

Министерство науки и высшего образования Российской Федерации
Казанский национальный исследовательский технический университет –
КАИ им. А.Н. Туполева

Институт компьютерных технологий и защиты информации
Отделение СПО ИКТЗИ «Колледж информационных технологий»

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЕ

Методические указания к лабораторным работам

Казань 2020

Составитель преподаватель СПО ИКТЗИ Мингалиев Заид Зульфатович

Методические указания к лабораторным работам по дисциплине «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЕ» предназначены для студентов направления подготовки 09.02.07 «Информационные системы и программирование»

КНИТУ-КАИ, 2020

ОГЛАВЛЕНИЕ

| | |
|---|---|
| ПРОЦЕСС СДАЧИ ВЫПОЛНЕННОЙ РАБОТЫ | 4 |
| ЛАБОРАТОРНАЯ РАБОТА №5. | 5 |
| Программирование алгоритмов с использованием динамического выделения памяти..... | 5 |

ПРОЦЕСС СДАЧИ ВЫПОЛНЕННОЙ РАБОТЫ

По итогам выполнения работы студент:

1. демонстрирует преподавателю правильно работающие программы;
2. демонстрирует приобретённые знания и навыки отвечает на пару небольших вопросов преподавателя по составленной программе, возможностям её доработки;
3. демонстрирует отчет по выполненной лабораторной работе.

Итоговая оценка складывается из оценок по трем указанным составляющим.

Шаблон оформления отчета представлен в приложении 1. Требования к формированию отчета представлены в приложении 2.

ЛАБОРАТОРНАЯ РАБОТА №5.

Программирование алгоритмов с использованием динамического выделения памяти.

ЦЕЛЬ РАБОТЫ

Приобрести умения и практические навыки для программирования алгоритмов с использованием механизма динамического выделения памяти.

ХОД РАБОТЫ

1) Динамическое выделение памяти

Поголовно у каждой переменной имеется свой индивидуальный адрес. **Адрес переменной** — это путь, по которому находится значение самой переменной. Он записывается в шестнадцатеричном виде. Переменные, которые вы создаёте в программе, по её завершению автоматически удаляются, чтобы не нагружать операционную память вашего компьютера.

В играх присутствует хорошая графика, различные спецэффекты. Например, тот же дым. Все это — переменная (может не одна!), которой в будущем придётся уничтожиться навсегда. А вот, если бы она не удалась, то она бы своей фоновой работой понемножку нагружала бы наш компьютер.

Поэтому в C/C++ присутствует возможность обратиться к переменной, и, если требует ситуация, удалить и создать её вовсе в другом участке программы, когда это, конечно, нам будет нужно.

Указатели — это переменные, в которых хранится адрес других переменных.

Чтобы пользоваться указателями, вам нужно использовать два оператора:

* — показывает значение переменной по заданному адресу (показывает, кто живет в этом номере). Если вы используете оператор *, то вы занимаетесь операцией разыменования указателя.

& — показывает адрес переменной (говорит, по какому адресу проживает этот человек).

Для создания указателей используется следующая конструкция:

| | |
|---|---|
| 1 | <code>*<имя переменной> = &<имя другой переменной></code> |
|---|---|

В самом начале мы ставим оператор `*` (звездочку). Так мы говорим компилятору, что хотим использовать тип данных — указатель. Далее мы должны указать имя нашей переменной. После знака равно нам нужно передать указателю адрес какой-то переменной, что мы и делаем с помощью оператора `&` (амперсанд). Чтобы передать адрес какой-то переменной, от одного указателя другому, нужно опускать оператор `*` для одного указателя (от которого мы передаем второму)

Чтобы получше понять указатели, давайте их разберем на примере ниже.

Отец работает программистом в крупной компании, которая находится в 0x145 городе, и ему предложили поехать в командировку в 0x195 город, которую он так долго ждал. Он смог оповестить только своего сына о том, что уезжает. Поэтому сыну придется передать это маме самому.

Пример выше мы сейчас реализуем на C++ с помощью указателей.

| | |
|---|---|
| 1 | <code>#include <iostream></code> |
| 2 | <code>using namespace std;</code> |
| 3 | <code>int main() {</code> |
| 4 | <code> int dad_gorod;</code> |
| 5 | <code> int *dad_son = &dad_gorod;</code> |
| 6 | <code> int *mama = dad_son;</code> |
| 7 | <code> return 0;</code> |
| 8 | <code>}</code> |

Давайте подробно разберем код выше:

- В строке 8: создали переменную `dad_gorod`, в которой находится адрес (город, в который уехал отец).
- В строке 9: создали указатель `dad_son` (это сын), он узнает имя города.
- В строке 11: объявили указатель `mama`, которая уже от сына получает имя города.

