Государственное бюджетное профессиональное образование учреждения Калужской области

"Калужский техникум электронных приборов"

” Различные способы ввода-вывода данных. Составление программ линейной структуры”

Вариант №10

Выполнил:

Студент группы ИСиП(п) 3/3

Толкачев Никита

Преподаватель:

Копьева М. Н.

1. Даны числа С,Р и D. Вычислить сумму корней квадратных этих чисел и произведение этих чисел..

*Код программы:*

#include <iostream>

#include <cmath>

using namespace std;

int main(){

    // Ввод данных

    int C,P,D;

    cout<<"\n Введите число C = ";

    cin>>C;

    cout<<"\n Введите число P = ";

    cin>>P;

    cout<<"\n Введите число D = ";

    cin>>D;

    int result1 = sqrt(C) + sqrt(P) + sqrt(D);

    int result2 = C\*P\*D;

    // Вывод суммы корней

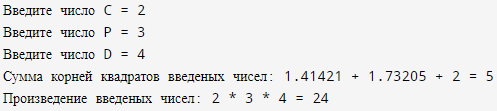
    cout<<"Сумма корней квадратов введеных чисел: "<<sqrt(C)<<" + "<<sqrt(P)<<" + "<<sqrt(D)<<" = "<<result1<<endl;

    // Вывод произведения чисел

    cout<<"Произведение введеных чисел: "<<C<<" \* "<<P<<" \* "<<D<<" = "<<result2;

}

*Результат:*



1. Составить программу для вычисления:



*Код программы:*

#include <iostream>

#include <cmath>

using namespace std;

int main(){

    // Ввод данных

    float a,b,y;

    cout<<"\n Введите число a = ";

    cin>>a;

    cout<<"\n Введите число b = ";

    cin>>b;

    cout<<"\n Введите число y = ";

    cin>>y;

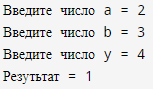
    // Считаем и выводим результат

    int result = (7 \* y + 3 \* sin(a) + sqrt(pow(b, 2) + 19)) / (7 \* y + sqrt(pow(b, 2) + 19) + 2);

    cout<<"Резутьтат = "<<result;

}

*Результат:*



” Составление программ циклической структуры”

1. Составить программу для определения таблицы значений функции у в произвольном диапазоне [a,b] изменения аргумента х с произвольным шагом h, вводимым с клавиатуры. Таблица должна содержать следующие столбцы: порядковый номер, значение аргумента x, значение функции, сообщение о возрастании или убывании функции, разность двух соседних значений функции. Определить максимальное и минимальное значения функции.



*Код программы:*

#include <iostream>

#include <cmath>

using namespace std;

// Функция, для которой строится таблица значений

double computeFunction(double x) {

    // Вычисление значения функции

    return exp(x+1)\*cos(x);

}

int main() {

    // Ввод значений

    double a, b, h;

    cout << "Введите начало диапазона a: ";

    cin >> a;

    cout << "Введите конец диапазона b: ";

    cin >> b;

    cout << "Введите шаг h: ";

    cin >> h;

    // Объявление переменных

    int count = 1;

    double x = a;

    double prevFuncValue = computeFunction(x);

    double maxFuncValue = prevFuncValue;

    double minFuncValue = prevFuncValue;

    // Заголовок таблицы значений

    cout << "-----------------------------------------------------------"<< endl;

    cout << "| № | Значение x | Значение функции | Сообщение| Разность|" << endl;

    cout << "----------------------------------------------------------" << endl;

    // Сравнение возрастает или убывает функция

    while (x <= b) {

        double funcValue = computeFunction(x);

        string message;

        if (funcValue > prevFuncValue) {

            message = "Функция возрастает";

        } else if (funcValue < prevFuncValue) {

            message = "Функция убывает";

        } else {

            message = "Функция не меняется";

        }

        double diff = funcValue - prevFuncValue;

        // Обновляем максимальное и минимальное значения функции

        if (funcValue > maxFuncValue) {

            maxFuncValue = funcValue;

        }

        if (funcValue < minFuncValue) {

            minFuncValue = funcValue;

        }

        // Вывод значений таблицы

        cout << "" << count << "" << x << "" << funcValue << "" << message << "" << diff<< endl;

        // Обновление и увеличение значений

        x += h;

        prevFuncValue = funcValue;

        count++;

