

**КУРСОВОЙ ПРОЕКТ**

**по МДК 01.01** Разработка программных модулей

**специальности** 09.02.07 Информационные системы и программирование

**на тему:** Реализация модуля для обработки вещественных одномерных массивов

Выполнил студент

группы 4ИС3010

Елистратов М.О

(Фамилия, инициалы)

Проверил преподаватель

Тарасов А.Д

(Фамилия, инициалы)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Подпись)

Москва, 2022

Содержание

Содержание 2

Введение 3

1 Теоретическая часть 5

1.1 Описание предметной области 5

1.2 Постановка задачи 8

1.3 Выбор технологии реализации 9

1.4 Этапы разработки программного продукта 11

2 Практическая часть 12

2.1 Описание сценариев использования 12

2.2 Проектирование модели данных 12

2.3 Проектирование интерфейса 12

2.4 Разработка программы 12

2.5 Тестирование программы 12

Заключение 13

Список использованных источников 14

Приложения 15

Введение

Актуальность данной темы обусловлена тем что массивы очень широко используются при разработке различного рода приложений. Массивы являются распространенным и полезным способом сохранения многих различных частей связанных данных. Массивы полезны при создании отсортированных и несортированных списков данных и для выполнения многих других задач. С понятием “Массив” приходится работать при решении научно- технических и научно-экономических задач связанных с обработкой совокупностей большого количества значений. До сих пор мы рассматривали переменные, которые имели только одно значение, которые могли содержать в себе только одну величину определенного типа. Исключением являлись лишь строковые переменные, которые представляют собой совокупность данных символьного типа, но и при этом мы говорили о строке, как об отдельной величине. Вы знаете, что компьютер предназначен в основном для облегчения работы человека с большими информационными объемами. Поэтому во всех существующих языках имеются типы переменных, отвечающие за хранение больших массивов данных. В языке Паскаль они так и называются: "массивы". Массивом будем называть упорядоченную последовательность данных одного типа, объединенных под одним именем. Кстати, под это определение подходит множество объектов из реального мира: словарь (последовательность слов), мультфильм (последовательность картинок) и т.д. Проще всего представить себе массив в виде таблицы, где каждая величина находится в собственной ячейке. Положение ячейки в таблице должно однозначно определяться набором координат (индексов). Самой простой является линейная таблица, в которой для точного указания на элемент данных достаточно знания только одного числа (индекса). Мы с вами пока будем заниматься только линейными массивами, так как более сложные структуры строятся на их основе. Единственным действием, которое возможно произвести с массивом целиком - присваивание. Однако, присваивать можно только массивы одинаковых типов. Никаких других операций с массивами целиком произвести невозможно, но с элементами массивов можно работать точно также, как с простыми переменными соответствующего типа. Обращение к отдельному элементу массива производится при помощи указания имени всего массива и в квадратных скобках - индекса конкретного элемента.

Объект для исследования: Массивы в программах.

Предмет исследования: Работа с массивами.

Цель: Рассмотреть основные алгоритмы обработки массивов максимально близко к практическому их применению.

Задачи:

* изучение типов массивов;
* изучение типовых алгоритмов обработки одномерных и двумерных массивов;
* группировка задачи по методам решения;
* обобщение и систематизация теоретического и практического материала по теме.

Методы исследования: Анализ языка программирования.

1. Теоретическая часть
   1. Описание предметной области

Массив — это структура, представляющая собой упорядоченную совокупность элементов одного типа объединенных одним именем. Массивы используются при обработке набора данных одного типа.

Для получения доступа к элементу массива используется индекс. Индекс определяет местоположения элемента в массиве является величиной целого типа.

Каждому массиву отводится место в памяти последовательно расположенных друг за другом ячеек в каждую из которых записывается значение соответствующего элемента.

Преимуществом использования массивов состоит в том то массив является удобным способом хранения нескольких связанных элементов данных в едином контейнере для большего удобства и эффективности программирования.

Массив позволяет сохранять и манипулировать многими элементами данных посредством единственной переменной.

Кроме уменьшения общего числа различных имен переменных, которые необходимо отслеживать другим основным преимуществом использования массивов является то что можно использовать циклы для легкой обработки различных элементов массивов.

Цикл — это разновидность управляющей конструкции в высокоуровневых языках программирования для организации многократного исполнения набора инструкций. Также циклом может называться любая многократно исполняемая последовательность инструкций, организованная любым способом (например, с помощью условного перехода).

Объединяя массивы и циклы можно написать небольшое число операторов, которые обрабатывают большой объем данных. Выполнение тех же задач с использованием отдельных переменных может потребовать написания сотен операторов и переменных.