Передавая функции аргумент и изменяя его в функции, переменная, значение которой мы передавали, никак не изменится. И это понятно, потому что мы передаем значение одной переменной — другой переменной.

Чтобы изменить значение переменной, нам понадобится:

- В аргументах функции создать указатель.
- Далее при вызове функции передать адрес переменной, которую мы собираемся изменить.

Динамические переменные — это переменные, которые созданы напрямую с помощью указателей.

Чтобы мы могли полноценно создавать динамические переменные, нам понадобится изучить конструктор — `new`, после его использования в оперативной памяти компьютера выделяются ячейки на тот тип данных, который мы указали.

На каждый тип данных выделяется разное количество ячеек.

Для создания динамических переменных нам понадобится применять конструкцию ниже:

| | |
|---|--|
| 1 | <code><тип данных указателя> *<имя указателя> = new <тип данных>(<первоначальное значение>)</code> |
|---|--|

Давайте подробно ее разберем:

- `<тип данных указателя>` — указанный тип данных почти ни на что не повлияет.
- `new` — это конструктор, который и будет заключительным звеном для создания нашей переменной.
- `<тип данных>` — здесь нам понадобится указать тип, какой будет храниться в переменной. Он необязательно должен совпадать с типом указателя.
- `<первоначальное значение>` — с помощью круглых скобок можно указать значение переменной еще при ее инициализации. Использование круглых скобок в этой конструкции необязательно.

Если тип переменной отличается от типа указателя — то эта динамическая переменная будет весить больше в оперативной памяти, чем такая же переменная с одинаковыми типами!

Важно помнить, что динамические переменные — это указатели, и поэтому перед ними обязательно должен стоять оператор *.

У нас есть возможность освобождать память переменной или, если понятным языком, удалять переменную из оперативной памяти ПК. Переменная и так удалится из оперативной памяти компьютера при завершении программы, но если нам захотелось удалить ее еще в середине программы, то это будет возможно благодаря оператору delete.

Чтобы его использовать, нужно применить конструкцию ниже:

| | |
|---|---|
| 1 | <code>delete <имя переменной>;</code> |
|---|---|

Использование динамических переменных имеет маленький плюс. Он заключается в освобождении памяти переменной до завершения программы. Благодаря этому мы можем сначала удалить переменную, а потом ее снова создать в другом участке программы (когда это нам будет нужно).

Динамический массив — это массив, у которого количество ячеек можно задавать и переменной, и числовой константой. Это большой плюс перед использованием статического массива.

Для работы динамических массивов нам понадобится при инициализации указатель (всего лишь при инициализации!) и уже знакомый конструктор new.

Чтобы создать динамический массив мы будем использовать конструкцию ниже:

| | |
|---|---|
| 1 | <code><тип данных> *<имя массива> = new <тип переменных> [<количество ячеек>];</code> |
|---|---|

Давайте подробно ее разберем:

- `<тип данных>` — без разницы какой тип данных тут будет находиться, но лучше тот, который будет совпадать с типом переменных.

- <тип переменных> — указанный сюда тип и будут иметь ячейки массива.

- <количество ячеек> — здесь мы задаем размер массива (например [n] или [25]).

Динамический массив полностью идентичен обычному массиву, кроме:

- Своей инициализации
- Возможностью своевременно освободить память.

Для создания двумерного динамического массива мы будем использовать похожую конструкцию (как и в одномерном динамическом массиве):

| | |
|---|---|
| 1 | <тип данных> **<имя массива> = new <тип данных массива>*[<количество ячеек>]; |
|---|---|

Дальше для каждой ячейки мы должны создать одномерный массив.

Чтобы это сделать, нам понадобится цикл for и конструктор new.

| | |
|---|--|
| 1 | for (int i = 0; i < n; i++) { |
| 2 | <имя массива>[i] = new <тип ячеек> [<количество ячеек>]; |
| 3 | } |

В <количество ячеек> можно задавать разные значения. Поэтому сначала для первого массива можно задать длину 1 (new int [1]), потом для второго — длину 2 (new int [2]).

Параметры функции в C++ могут представлять указатели. Указатели передаются в функцию по значению, то есть функция получает копию указателя. В то же время копия указателя будет в качестве значения иметь тот же адрес, что оригинальный указатель. Поэтому используя в качестве параметров указатели, мы можем получить доступ к значению аргумента и изменить его.

