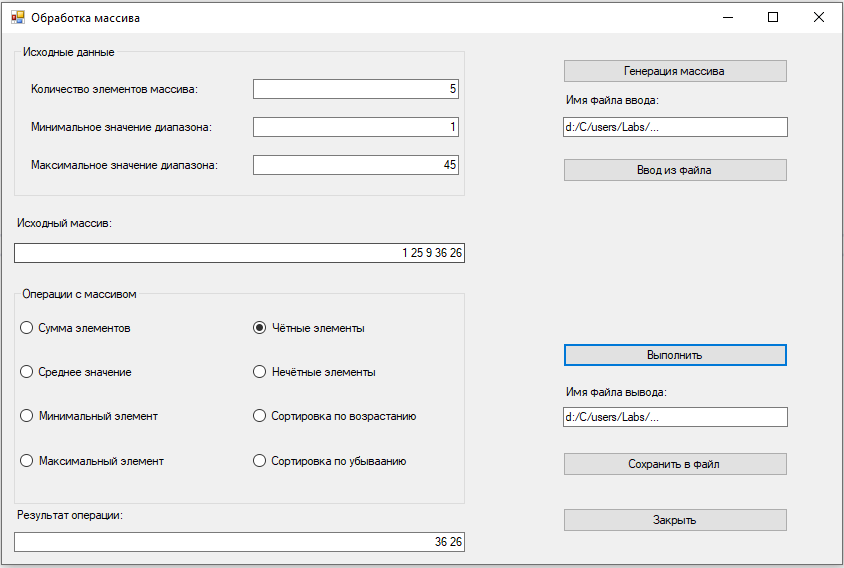
**Ответ:** стандартные элементы управления включают в себя кнопки, чекбоксы, радиокнопки, полосы прокрутки, поля ввода и др. Они используются для взаимодействия пользователя с программой и обеспечивают удобство использования.

**Результат работы программы:**

****

****

****

****