Массив — это множество однотипных элементов, объединенных общим именем и занимающих в компьютере определенную область памяти. Количество элементов в массиве всегда конечно. В общем случае массив – это структурированный тип данных состоящий из фиксированного числа элементов, имеющих один и тот же тип.

Другими словами, можно сказать что массив — это фиксированное количество упорядоченных однотипных компонент снабженный индексами, то есть является совокупностью конечного числа данных одного типа. В качестве элементов массива можно использовать любой тип данных поэтому правомерно существование массивов записей массивов указателей массивов строк массивов и так далее.

Массивы могут быть:

* Одномерными (одна строка и несколько столбцов);
* Двумерными (несколько строк и несколько столбцов);
* Многомерными (кубические таблицы).
  + 1. Одномерные массивы

Каждому используемому в программе конкретному массиву должно быть дано свое имя. Каждый компонент массива может быть явно обозначен путем указания имени массива, в классических языках программирования для указания конкретного элемента массива используется индекс, указывающий на определенную ячейку памяти.

Одномерный массив – это последовательность связанных одной переменной элементов, в которой все элементы имеют свой уникальный индекс (номер элемента в массиве). По данному индексу возможно обращаться к определенному значению.



Рисунок 1 - пример одномерного массива с указанием легенды способа работы массива.

* + 1. Многомерные массивы

Существуют так же многомерные массивы. Их основное отличие от одномерного массива заключается в том, что одномерные массивы по своей структуре являются последовательностью, а двумерные состоят из нескольких подобных последовательностей и более схожи с таблицей по своей структуре.

Многомерный массив состоит из нескольких измерений. По сути для указания адреса конкретной ячейки массива сначала выбирается через два адреса – первичный и вторичный. Первичный – это номер в последовательности. Вторичный – это номер самой последовательности.

Такие массивы могут использоваться для хранения информации в табличном виде, например, номер последовательности может быть указателем для программиста в виде набора значений, относящихся к какому-либо определенному методу или объекту.

Данные массивы чаще всего используются при работе с объектно-ориентированным программированием.



Рисунок X – пример строения многомерного (двумерного) массива.

* + 1. Трехмерные массивы

Еще один вид массива – трехмерные массив. Трехмерный массив – это некий набор двумерных массивов, используется в основном только для работы с трехмерной графиков и состоит из трех указателей координат. Первичная координата как правило означает номер последовательности, вторичная – номер элемента в последовательности, а третий указатель – это номер стороны “многогранника”, на котором расположен двумерный массив. Пример трехмерного массива и его визуальные отличия указаны на рисунке X.