2) Задание на лабораторную работу

Все части лабораторной работы выполняются на языке программирования C++.

При выполнении первого задания необходимо создать только одну функцию и, при помощи указателей, обеспечить возврат более одного полученного значения. При выполнении третьего задания необходимо использовать динамическую реализацию массивов.

| Номер | Текст задания |
|------------------|--|
| Вариант 1 | |
| 1. | Написать программу, вычисляющую объем и площадь поверхности цилиндра по известному радиусу основания и высоте. Расчет должен производиться в функции, которая получает данные по указателю и возвращает результат по указателю. |
| 2. | Написать программу, определяющую максимальный элемент одномерного массива. При вводе/выводе элементов использовать индексы, а обработку элементов массива осуществлять с помощью указателей. Расчет должен производиться в функции, в которую массив передается по указателю |
| 3. | Написать программу, выполняющую транспонирование неквадратной матрицы. Матрицы должны храниться в памяти в виде двумерного динамического массива, размерности исходной матрицы вводятся пользователем с клавиатуры. Предусмотреть генерацию значений матрицы как случайных целых чисел в диапазоне от -10 до 10. |
| Вариант 2 | |
| 1. | Написать программу для определения максимального из трех положительных чисел, введенных пользователем, а также их произведения. Расчет должен производиться в функции, которая получает данные по указателю и возвращает результат по указателю. |
| 2. | Написать программу, определяющую модуль разности между количеством отрицательных и положительных элементов |

| | |
|------------------|---|
| | одномерного массива. При вводе/выводе элементов использовать индексы, а обработку элементов массива осуществлять с помощью указателей. Расчет должен производиться в функции, в которую массив передается по указателю |
| 3. | Написать программу, выполняющую расчет произведения двух неквадратных матриц. Матрицы должны храниться в памяти в виде двумерного динамического массива, размерности матриц вводятся пользователем с клавиатуры. Предусмотреть генерацию значений матриц как случайных вещественных чисел в диапазоне от -0.5 до 0.5. |
| Вариант 3 | |
| 1. | Написать программу, вычисляющую сопротивление электрической цепи, состоящей из двух последовательно соединенных резисторов, а также падения напряжения на каждом из них (сила тока известна). Расчет должен производиться в функции, которая получает данные по указателю и возвращает результат по указателю. |
| 2. | Написать программу, определяющую минимальный элемент в одномерном массиве и увеличивающую его в два раза. При вводе/выводе элементов использовать индексы, а обработку элементов массива осуществлять с помощью указателей. Расчет должен производиться в функции, в которую массив передается по указателю |
| 3. | Написать программу, выполняющую поворот неквадратной матрицы на 90° по часовой стрелке. Матрицы должны храниться в памяти в виде двумерного динамического массива, размерности исходной матрицы вводятся пользователем с клавиатуры. Предусмотреть генерацию значений матрицы как случайных целых чисел в диапазоне от -20 до 20. |
| Вариант 4 | |

| | |
|------------------|---|
| 1. | Написать программу, вычисляющую значения a^b и b^a . Расчет должен производиться в функции, которая получает данные по указателю и возвращает результат по указателю. |
| 2. | Написать программу, заменяющую положительные элементы массива на их квадраты. При вводе/выводе элементов использовать индексы, а обработку элементов массива осуществлять с помощью указателей. Расчет должен производиться в функции, в которую массив передается по указателю |
| 3. | Написать программу, выполняющую расчет суммы двух неквадратных матриц. Матрицы должны храниться в памяти в виде двумерного динамического массива, размерности матриц вводятся пользователем с клавиатуры. Предусмотреть генерацию значений матриц как случайных вещественных чисел в диапазоне от -1.0 до 1.0. |
| Вариант 5 | |
| 1. | Написать программу, вычисляющую доход по вкладу и сумму к выдаче. Исходными данными для функции являются: сумма вклада, процентная ставка (годовых) и срок вклада (количество дней). Расчет должен производиться в функции, которая получает данные по указателю и возвращает результат по указателю. |
| 2. | Написать программу, заменяющую все элементы одномерного массива, кроме максимального, на их отрицательные значения. При вводе/выводе элементов использовать индексы, а обработку элементов массива осуществлять с помощью указателей. Расчет должен производиться в функции, в которую массив передается по указателю |
| 3. | Написать программу, выполняющую поворот неквадратной матрицы на 90° против часовой стрелки. Матрицы должны храниться в памяти в виде двумерного динамического массива, |