    }

    // Вывод максимального и минимального значения функции

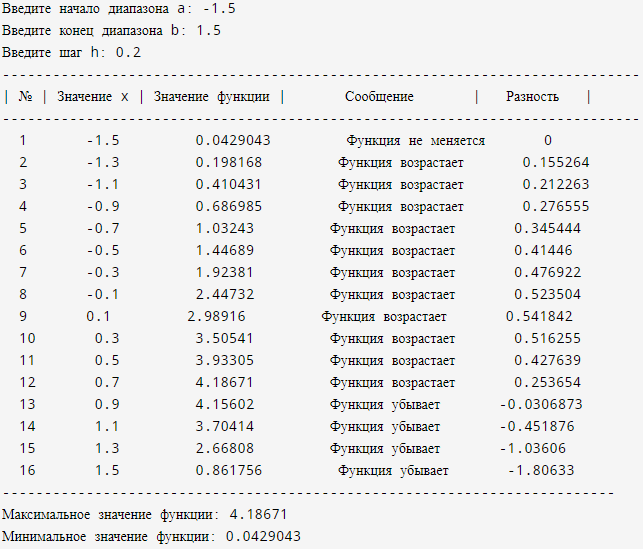
    cout << "----------------------------------------------------------" << endl;

    cout << "Максимальное значение функции: " << maxFuncValue << endl;

    cout << "Минимальное значение функции: " << minFuncValue << endl;

}

*Результат:*



1. Составить программу вывода всех трехзначных десятичных чисел, сумма цифр которых равна данному целому числу.

*Код программы:*

#include <iostream>

using namespace std;

// Получем сумму введенного числа

int getSum(int num) {

    int sum = 0;

    while (num > 0) {

        sum += num % 10;

        num /= 10;

    }

    return sum;

}

int main() {

    // Ввод данных

    int targetSum;

    cout << "Введите целое число: ";

    cin >> targetSum;

    // Вывод заголовка

    cout << "Трехзначные числа, сумма цифр которых равна " << targetSum << ":\n";

    // Нахождение подходящих чесил и вывод их

    for (int num = 100; num <= 999; num++) {

        if (getSum(num) == targetSum) {

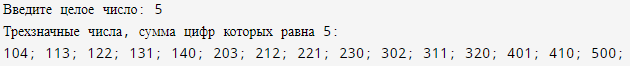
            cout << num << "; ";

        }

    }

}

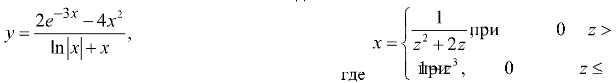
*Результат:*



” Составление программ разветвляющейся структуры”

Вариант №9

1. Составить программу для определения значения функции у. Предусмотреть вывод сообщения о том, по какой ветви происходило вычисление значения аргумента функции х.



*Код программы:*

#include <iostream>

#include <cmath>

using namespace std;

int main() {

    double z, x, y;

    // Ввод данных

    cout << "Введите значение z: ";

    cin >> z;

    // Выбираем по какой ветви пойдет вычисление и выводим результат x

    if (z > 0) {

        x = 1 / (z \* z + 2 \* z);

        cout << "Вычисление значения аргумента х прошло по ветви z > 0" << endl;

    } else {

        x = 1 - pow(z, 3);

        cout << "Вычисление значения аргумента х прошло по ветви z <= 0" << endl;

    }

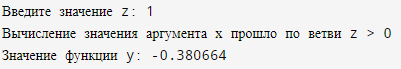
    // Вычисляем значение y и выводим его

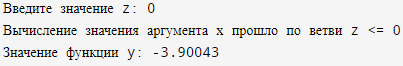
    y = (2 \* exp(-3 \* x) - 4 \* pow(x, 2)) / (log(fabs(x)) + x);

    cout << "Значение функции y: " << y << endl;

}

*Результат:*





1. Написать программу, которая по заданному номеру месяца n определяет количество дней в этом месяце k.

*Код программы:*

#include <iostream>

using namespace std;

int main() {

    // Ввод данных

    int n;

    cout << "Введите номер месяца: ";

    cin >> n;

    int k;

    // Выбираем номер какого месяца ввел пользователь

    switch (n) {

        case 1: // Январь

        case 3: // Март

        case 5: // Май

        case 7: // Июль

        case 8: // Август

        case 10: // Октябрь

        case 12: // Декабрь

            k = 31;

            break;

        case 4: // Апрель

        case 6: // Июнь

        case 9: // Сентябрь

        case 11: // Ноябрь

            k = 30;

            break;

        case 2: // Февраль

            k = 28;

            break;

        // Если такого месяца нет то выводи ошибку

        default:

            cout << "Неверный номер месяца!" << endl;

    }

    // Выводим количество дней в выбранном месяце

    cout << "Количество дней в месяце: " << k << endl;

}

*Результат:*



” Функции пользователя”

Вариант №9

Ответы на вопросы:

1. Чем функция пользователя отличается от стандартной функции? - Функция пользователя (или пользовательская функция) отличается от стандартной функции тем, что пользовательская функция создается программистом самостоятельно для выполнения определенной задачи или набора задач, а стандартные функции уже предопределены в языке программирования и обычно используются для выполнения стандартных операций (например, вывод на экран, сортировка и т. д.).
2. Способы передачи аргументов в функцию? - Способы передачи аргументов в функцию в C++ включают передачу по значению, передачу по ссылке и передачу указателя. При передаче по значению функция получает копию значения аргумента, при передаче по ссылке функция получает возможность изменять оригинальное значение аргумента, а при передаче указателя функция получает доступ к оригинальному значению аргумента через его адрес.
3. Поясните понятие “локальные” и “глобальные” переменные? - Локальные переменные - это переменные, объявленные внутри блока кода функции или другого компонента программы, и их область видимости ограничена этим блоком кода. Глобальные переменные - это переменные, объявленные вне всех функций и их область видимости распространяется на всю программу.
4. Для чего и каким образом применяется оператор return? - Оператор return используется в функции для возврата значения обратно в вызывающую программу. Это позволяет функции выполнить вычисления, обработку данных и вернуть результат для дальнейшего использования в программе. Кроме того, оператор return также может использоваться для завершения выполнения функции в случае, если функция ничего не возвращает (например, функция типа void).