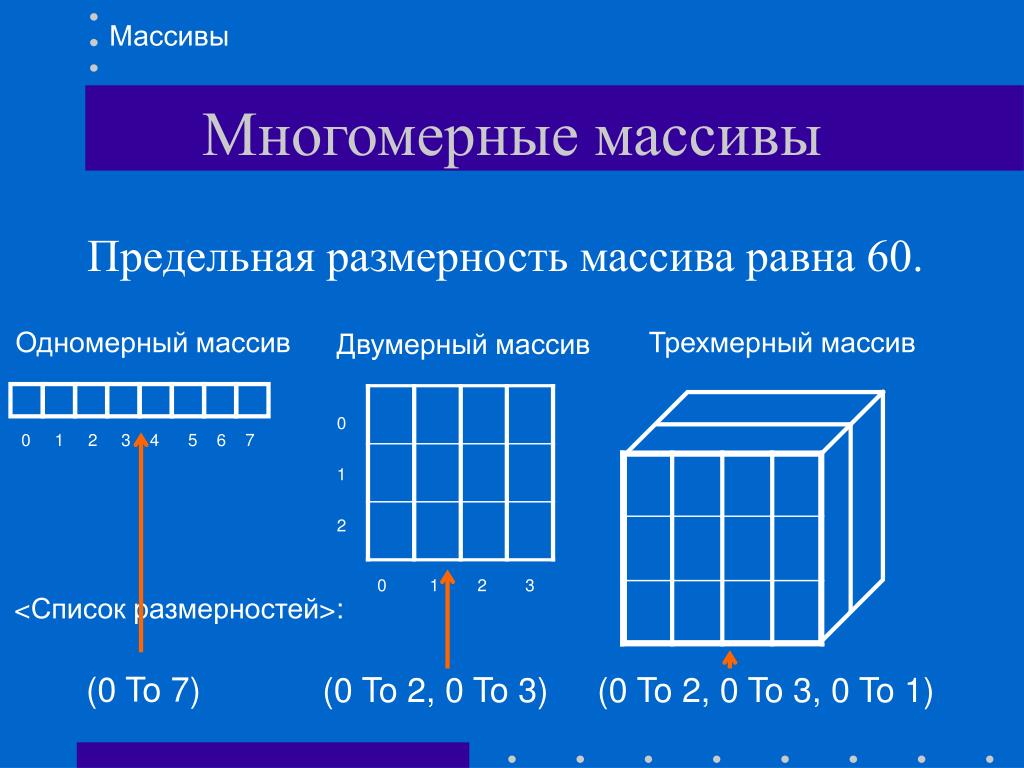


Рисунок X – отличие многомерного массива от одномерного и двумерного

* + 1. Зубчатые массивы

Помимо этих массивов существует зубчатый массив (ступенчатый массив). Это двумерный массив, длина последовательностей которого отличается друг от друга. Пример зубчатого массива и его визуальные отличия от двумерного указаны на рисунке X.

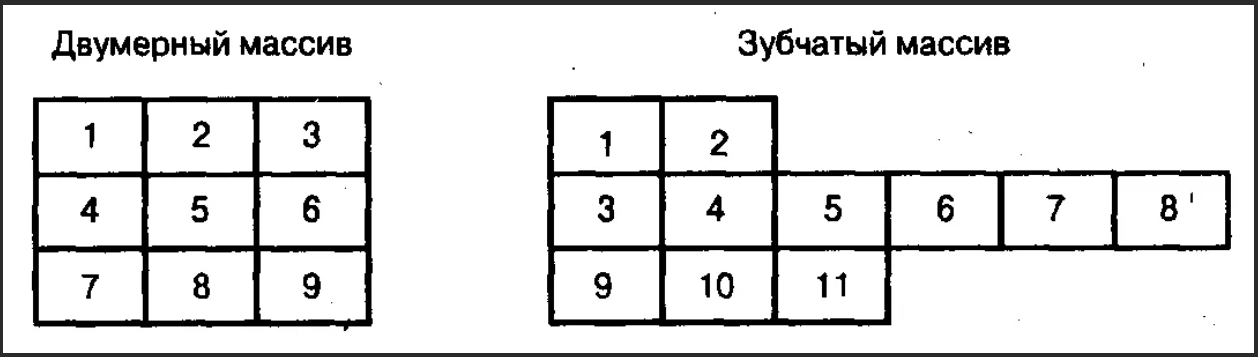


Рисунок X – отличия зубчатого массива от двумерного массива

* 1. Постановка задачи

Для того чтобы программисту было легче работать с массивами необходимо разработать программный модуль для обработки массивов.

Методы обработки массивов:

* сортировка массива (от меньшего к большему);
* удаление нечетных элементов массива;
* удаление четных элементов массива;
* удаление элемента массива, если заданное число меньше установленного;
* удаление элемента массива, если заданное число больше установленного;
* инверсия массива;
* сортировка массива (от большего к меньшему);
* Удаление элемента массива по установленному условию метода.
  1. Выбор технологии реализации

Для реализации поставленной задачи необходимо выбрать язык программирования, который соответствует следующим требованиям:

* расширяемость – это возможность увеличения функционала программного обеспечения без особых усилий;
* актуальность – язык должен быть поддерживаемым для того, чтобы разрабатываемое программное обеспечение имело актуальность;
* быстродействие – так как разработка связана со скоростью обработки элементов, необходимо, чтобы каждый метод обработки возвращал результат как можно быстрее;

Под данные условия подходит несколько языков. Языки, между которыми будет выбор – это С++ и C#.

C# (англ. C-sharp) - это язык программирования, основанный на объектах. Любой элемент данного языка является объектом, к которому привязаны какие-либо свойства и методы. Он включает в себя множество готовых к использованию библиотек, что является одним из его преимуществ. Этот язык наиболее простой, подходящий для новичков язык программирования. Пример реализации сортировки одномерного массива методом пузырька на языке С# указан на рисунке X.

…

С++ (англ. C-plus-plus) – это язык программирования, основанный на функциональном подходе. Переменные в данном языке наиболее простые и не имеют множества встроенных методов. Сам язык по классификации является наиболее низкоуровневым, менее затратным по ресурсам и имеет неограниченный доступ к памяти программы. Пример реализации сортировки одномерного массива мет

….

* 1. Этапы разработки программного продукта

1. Практическая часть
   1. Описание сценариев использования
   2. Проектирование модели данных
   3. Проектирование интерфейса
   4. Разработка программы
   5. Тестирование программы

Заключение

Список использованных источников

Приложения