| | |
|------------------|---|
| | размерности исходной матрицы вводятся пользователем с клавиатуры. Предусмотреть генерацию значений матрицы как случайных целых чисел в диапазоне от 0 до 100. |
| Вариант 6 | |
| 1. | Написать программу, вычисляющую периметр и площадь треугольника по известным сторонам. Расчет должен производиться в функции, которая получает данные по указателю и возвращает результат по указателю. |
| 2. | Написать программу, заменяющую в одномерном массиве максимальные элементы на минимальные, а минимальные на максимальные. При вводе/выводе элементов использовать индексы, а обработку элементов массива осуществлять с помощью указателей. Расчет должен производиться в функции, в которую массив передается по указателю |
| 3. | Написать программу, выполняющую отражение неквадратной матрицы относительно центральной горизонтальной оси. Матрицы должны храниться в памяти в виде двумерного динамического массива, размерности матриц вводятся пользователем с клавиатуры. Предусмотреть генерацию значений матриц как случайных вещественных чисел в диапазоне от -1.0 до 1.0. |
| Вариант 7 | |
| 1. | Написать программу решающую квадратное уравнение. Расчет должен производиться в функции, которая получает данные по указателю и возвращает результат по указателю. |
| 2. | Написать программу, определяющую разность между суммой модулей отрицательных элементов и суммой положительных элементов одномерного массива. При вводе/выводе элементов использовать индексы, а обработку элементов массива |

| | |
|------------------|---|
| | осуществлять с помощью указателей. Расчет должен производиться в функции, в которую массив передается по указателю |
| 3. | Написать программу, выполняющую поворот неквадратной матрицы на 90° по часовой стрелке. Матрицы должны храниться в памяти в виде двумерного динамического массива, размерности исходной матрицы вводятся пользователем с клавиатуры. Предусмотреть генерацию значений матрицы как случайных вещественных чисел в диапазоне от 0.0 до 10.0. |
| Вариант 8 | |
| 1. | Написать программу, вычисляющую определитель и произведение диагональных элементов матрицы второго порядка. Расчет должен производиться в функции, которая получает данные по указателю и возвращает результат по указателю. |
| 2. | Написать программу, определяющую максимальный(-ые) элемент одномерного массива, и выводящую его номер. При вводе/выводе элементов использовать индексы, а обработку элементов массива осуществлять с помощью указателей. Расчет должен производиться в функции, в которую массив передается по указателю |
| 3. | Написать программу, выполняющую транспонирование неквадратной матрицы. Матрицы должны храниться в памяти в виде двумерного динамического массива, размерности матриц вводятся пользователем с клавиатуры. Предусмотреть генерацию значений матриц как случайных вещественных чисел в диапазоне от -2.0 до 2.0. |
| Вариант 9 | |
| 1. | Написать программу, вычисляющую разность квадратов и разность кубов двух чисел. Расчет должен производиться в функции, которая получает данные по указателю и возвращает результат по указателю. |

| | |
|-------------------|--|
| 2. | Написать программу, определяющую модуль максимального отклонения элементов массива от среднего значения всех элементов одномерного массива. При вводе/выводе элементов использовать индексы, а обработку элементов массива осуществлять с помощью указателей. Расчет должен производиться в функции, в которую массив передается по указателю |
| 3. | Написать программу, выполняющую отражение неквадратной матрицы относительно центральной вертикальной оси. Матрицы должны храниться в памяти в виде двумерного динамического массива, размерности исходной матрицы вводятся пользователем с клавиатуры. Предусмотреть генерацию значений матрицы как случайных вещественных чисел в диапазоне от -10 до 10. |
| Вариант 10 | |
| 1. | Написать программу, вычисляющую площадь и периметр треугольника по двум сторонам и углу между ними. Расчет должен производиться в функции, которая получает данные по указателю и возвращает результат по указателю. |
| 2. | Написать программу, определяющую минимальный элемент в одномерном массиве и выводящую его адрес. При вводе/выводе элементов использовать индексы, а обработку элементов массива осуществлять с помощью указателей. Расчет должен производиться в функции, в которую массив передается по указателю. |
| 3. | Написать программу, выполняющую поворот неквадратной матрицы на 90° против часовой стрелки. Матрицы должны храниться в памяти в виде двумерного динамического массива, размерности исходной матрицы вводятся пользователем с клавиатуры. Предусмотреть генерацию значений матрицы как случайных целых чисел в диапазоне от -50 до 50. |
| Вариант 11 | |