Задание: Дано натуральное число N. Получить все такие натуральные K, что N делится на K\*\*2 и не делится на K\*\*3.

*Код программы:*

#include <iostream>

using namespace std;

// Функция для проверки, делится ли число N на K^2

bool isDivisibleByKSquare(int N, int K) {

    return N % (K \* K) == 0;

}

// Функция для проверки, не делится ли число N на K^3

bool isNotDivisibleByKCubed(int N, int K) {

    return N % (K \* K \* K) != 0;

}

int main() {

    int N;

    cout << "Введите натуральное число N: ";

    cin >> N;

    cout << "Все натуральные K, такие что N делится на K\*\*2 и не делится на K\*\*3: ";

    for (int K = 1; K <= N; K++) {

        if (isDivisibleByKSquare(N, K) && isNotDivisibleByKCubed(N, K)) {

            cout << K << "; ";

        }

    }

}

*Результат:*

**

” Обработка одномерных массивов”

Вариант №9

Ответы на вопросы:

1. Определение массива - Массив - это структура данных, которая представляет собой набор элементов одного типа, расположенных последовательно в памяти.
2. Укажите типы массивов, применяемых в языке С - В языке C можно использовать следующие типы массивов: Одномерные массивы, Многомерные массивы
3. Правила описания массива в программе на С - Правила описания массива в программе на языке C: Массив объявляется указанием типа его элементов, за которым следует имя массива и размер массива в квадратных скобках. Например: int myArray[10]; Размер массива должен быть целым положительным числом или выражением, вычисляемым на этапе компиляции. Имена массивов ведут себя как указатели на первый элемент массива.
4. Формы (способы) работы с элементами массива - Формы (способы) работы с элементами массива: Чтение и запись значений элементов массива по их индексам. Использование циклов (например, for, while) для итерации по элементам массива. Передача массива в функцию в качестве аргумента. Использование указателей для работы с элементами массива.

Задание: Найти среднее арифметическое элементов массива х(n), исключив из них максимальное и минимальное значение.

*Код программы:*

#include <iostream>

#include <random>

#include <limits>

using namespace std;

int main() {

    int n;

    cout << "Введите размер массива: ";

    cin >> n;

    // Инициализация генератора случайных чисел

    random\_device rd;

    mt19937 gen(rd());

    uniform\_int\_distribution<int> distrib(1, 10);

    int\* x = new int[n];

    cout << "Элементы массива: ";

    for (int i = 0; i < n; i++) {

        x[i] = distrib(gen);

        cout << x[i] << " ";

    }

    cout << endl;

    // Находим минимальное и максимальное значение в массиве

    int minVal = numeric\_limits<int>::max();

    int maxVal = numeric\_limits<int>::min();

    for (int i = 0; i < n; i++) {

        if (x[i] < minVal) {

            minVal = x[i];

        }

        if (x[i] > maxVal) {

            maxVal = x[i];

        }

    }

    int sum = 0;

    int count = 0;

    // Находим сумму всех элементов, исключая минимальное и максимальное значение

    for (int i = 0; i < n; i++) {

        if (x[i] != minVal && x[i] != maxVal) {

            sum += x[i];

            count++;

        }

    }

    if (count > 0) {

        double average = static\_cast<double>(sum) / count;

        cout << "Минимальное значение массива: " << minVal << endl;

        cout << "Максимальное значение массива: " << maxVal << endl;

        cout << "Элементы массива без минимальных и максимальных значений: ";

            for (int i = 0; i < n; i++) {

                if(x[i] != minVal && x[i] != maxVal){

                    cout << x[i] << " ";

                }

            }

        cout << endl;

        cout << "Среднее арифметическое элементов массива (исключив минимальное и максимальное значение): " <<sum<<" / "<<count<<" = "<< average << endl;

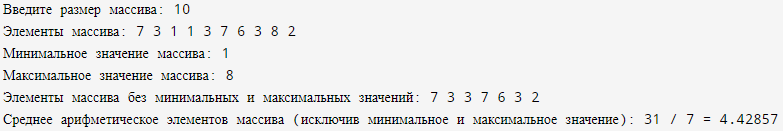
    } else {

        cout << "В массиве нет уникальных элементов для вычисления среднего значения." << endl;

    }

}

*Результат:*

**

” Обработка двумерных массивов”

Вариант №19

Ответы на вопросы:

1. Укажите способы декларирования двумерных массивов - Способы декларирования двумерных массивов: Статическое декларирование, Динамическое декларирование, Декларирование с помощью векторов.
2. Способы заполнения двумерных массивов - Способы заполнения двумерных массивов: Ввод значений с клавиатуры, Инициализация значений при декларировании, Заполнение случайными числами.
3. Формы (способы) работы с элементами двумерного массива - Формы (способы) работы с элементами двумерного массива: Обращение к конкретному элементу, Использование вложенных циклов для перебора всех элементов, Преобразование двумерного массива в одномерный.

Задание: Дан двумерный массив. Выяснить, является ли произведение элементов n-го столбца массива трехзначным числом.

*Код программы:*

#include <iostream>

#include <random>

using namespace std;

const int rows = 3;

const int colon = 3;

// Функция для проверки, является ли произведение элементов столбца трехзначным числом

bool isThreeDigitProduct(int arr[][colon], int colIndex, int& product) {

    product = 1;

    for (int i = 0; i < rows; ++i) {

        product \*= arr[i][colIndex]; // Вычисление произведения элементов столбца

    }

    return (product >= 100 && product <= 999); // Проверка, является ли произведение трехзначным числом

}

int main() {

    int arr[rows][colon];

    // Инициализация генератора случайных чисел

    random\_device rd;

    mt19937 gen(rd());

    uniform\_int\_distribution<int> distrib(1, 10);

    // Заполнение двумерного массива случайными числами

    for (int i = 0; i < rows; i++) {

        for (int j = 0; j < colon; j++) {

            arr[i][j] = distrib(gen);

        }

    }

    int colIndex;

    int product;

    cout << "Массив:" << endl;

    for (int i = 0; i < rows; i++) {

        for (int j = 0; j < colon; j++) {

            cout << arr[i][j] << " ";

        }

        cout << endl;

    }

    cout << "Введите номер столбца (от 0 до " << colon - 1 << "): ";

    cin >> colIndex;

    if (colIndex < 0 || colIndex >= colon) {

        cout << "Ошибка: Некорректный номер столбца."; // В случае некорректного ввода номера столбца

    } else {

        if (isThreeDigitProduct(arr, colIndex, product)) { // Проверка произведения элементов столбца

            cout << "Произведение элементов " << colIndex << "-го столбца является трехзначным числом: " << product<< endl; // Вывод результата (и произведения)

        } else {

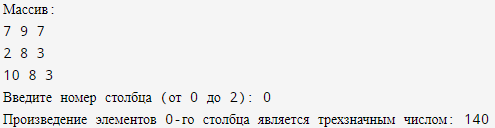
            cout << "Произведение элементов " << colIndex << "-го столбца НЕ является трехзначным числом: " << product<< endl; // Вывод результата (и произведения)

        }

    }

}

*Результат:*



” Обработка символьных строк”

Вариант №9

Ответы на вопросы:

1. Как описываются строки в языке С? - В языке C строки обычно описываются как массивы символов, завершающихся нулевым символом (символ конца строки). Например, строка "Hello" будет представлена в виде массива символов ['H', 'e', 'l', 'l', 'o', '\0'].
2. Чем отличаются функции scanf() и gets(), printf() и puts()? - Функции scanf() и gets() используются для ввода строк. Однако, gets() принимает ввод из стандартного входного потока (обычно клавиатуры) и сохраняет его в указанной переменной как строку, в то время как scanf() позволяет задавать формат ввода и может пропускать пробелы и символы новой строки. Аналогично, printf() и puts() используются для вывода строк. printf() дает возможность форматировать вывод, в то время как puts() просто выводит строку на экран с добавлением символа новой строки в конце.
3. Что понимают под обработкой строк? - Под обработкой строк понимается выполнение различных операций со строками, таких как поиск подстроки, разделение строки на части, конкатенация строк, замена символов и другие манипуляции. Обработка строк может включать в себя использование различных функций библиотеки C для работы со строками, таких как strlen(), strcpy(), strcat(), strcmp() и других. Также обработка строк может включать в себя работу с указателями на символы.

Задание: Ввести с клавиатуры строку символов. Признак окончания ввода строки – нажатие клавиши "Ввод". Программа должна определить длину введенной строки L, и, если длина L кратна 5, то подсчитывается количество скобок все.

*Код программы:*

#include <iostream>

#include <string>

using namespace std;

int main() {

    // Ввод данных

    cout << "Введите любое количество скобок: ";

    string input;

    getline(cin, input);

    int L = input.length();

    // Объявляем переменные для скобок по их названию

    if (L % 5 == 0) {

        int count\_open = 0;

        int count\_close = 0;

        int count\_curly\_open = 0;

        int count\_curly\_close = 0;

        int count\_square\_open = 0;

        int count\_square\_close = 0;

        // Находим количество видов всех скобок

        for (char c : input) {

            if (c == '(' ) {

                count\_open++;

            } else if (c == ')') {

                count\_close++;

            } else if (c == '{') {

                count\_curly\_open++;

            } else if (c == '}') {

                count\_curly\_close++;

            } else if (c == '[') {

                count\_square\_open++;