| | |
|-------------------|---|
| 1. | Написать программу, вычисляющую квадрат разности и квадрат суммы двух чисел. Расчет должен производиться в функции, которая получает данные по указателю и возвращает результат по указателю. |
| 2. | Написать программу, меняющую в одномерном массиве местами первый элемент с минимальным, а последний с максимальным. При вводе/выводе элементов использовать индексы, а обработку элементов массива осуществлять с помощью указателей. Расчет должен производиться в функции, в которую массив передается по указателю |
| 3. | Написать программу, выполняющую расчет суммы двух неквадратных матриц. Матрицы должны храниться в памяти в виде двумерного динамического массива, размерности матриц вводятся пользователем с клавиатуры. Предусмотреть генерацию значений матриц как случайных вещественных чисел в диапазоне от -2.5 до 2.5. |
| Вариант 12 | |
| 1. | Написать программу, вычисляющую площадь и периметр прямоугольника по известным координатам его вершин. Расчет должен производиться в функции, которая получает данные по указателю и возвращает результат по указателю. |
| 2. | Написать программу, меняющую местами два наибольших элемента одномерного массива с первым и последним. При вводе/выводе элементов использовать индексы, а обработку элементов массива осуществлять с помощью указателей. Расчет должен производиться в функции, в которую массив передается по указателю |
| 3. | Написать программу, выполняющую расчет произведения двух неквадратных матриц. Матрицы должны храниться в памяти в виде |

| | |
|-------------------|--|
| | двумерного динамического массива, размерности матриц вводятся пользователем с клавиатуры. Предусмотреть генерацию значений матриц как случайных вещественных чисел в диапазоне от -0.5 до 0.5. |
| Вариант 13 | |
| 1. | Написать программу, вычисляющую объем и площадь поверхности куба по заданным координатам двух смежных вершин (то есть обе точки являются концами одной из сторон). Расчет должен производиться в функции, которая получает данные по указателю и возвращает результат по указателю. |
| 2. | Написать программу, определяющую количество отрицательных элементов в массиве и выводящую их адреса. При вводе/выводе элементов использовать индексы, а обработку элементов массива осуществлять с помощью указателей. Расчет должен производиться в функции, в которую массив передается по указателю |
| 3. | Написать программу, выполняющую расчет произведения двух неквадратных матриц. Матрицы должны храниться в памяти в виде двумерного динамического массива, размерности матриц вводятся пользователем с клавиатуры. Предусмотреть генерацию значений матриц как случайных целых чисел в диапазоне от -25 до 25. |
| Вариант 14 | |
| 1. | Написать программу, вычисляющую объем и площадь поверхности треугольной пирамиды, если известны координаты ее вершин. Расчет должен производиться в функции, которая получает данные по указателю и возвращает результат по указателю. |
| 2. | Написать программу, определяющую среднее значение отрицательных и положительных элементов одномерного массива. При вводе/выводе элементов использовать индексы, а обработку элементов массива осуществлять с помощью указателей. Расчет |

| | |
|-------------------|--|
| | должен производиться в функции, в которую массив передается по указателю |
| 3. | Написать программу, выполняющую отражение неквадратной матрицы относительно центральной горизонтальной оси. Матрицы должны храниться в памяти в виде двумерного динамического массива, размерности матриц вводятся пользователем с клавиатуры. Предусмотреть генерацию значений матриц как случайных целых чисел в диапазоне от 0 до 100. |
| Вариант 15 | |
| 1. | Написать программу, вычисляющую среднее арифметическое и среднее геометрическое трех чисел. Расчет должен производиться в функции, которая получает данные по указателю и возвращает результат по указателю. |
| 2. | Написать программу, определяющую максимальное квадратичное отклонение элементов одномерного массива от среднего значения. При вводе/выводе элементов использовать индексы, а обработку элементов массива осуществлять с помощью указателей. Расчет должен производиться в функции, в которую массив передается по указателю |
| 3. | Написать программу, выполняющую поворот неквадратной матрицы на 90° против часовой стрелки. Матрицы должны храниться в памяти в виде двумерного динамического массива, размерности исходной матрицы вводятся пользователем с клавиатуры. Предусмотреть генерацию значений матрицы как случайных целых чисел в диапазоне от 0 до 50. |

3) Контрольные вопросы

1. Что такое указатель?

2. Чем отличается статическое выделение памяти от динамического выделения памяти?
3. С помощью каких операций можно манипулировать над указателями?
4. Может ли функция возвращать значение типа указатель? Если да, то как объявляется прототип такой функции?
5. Могут ли параметрами функции быть указатели на функции? Если да, то как происходит передача фактических параметров при вызове функции?
6. Для чего применяются двойные указатели?