            } else if (c == ']') {

                count\_square\_close++;

            }

        }

        // Выводим количество скобок всех видов

        cout << "Количество круглых скобок: " << count\_open + count\_close << endl;

        cout << "Количество фигурных скобок: " << count\_curly\_open + count\_curly\_close << endl;

        cout << "Количество квадратных скобок: " << count\_square\_open + count\_square\_close << endl;

    }

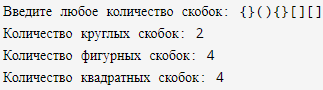
    else{

        cout << "Введенное количество символов не кратно 5";

    }

}

*Результат:*



” Программирование алгоритмов с использованием структур”

Вариант №19

Ответы на вопросы:

1. Как описываются структуры в языке С? - Описание структур в языке C: Структуры в языке C описываются с использованием ключевого слова struct, после которого идет имя структуры и фигурные скобки, в которых объявляются поля структуры.
2. Как получить доступ к элементам структуры? - Доступ к элементам структуры: Доступ к элементам структуры осуществляется с использованием оператора точки (.), и имя структуры, за которым следует точка и имя поля.
3. Поясните термин “указатель на структуру”? - Указатель на структуру: Указатель на структуру - это переменная, которая содержит адрес памяти структуры. Она позволяет получить доступ к полям структуры и выполнять операции с самой структурой, используя указатель и оператор разыменования (\*).

Задание: Известны баллы, набранные каждым из 10 спортсменов-пятиборцев в каждом из пяти видов спорта. Определить фамилию спортсмена-победителя соревнований.

*Код программы:*

#include <iostream>

#include <string>

using namespace std;

const int numAthletes = 10;

const int numSports = 5;

// Структура для хранения данных о спортсмене

struct Athlete {

    string name;

    int scores[numSports];

};

// Функция для вычисления общего балла спортсмена

int getTotalScore(const Athlete& athlete) {

    int total = 0;

    for (int i = 0; i < numSports; i++) {

        total += athlete.scores[i];

    }

    return total;

}

int main() {

    // Инициализация данных спортсменов

    Athlete athletes[numAthletes] = {

        {"Иванов", {10, 5, 8, 7, 9}},

        {"Петров", {8, 7, 6, 5, 9}},

        {"Сидоров", {9, 8, 7, 6, 5}},

        {"Козлов", {8, 7, 8, 9, 10}},

        {"Михайлов", {7, 7, 7, 7, 7}},

        {"Александров", {10, 9, 8, 7, 6}},

        {"Дмитриев", {5, 6, 7, 8, 9}},

        {"Андреев", {8, 8, 8, 8, 8}},

        {"Сергеев", {7, 6, 8, 5, 9}},

        {"Павлов", {9, 6, 8, 7, 5}}

    };

    // Вывод информации о баллах каждого участника

    cout << "Информация о баллах участников:" << endl;

    for (int i = 0; i < numAthletes; i++) {

        int totalScore = getTotalScore(athletes[i]);

        cout << athletes[i].name << ": " << totalScore << " очков" << endl;

    }

    // Поиск спортсмена с наибольшим общим баллом

    int maxScore = getTotalScore(athletes[0]);

    string winner = athletes[0].name;

    for (int i = 1; i < numAthletes; i++) {

        int totalScore = getTotalScore(athletes[i]);

        if (totalScore > maxScore) {

            maxScore = totalScore;

            winner = athletes[i].name;

        }

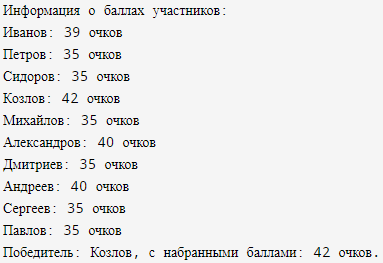
    }

    // Вывод результата

    cout << "Победитель: " << winner << ", с набранными баллами: " << maxScore << " очков." << endl;

}

*Результат:*



” Работа с файлами в С/С++”

Вариант №19

Ответы на вопросы:

1. Что называется физическим и логическим файлами? - Физический файл - это абстрактный набор данных, хранящийся на диске или другом постоянном носителе. Физический файл представляет собой непрерывную последовательность байтов, которая хранится на диске. Логический файл - это абстрактный набор данных, который представляется операционной системой как последовательность данных, доступных для программы ввода/вывода. Логический файл может быть ассоциирован с физическим файлом, но может также представлять данные, которые могут быть произведены программно, например, через сетевое соединение или другие источники данных.
2. Правила и способы открытия файла в языке С - В языке программирования C открытие файла осуществляется с помощью функции fopen(). Эта функция принимает два параметра: имя файла и режим открытия файла. Режимы открытия могут быть "r" (открытие файла для чтения), "w" (открытие файла для записи, если файл уже существует, его содержимое удаляется, если файла нет, то создается новый), "a" (открытие файла для записи в конец файла, если файл не существует, создается новый), "r+" (открытие файла для чтения и записи), "w+" (открытие файла для чтения и записи, файл усекается до нулевой длины, если файл не существует, он создается), "a+" (открытие файла для чтения и записи в конец файла, если файла не существует, создается новый).
3. Порядок работы с файлами –

FILE \*file = fopen("example.txt", "w");

if (file == NULL) {

    printf("Ошибка открытия файла");

} else {

    // Запись данных в файл

    fprintf(file, "Hello, World!");

    fclose(file); // Закрытие файла

}

Задание: Составить программу записи в файл GS всех чисел файла FS кратных 5, а файл HS – всех отрицательных чисел кратных 3.

*Код программы:*

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

struct FileData

{

    vector<int> data;

};

void createNewFile(const string& filename)

{

    cout << "Создание нового файла " << filename << endl;

    ofstream file(filename);  // Создание нового файла

    // Здесь можно предусмотреть ввод данных пользователем и запись их в файл

    file.close();  // Закрытие файла

}

void viewFile(const string& filename)

{

    cout << "Просмотр содержимого файла " << filename << endl;

    ifstream file(filename);  // Открытие файла для чтения

    file.close();  // Закрытие файла

}

void correctFile(const string& filename)

{

    cout << "Коррекция файла " << filename << endl;

    ofstream file(filename, ios::app);  // Открытие файла для дозаписи новых данных или исправления старых

       file.close();  // Закрытие файла

}

void writeDataToFile(const string& filename, const FileData& fileData)

{

    ofstream file(filename);  // Открытие файла для записи

    if (file.is\_open()) {

        for (int val : fileData.data) {

            file << val << " ";  // Запись данных в файл

        }

        file.close();  // Закрытие файла

        cout << "Данные успешно записаны в файл " << filename << endl;

    } else {

        cout << "Ошибка при открытии файла " << filename << " для записи данных" << endl;

    }

}

int main() {

    setlocale(LC\_ALL, "ru");  // Установка локали для вывода на русском языке

    const string fsFilename = "FS.txt";

    const string hsFilename = "HS.txt";

    // Создание структуры для данных, которые будут записаны в файл

    FileData fsData;

    FileData hsData;

    // Наполнение структуры данными, кратными 5 и отрицательными числами кратными 3

    for (int i = -50; i <= 50; ++i) {

        if (i % 5 == 0) {

            fsData.data.push\_back(i);

        }

        if (i % 3 == 0 && i < 0) {

            hsData.data.push\_back(i);

        }

    }

    // Запись данных в соответствующие файлы

    writeDataToFile(fsFilename, fsData);

    writeDataToFile(hsFilename, hsData);

    // Пример использования создания нового файла, просмотра и коррекции

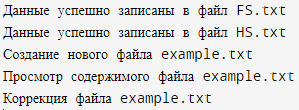
    createNewFile("example.txt");  // Пример создания нового файла

    viewFile("example.txt");  // Пример просмотра содержимого файла

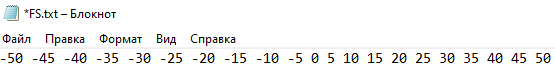
    correctFile("example.txt");  // Пример коррекции файла

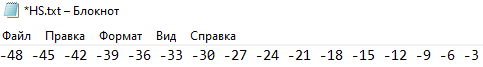
}

*Результат:*









Создание Windows-приложений

Цель работы: создать просто оконное windows приложение.

Созданное мною приложение: я создал простое оконное windows приложение, в котором при нажатии на кнопку закрыть оконное приложение закрывается, а при нажатии на кнопку вывести выводится 1 факт о с++ из массива, при выводе все фактов факты из массива выводятся повторно.

Код программы:

#include <windows.h>

#include <iostream>

#include <vector>

#include <string>

#include <ctime>

// Глобальные переменные

HINSTANCE hInst;

HWND hwndButtonClose, hwndButtonFact;

// Вспомогательная функция для получения случайного факта о C++

std::wstring GetRandomFact() {

    std::vector<std::wstring> facts = {

        L"C++ был разработан в 1983 году Бьярном Страуструпом.",

        L"C++ - это расширение языка программирования Си.",

        L"C++ поддерживает объектно-ориентированное, процедурное и обобщенное программирование.",

        L"Стандарт C++ обновляется каждые несколько лет; последний стандарт - C++17.",

        L"C++ широко используется для создания операционных систем, игр, браузеров и других приложений.",

        L"C++ является одним из самых популярных языков программирования в мире."

    };

    // Генерируем случайный индекс для выбора факта

    srand(static\_cast<unsigned int>(time(0)));

    int index = rand() % facts.size();

    return facts[index];

}

// Обработчик сообщений окна

LRESULT CALLBACK WndProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lParam) {

    switch (msg) {

    case WM\_CREATE:

        // Создаем кнопку "Закрыть окно"

        hwndButtonClose = CreateWindow(

            L"BUTTON",   // класс кнопки

            L"Закрыть окно",  // текст кнопки

            WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,  // стиль кнопки

            50, 50, 150, 50,  // координаты и размеры кнопки

            hwnd,  // дескриптор родительского окна

            (HMENU)1,  // идентификатор кнопки

            hInst, NULL

        );

        // Создаем кнопку "Вывести факт"

        hwndButtonFact = CreateWindow(

            L"BUTTON",

            L"Вывести факт",

            WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

            50, 120, 150, 50,

            hwnd,

            (HMENU)2,

            hInst, NULL

        );

        break;

    case WM\_COMMAND:

        // Обработка нажатий кнопок

        switch (LOWORD(wParam)) {

        case 1:

            // Нажата кнопка "Закрыть окно"

            DestroyWindow(hwnd);

            break;

        case 2:

            // Нажата кнопка "Вывести факт"

            MessageBoxW(hwnd, GetRandomFact().c\_str(), L"Факт о C++", MB\_OK | MB\_ICONINFORMATION);

            break;

        }

        break;

    case WM\_DESTROY:

        // Завершение работы приложения при закрытии окна

        PostQuitMessage(0);

        break;

    default:

        return DefWindowProc(hwnd, msg, wParam, lParam);

    }

    return 0;

}

// Функция для создания и отображения окна

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow) {

    // Регистрация класса окна

    WNDCLASSEX wc = { sizeof(WNDCLASSEX), CS\_HREDRAW | CS\_VREDRAW, WndProc, 0, 0, hInstance, NULL, NULL, NULL, NULL, L"WindowClass", NULL };

    RegisterClassEx(&wc);

    // Создание окна

    HWND hwnd = CreateWindowEx(0, L"WindowClass", L"Пример приложения на C++", WS\_OVERLAPPEDWINDOW, CW\_USEDEFAULT, CW\_USEDEFAULT, 350, 250, NULL, NULL, hInstance, NULL);

    // Отображение окна

    ShowWindow(hwnd, nCmdShow);

    UpdateWindow(hwnd);

    // Цикл обработки сообщений

    MSG msg;

    while (GetMessage(&msg, NULL, 0, 0)) {

        TranslateMessage(&msg);

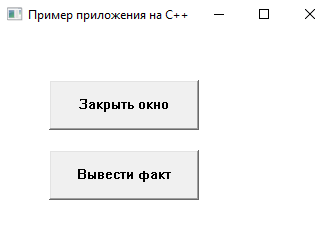
        DispatchMessage(&msg);

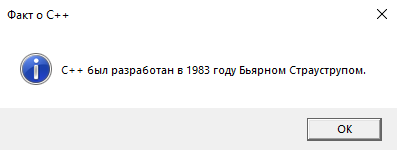
    }

    return msg.wParam;

}

Результат:





Создание традиционного классического приложения Windows

Цель работы: создать оконное windows приложение с функционалом.

Созданное мною приложение: я создал игру “угадай число” в оконном приложении. При запуске приложения есть окно ввода, в которое мы вводим число и 3 кнопки – проверить правильно ли мы ввели число при нажатии на которую нам дают подсказку (введеное нами число больше или меньше загаданного), закрыть игру чтобы закрыть окно и начать заново, чтобы загадать новое число после того, как отгадали предыдущее.

Код программы:

#include <windows.h>

#include <string>

#include <ctime>

#include <cstdlib>

// Глобальные переменные

HWND hWndButtonCheck;

HWND hWndButtonClose;

HWND hWndButtonRestart;

HWND hWndEdit;

int secretNumber;

bool guessed = false;

// Функция для создания окна

HWND CreateMainWindow(HINSTANCE hInstance, int nCmdShow);

// Функция для обработки сообщений окна

LRESULT CALLBACK WindowProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam);

// Функция для обработки нажатия кнопки "Проверить"

void OnCheckButtonClicked(HWND hWnd);

// Функция для обработки нажатия кнопки "Закрыть игру"

void OnCloseButtonClicked(HWND hWnd);

// Функция для обработки нажатия кнопки "Начать заново"

void OnRestartButtonClicked(HWND hWnd);

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow) {

    srand(time(nullptr)); // Инициализация генератора случайных чисел

    secretNumber = rand() % 100 + 1; // Загаданное число от 1 до 100

    // Создание главного окна

    HWND hWnd = CreateMainWindow(hInstance, nCmdShow);

    // Цикл обработки сообщений

    MSG msg;

    while (GetMessage(&msg, nullptr, 0, 0)) {

        TranslateMessage(&msg);

        DispatchMessage(&msg);

    }

    return msg.wParam;

}

HWND CreateMainWindow(HINSTANCE hInstance, int nCmdShow) {

    // Регистрация класса окна

    WNDCLASS wc = {};

    wc.lpfnWndProc = WindowProc;

    wc.hInstance = hInstance;

    wc.lpszClassName = L"MainWindowClass";

    RegisterClass(&wc);

    // Создание окна

    HWND hWnd = CreateWindowEx(

        0,                          // Расширенные стили окна

        L"MainWindowClass",         // Имя класса окна

        L"Угадай число",            // Заголовок окна

        WS\_OVERLAPPEDWINDOW,        // Стиль окна

        CW\_USEDEFAULT, CW\_USEDEFAULT, 400, 200, // Размеры и позиция окна

        nullptr,                    // Родительское окно

        nullptr,                    // Меню окна

        hInstance,                  // Дескриптор экземпляра приложения

        nullptr                     // Дополнительные параметры создания окна

    );

    // Создание кнопки "Проверить"

    hWndButtonCheck = CreateWindow(

        L"BUTTON",                  // Имя класса кнопки

        L"Проверить",               // Текст кнопки

        WS\_TABSTOP | WS\_VISIBLE | WS\_CHILD | BS\_DEFPUSHBUTTON, // Стиль кнопки

        155, 100, 80, 30,           // Размеры и позиция кнопки

        hWnd,                       // Родительское окно

        nullptr,                    // Идентификатор кнопки

        hInstance,                  // Дескриптор экземпляра приложения

        nullptr                     // Дополнительные параметры создания кнопки

    );

    // Создание кнопки "Закрыть игру"

    hWndButtonClose = CreateWindow(

        L"BUTTON",                  // Имя класса кнопки

        L"Закрыть игру",            // Текст кнопки

        WS\_TABSTOP | WS\_VISIBLE | WS\_CHILD | BS\_DEFPUSHBUTTON, // Стиль кнопки

        250, 100, 100, 30,          // Размеры и позиция кнопки

        hWnd,                       // Родительское окно

        nullptr,                    // Идентификатор кнопки

        hInstance,                  // Дескриптор экземпляра приложения

        nullptr                     // Дополнительные параметры создания кнопки

    );

    // Создание кнопки "Начать заново"

    hWndButtonRestart = CreateWindow(

        L"BUTTON",                  // Имя класса кнопки

        L"Начать заново",           // Текст кнопки

        WS\_TABSTOP | WS\_VISIBLE | WS\_CHILD | BS\_DEFPUSHBUTTON | WS\_DISABLED, // Стиль кнопки

        30, 100, 105, 30,           // Размеры и позиция кнопки

        hWnd,                       // Родительское окно

        nullptr,                    // Идентификатор кнопки

        hInstance,                  // Дескриптор экземпляра приложения

        nullptr                     // Дополнительные параметры создания кнопки

    );

    // Создание текстового поля для ввода числа

    hWndEdit = CreateWindow(

        L"EDIT",                    // Имя класса текстового поля

        L"",                        // Текст по умолчанию

        WS\_VISIBLE | WS\_CHILD | WS\_BORDER | ES\_NUMBER, // Стиль текстового поля

        150, 50, 100, 30,           // Размеры и позиция текстового поля

        hWnd,                       // Родительское окно

        nullptr,                    // Идентификатор текстового поля

        hInstance,                  // Дескриптор экземпляра приложения

        nullptr                     // Дополнительные параметры создания текстового поля

    );

    // Отображение окна

    ShowWindow(hWnd, nCmdShow);

    UpdateWindow(hWnd);

    return hWnd;

}

LRESULT CALLBACK WindowProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam) {

    switch (uMsg) {

    case WM\_DESTROY:

        PostQuitMessage(0);

        return 0;

    case WM\_COMMAND:

        if (reinterpret\_cast<HWND>(lParam) == hWndButtonCheck && HIWORD(wParam) == BN\_CLICKED) {

            OnCheckButtonClicked(hwnd);

        }

        else if (reinterpret\_cast<HWND>(lParam) == hWndButtonClose && HIWORD(wParam) == BN\_CLICKED) {

            OnCloseButtonClicked(hwnd);

        }

        else if (reinterpret\_cast<HWND>(lParam) == hWndButtonRestart && HIWORD(wParam) == BN\_CLICKED) {

            OnRestartButtonClicked(hwnd);

        }

        return 0;

    default:

        return DefWindowProc(hwnd, uMsg, wParam, lParam);

    }

}

void OnCheckButtonClicked(HWND hWnd) {

    wchar\_t buffer[10];

    GetWindowText(hWndEdit, buffer, 10);

    int guess = \_wtoi(buffer);

    std::wstring message;

    if (guess > secretNumber) {

        message = L"Слишком большое число!";

    }

    else if (guess < secretNumber) {

        message = L"Слишком маленькое число!";

    }

    else {

        message = L"Поздравляю! Вы угадали число!";

        EnableWindow(hWndButtonRestart, TRUE);

        guessed = true;

    }

    MessageBox(hWnd, message.c\_str(), L"Результат", MB\_OK | MB\_ICONINFORMATION);

}

void OnCloseButtonClicked(HWND hWnd) {

    DestroyWindow(hWnd);

}

void OnRestartButtonClicked(HWND hWnd) {

    secretNumber = rand() % 100 + 1;

    EnableWindow(hWndButtonRestart, FALSE);

    guessed = false;

    SetWindowText(hWndEdit, L"");

    MessageBox(hWnd, L"Новое число загадано! Удачи в угадывании!", L"Начать заново", MB\_OK | MB\_ICONINFORMATION);

}

Результат:

