

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 15.05.2022 01:50:23
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра нанотехнологий, микроэлектроники,
общей и прикладной физики



УТВЕРЖДАЮ

Проректор по учебной работе

О.Г. Локтионова

« 10 » 02 2022 г.

МОДЕЛИРОВАНИЕ В МАТЕРИАЛОВЕДЕНИИ
методические указания к выполнению практических работ
для студентов направления подготовки
28.03.01 «Нанотехнологии и микросистемная техника»

Курск 2022

УДК 53

Составители: А.В. Кузько, А.П. Кузьменко, А.Е. Кузько,
М.А. Пугачевский, А.В. Кочура, В.В. Родионов

Рецензент

Проректор по цифровой трансформации, доцент, к.т.н.

А.И. Пыхтин

Моделирование в материаловедении: методические указания к выполнению практических работ для студентов направления подготовки 28.03.01 «Нанотехнологии и микросистемная техника» / Юго-Зап. гос. ун-т; сост.: Кузько А.В., Кузьменко А.П., Кузько А.Е., Пугачевский М.А., Кочура А.В., Родионов В.В. Курск, 2022. 37 с.: Библиогр.: с. 33.

Излагаются методические рекомендации по выполнению практических работ на языках программирования Python и C++, в которых предлагается программное решение заданий с использованием условных операторов, циклов, функций, одномерных массивов, матриц, численного интегрирования методами прямоугольников, трапеций, Симпсона (парабол), методом квадратуры Гаусса с произвольным числом точек, которые необходимы для освоения метода конечных элементов при расчетах поведения материалов.

Методические указания соответствуют требованиям Федерального государственного образовательного стандарта высшего образования и учебного плана направления подготовки 28.03.01 Нанотехнологии и микросистемная техника, степень (квалификация) – бакалавр. Предназначены для студентов всех форм обучения.

Работа выполнена в рамках реализации стратегического проекта по программе развития федерального государственного бюджетного образовательного учреждения высшего образования "Юго-Западный государственный университет", в рамках реализации программы стратегического академического лидерства «Приоритет-2030» (Соглашения № 075-15-2021-1155 и № 075-15-2021-1213, ПРОЕКТ НАНО №1.7.21/3)

Текст печатается в авторской редакции

Подписано в печать 10.02.2022 Формат 60 x 84 1/16.

Усл. печ. л. 2,31. Уч.-изд. л. 1,9. Тираж 50 экз. Заказ 732. Бесплатно.

Юго-Западный государственный университет.

305040 Курск, ул. 50 лет Октября, 94.

ПРАКТИЧЕСКАЯ РАБОТА № 1.

Простейшие программы на языках Python и C++. Типы данных

1) Вывод текста на экран

Python	C++
<pre>print('Hello, world!') print() print('World, hello!') print('Как тебя зовут?') name = input() print('Привет,', name) name = input('Как тебя зовут?') print('Привет,', name) print('a', 'b', 'c') print('a', 'b', 'c', sep='*') print('d', 'e', 'f', sep='\n', end='&&') print('a', 'b', 'c', end='@') print('d', 'e', 'f', end='@@')</pre>	<pre>#include <iostream> using namespace std; main() { cout << "Hello, world! \n"; cout << "Hello, world!" << endl<< endl; setlocale(0, ""); cout << "Привет, мир!" << endl; cout << "Привет, мир! \n"; return 0; }</pre>

2) Дробные числа, типы данных

Python	C++
<pre>num = int(input()) print(2 ** 2 ** 3) # 2 ** (2 ** 3) = 2 ** 8 print(10 // 6) # 1 print(10 // 12) # 0 print(10 % 6) # 4 print(10 % 12) # 10 num = 754 a = num % 10 b = (num % 100) // 10 c = num // 100 print(a, b, c) num1 = 25_000_000 num2 = 25000000 print(num1, num2) e = 2.71828 # литерал с плавающей точкой pi = 3.1415 # литерал с плавающей точкой num = float(input()) # преобразование считанной строки в число с плавающей точкой n = float('1.2345') print(num, n)</pre>	<pre>#include <iostream> main () { int a; int b, c; int d=1./2., e=3, f; unsigned int g=10; double h=1.3, i; double k=1./2; std::cout << "d is " << d << std::endl; std::cout << 7%3<<std::endl; std::cout << "k is " << k; return 0; }</pre>

Задания

I. Дано положительное действительное число. Выведите его первую цифру после десятичной точки.

II. Напишите программу, которая находит наименьшее и наибольшее из пяти чисел.

III. Напишите программу для нахождения цифр четырёхзначного числа.

IV. Напишите программу для пересчёта величины временного интервала, заданного в минутах, в величину, выраженную в часах и минутах.

ПРАКТИЧЕСКАЯ РАБОТА № 2.

Ветвления в языках Python и C++. Условный оператор. Сложные условия

1) Оператор if и условия или/и

Python	C++
<pre> a, b, c = int(input()),int(input()),int(input()) if a != b != c: print('числа не равны') else: print('числа равны') grade = int(input('Введите количество баллов по тесту: ')) if grade >= 90: print(5) elif grade >= 80: print(4) elif grade >= 70: print(3) elif grade >= 60: print(2) else: print(1) age = int(input('Сколько вам лет?: ')) city = input('В каком городе вы живете?: ') if age >= 12 and grade >= 7 and (city == 'Курск' or city == 'Белгород'): print('Доступ разрешен.') else: print('Доступ запрещен.') num = int(input()) flag = True for i in range(2, num): if num % i == 0: # если исходное число делится на какое-либо отличное от 1 и самого себя flag = False if num == 1: print('Это единица, она не простая и не составная') elif flag == True: print('Число простое') else: print('Число составное') </pre>	<pre> #include <iostream> int main() { int a=5, b=600; if (a!=5 b==500) //"!=" - не равно { std::cout << "Condition1 is met" <<std:: endl; } else { std::cout << " Condition1 isn't met "; } if (a==5 && b==600) { std::cout << "Condition2 is met" <<std:: endl; } else { std::cout << " Condition2 isn't met "; } return 0; } #include<iostream> int main() { int a=5, b=500; if (a==6 && b==500) { std::cout<<"Stament1-true\n "<<std::endl; } else if (a==5) { std::cout<<a<<std::endl; std::cout<<"Stament2-true\n"; } else { std::cout<<"staments-false\n"; } } return 0; } </pre>

Задания

I. При регистрации на сайтах требуется вводить пароль дважды. Это сделано для безопасности, поскольку такой подход уменьшает возможность неверного ввода пароля. Напишите программу, которая сравнивает пароль и

его подтверждение. Если они совпадают, то программа выводит: «Пароль принят», иначе: «Пароль не принят».

II. Напишите программу, которая определяет, является число четным или нечетным.

III. Напишите программу, которая принимает три положительных числа и определяет вид треугольника, длины сторон которого равны введенным числам.

IV. Дан порядковый номер месяца (1, 2, ..., 12). Напишите программу, которая выводит на экран количество дней в этом месяце. Принять, что год является невисокосным.

ПРАКТИЧЕСКАЯ РАБОТА № 3.

Циклы в языках Python и C++. Вложенные циклы

1) Циклы for и while

Python	C++
<pre>for i in range(5, 0, -1): print(i, end=' ') print('Взлетаем!!!') counter1 = 0 counter2 = 0 for i in range(10): num = int(input()) if num > 10: counter1 = counter1 + 1 if num == 0: counter2 = counter2 + 1 print('Было введено', counter1, 'чисел, больших 10.') print('Было введено', counter2, 'нулей.') num = int(input()) has_seven = False # сигнальная метка while num != 0: last_digit = num % 10 if last_digit == 7: has_seven = True num = num // 10 if has_seven == True: print('В числе есть цифра 7') else: print('В числе нет цифры 7') for i in range(3): for j in range(3): print(i, j)</pre>	<pre>Найти сумму чисел от 0 до 9 #include <iostream> int main() { int s=0; for (int i=0; i<10; i++) { std::cout << i << std::endl; s+= i;// s=s+ i } std::cout << "S="; std::cout << s << std::endl; return 0; } Найти факториал числа #include <iostream> int main() { unsigned int a, p=1; std::cout << "N! N="; std::cin >>a; for (int i=1; i<=a; i++) { std::cout << i << std::endl; p=p*i; } std::cout << "N!="<<p; return 0; }</pre>

Задания

I. Даны два целых числа m и n ($m \leq n$). Напишите программу, которая выводит все числа от m до n включительно.

II. Дано натуральное число n . Напишите программу, которая выводит таблицу умножения на n .

III. Дано натуральное число n , ($n \geq 10$). Напишите программу, которая определяет его максимальную и минимальную цифры.

IV. На вход программе подается натуральное число n . Напишите программу, которая находит цифровой корень данного числа. Цифровой корень числа n получается следующим образом: если сложить все цифры этого числа, затем все цифры найденной суммы и повторить этот процесс, то

в результате будет получено однозначное число (цифра), которое и называется цифровым корнем данного числа.

V. Дано натуральное число n . Напишите программу, которая выводит значение суммы $1!+2!+3!+\dots+n!$

ПРАКТИЧЕСКАЯ РАБОТА № 4.

Функции в языках Python и C++

1) Функции с параметрами и возвратом значений

Python	C++
<p>Функция распечатки прямоугольника из звездочек (*) с заданными высотой и шириной</p> <pre>def draw_box(height, width): # функция принимает два параметра for i in range(height): print('*' * width) draw_box(5, 7)</pre> <p>Функция проверки корректности вводимого числа</p> <pre>model = int(input()) while model != 100 and model != 200 and model != 300: print('Допустимыми номерами моделей являются 100, 200 и 300.') model = int(input()) или def is_invalid(model): if model != 100 and model != 200 and model != 300: return True else: return False while is_invalid(model): print('Допустимыми номерами моделей являются 100, 200 и 300.') model = int(input())</pre> <p>Функции с несколькими выводами</p> <pre>def get_powers(num): return num**2, num**3, num**4 a, b, c = get_powers(2) print(a) print(b) print(c)</pre>	<p>Функция нахождения площади треугольника</p> <pre>#include <iostream> #include <vector> double triangle_area(double base, double height) { double area; area=1./2*base*height; return area; } int main() { double area; area=triangle_area(2, 3); std::cout << area << std::endl; return 0; } или #include <iostream> double triangle_area(double base, double height) {return 1./2*base*height;} int main() { std::cout << triangle_area(2, 3) << std::endl; return 0; } Функции с несколькими выводами #include <iostream> #include <cmath> void surface_perimeter(double a, double b, double &S, double &P) { S=a*b/2.; P=a+b+sqrt(a*a+b*b); } int main() { double a,r; surface_perimeter(3,4,a,r); std::cout << "S = " << a << ", " << "P = " << r << "; " << std::endl; return 0; }</pre>

Задания

I. Напишите функцию `draw_box()`, которая выводит звездный прямоугольник (*) с размерами 14×10 .

II. Напишите функцию `print_digit_sum()`, которая принимает одно целое число `num` и выводит на печать сумму его цифр.

III. Напишите функцию `get_middle_point(x1, y1, x2, y2)`, которая принимает в качестве аргументов координаты концов отрезка (`x1; y1` и `x2; y2`) и возвращает координаты точки являющейся серединой данного отрезка.

IV. Напишите функцию `is_valid_triangle(side1, side2, side3)`, которая принимает в качестве аргументов три натуральных числа, и возвращает значение `True` если существует невырожденный треугольник со сторонами `side1, side2, side3` и `False` в противном случае.

V. Напишите функцию `is_one_away(word1, word2)`, которая принимает в качестве аргументов два слова `word1` и `word2` и возвращает значение `True` если слова имеют одинаковую длину и отличаются ровно в 1 символе и `False` в противном случае.

ПРАКТИЧЕСКАЯ РАБОТА № 5.

Списки в языке Python. Массивы в языке C++. Алгоритмы обработки массивов

1) Списки

Python	C++
<pre> mlist = [] # пустой список mylist = list() # пустой список numbers = [2, 4, 6, 8, 10] print('numbers', numbers) print(len(numbers)) print('Сумма всех элементов списка =', sum(numbers)) print('numbers[4]=', numbers[4]) numbers1 = list(range(5)) print('numbers1', numbers1) print(len(numbers1)) print('Сумма всех элементов списка =', sum(numbers1)) numbers2 = [] for i in range(10): numbers2.append(i) print('numbers2', numbers2) numbers3 = [i for i in range(10)] print('numbers3', numbers3) even_numbers = list(range(0, 10, 2)) print('even_numbers', even_numbers) odd_numbers = list(range(1, 10, 2)) print('odd_numbers', odd_numbers) evens = [i for i in range(21) if i % 2 == 0] print('evens', evens) print() s = 'abcde' print(s) chars = list(s) print('chars', chars) chars1 = [c for c in 'abcdefg'] print('chars1', chars1) ss = 'aa bb cc dd ff' print(ss) l = ss.split() print(l) print() zeros = [] for i in range(10): zeros.append(0) print('zeros', zeros) zero = [0] * 10 print('zero', zero) numbers01 = [i * j for i in range(1, 5) for j in range(2)] </pre>	<pre> #include <iostream> #include <vector> main() { std::vector<double> vec1; std::vector<double> vec2(3); std::vector<double> vec3(3,1.), vec4(4); vec1.resize(5); std::cout<<vec1.size()<<std::endl; std::cout<<vec3[2]<<std::endl; vec4.push_back(38.5); std::cout<<vec4.size()<<std::endl; std::cout<<vec4[4]<<std::endl; return 0; } </pre> <p>Преобразовать компоненты вектора и найти их сумму компонент вектора</p> <pre> #include <iostream> #include <vector> int main() { int s=0; std::vector <double> mari (10,5.); for (int i=0; i<10; i++) // i=i+1 { mari[i]+=i;// mari[i]=mari[i]+i; std::cout << mari[i] << std::endl; s+=mari[i]; } std::cout <<"S="; std::cout << s << std::endl; return 0; } </pre> <p>Распечатка компонент вектора с помощью функций</p> <pre> #include<iostream> #include<vector> void print_vec (std::vector <double> out) { for(int i=0; i<out.size(); i++) { std::cout << out[i] << std::endl; } } int main() { std::vector <double> test(3, 1.2); print_vec(test); return 0; } </pre>

<pre> print('numbers01', numbers01) if 2 in numbers: print('Список numbers содержит число 2') else: print('Список numbers не содержит число 2') print() for i in range(len(numbers)): print(numbers[i]) print() for num in numbers: print(num) print() for num in numbers: print(num, end=' ') print() print(*numbers) print() print(*numbers, sep='\n') print() numbers5 = [2, 6, 3, 14, 10, 4, 11, 16, 12, 5, 4, 16, 1, 0, 8, 16, 10, 10, 8, 5, 1, 11, 10, 10, 12, 0, 0, 6, 14, 8, 2, 12, 14, 5, 6, 12, 1, 2, 10, 14, 9, 1, 15, 1, 2, 14, 16, 6, 7, 5] print(*numbers5) print(len(numbers5)) print(numbers5[-1]) print(numbers5[::-1]) if 5 in numbers5 and 17 in numbers5: print('YES') else: print('NO') del numbers5[0] del numbers5[-1] print(*numbers5) print() a = [1, 7, -3, 9, 0, -67, 34, 12, 45, 1000, 6, 8, -2, 99] print(a) n = len(a) for i in range(n - 1): for j in range(n - i - 1): if a[j] > a[j + 1]: # если порядок элементов пары неправильный a[j], a[j + 1] = a[j + 1], a[j] # меняем элементы пары местами print('Отсортированный список:', a) </pre>	<pre> } #include <iostream> #include <vector> using namespace std; void print_vec(std::vector<double> NAME) { for(int i=0; i< NAME.size(); i++) { cout << NAME[i] << endl << endl; } } main() { vector<double> vec1; vector<double> vec2(3); vector<double> vec3(3,1.), vec4(4); cout << "Komponenty vectora vec1" << endl; print_vec(vec1); cout << "Komponenty vectora vec2" << endl; print_vec(vec2); cout << "Komponenty vectora vec3" << endl; print_vec(vec3); cout << "Komponenty vectora vec4" << endl; print_vec(vec4); vec4.push_back(38.5); vec1.resize(5); cout << "Komponenty vectora vec1" << endl; print_vec(vec1); cout << "Komponenty vectora vec2" << endl; print_vec(vec2); cout << "Komponenty vectora vec3" << endl; print_vec(vec3); cout << "Komponenty vectora vec4" << endl; print_vec(vec4); return 0;} </pre>
---	--

Задания

I. Задайте список четных цифр от 2 до 20 включительно и рассчитайте среднее арифметическое его элементов.

II. Напишите программу, выводящую следующий список:

['a', 'bb', 'ccc', 'dddd', 'eeeee', 'ffffff', ...]

III. На вход программе подается натуральное число n . Напишите программу, которая создает список, состоящий из делителей введенного числа.

IV. На вход программе подается натуральное число $n \geq 2$, а затем n целых чисел. Напишите программу, которая создает из указанных чисел список, состоящий из сумм соседних чисел (0 и 1, 1 и 2, 2 и 3 и т.д.).

V. На вход программе подается строка текста, содержащая различные натуральные числа. Из данной строки формируется список чисел. Напишите программу, которая меняет местами минимальный и максимальный элемент этого списка.

VI. Используя списочное выражение, распечатайте список всех чисел палиндромов от 100 до 1000.

VII. На вход программе подается натуральное число n . Напишите программу, использующую списочное выражение, которая создает список, содержащий кубы чисел от 1 до n , а затем выводит его элементы построчно, то есть каждый на отдельной строке.

VIII. На вход программе подается строка текста, содержащая целые числа. Напишите программу, использующую списочное выражение, которая выведет квадраты четных чисел, которые не оканчиваются на цифру 4.

ПРАКТИЧЕСКАЯ РАБОТА № 6.

Матрицы в языках Python и C++. Обработка элементов матрицы

1) Двумерные массивы

Python	C++
<pre> n = 10 def print_mtrx(a): [print(*[str(a[i][j]).ljust(5) for j in range(n)]) for i in range(n)] def zero_mtrx(): b = [[0] * n for i in range(n)] return b # ввод матрицы 10x10 значений от 1 до 100 mxlist = [[i + j for j in range(1, 11)] for i in range(0, 100, 10)] print_mtrx(mxlist) print() A = zero_mtrx() for i in range(n): # главная диагональ A[i][i] = mxlist[i][i] print_mtrx(A) print() A = zero_mtrx() for i in range(n): # побочная диагональ A[i][n - 1 - i] = mxlist[i][n - 1 - i] print_mtrx(A) print() A = zero_mtrx() for i in range(n): # нижний треугольник под главной диагональю for j in range(i + 1): A[i][j] = mxlist[i][j] print_mtrx(A) print() for j in range(n): # рокировка 3-ей и 5-ой строк mxlist[2][j], mxlist[4][j] = mxlist[4][j], mxlist[2][j] print_mtrx(mxlist) </pre>	<pre> #include <iostream> using namespace std; const int N=10, M=10; void print(int a[N][M]) { for(int i=0; i<N; i++) { for (int j=0; j<M; j++) { cout.width(6); cout << a[i][j]; } cout << endl; } } int ZeroMatrix(int b[N][M]) { for(int i=0; i<N; i++) { for (int j=0; j<M; j++) { b[i][j]=0; } } return b[N][M]; } int main() { int Anya [N][M]; int k=1; for(int i=0; i<N; i++) // ввод матрицы 10x10 значений от 1 до 100 for(int j=0;j<M; j++) { Anya[i][j]=k; k++; } print(Anya); cout<<endl; int GDiag [N][M]; ZeroMatrix(GDiag); for(int i=0; i<N; i++) // главная диагональ { GDiag [i][i] = Anya[i][i]; } print(GDiag); cout<<endl; ZeroMatrix(GDiag); for(int i=0; i<N; i++) // побочная диагональ { GDiag [i][N-1-i] = Anya[i][N-1-i]; } } </pre>

	<pre> print(GDiag); cout<<endl; ZeroMatrix(GDiag); for(int i=0; i<N; i++) // нижний треугольник { for(int j=0; j<=i; j++) // под главной диагональю матрицы { GDiag [i][j] = Anya[i][j]; } } print(GDiag); cout<<endl; ZeroMatrix(GDiag); int c=0; for (int j=0; j<M; j++) // рокировка 3-ей и 5-ой строк { c=Anya[2][j]; Anya[2][j]=Anya[4][j]; Anya[4][j]=c; } print(Anya); return 0; } </pre>
--	---

Задания

I. Следом квадратной матрицы называется сумма элементов главной диагонали. Напишите программу, которая выводит след заданной квадратной матрицы. На вход программе подаётся натуральное число n – количество строк и столбцов в матрице, затем элементы матрицы (целые числа) построчно через пробел.

II. Напишите программу, которая выводит количество элементов квадратной матрицы в каждой строке, больших среднего арифметического элементов данной строки. На вход программе подаётся натуральное число n – количество строк и столбцов в матрице, затем элементы матрицы (целые числа) построчно через пробел. Программа должна вывести n чисел – для каждой строки количество элементов матрицы, больших среднего арифметического элементов данной строки.

III. Напишите программу, которая меняет местами столбцы в матрице. На вход программе на разных строках подаются два натуральных числа n и m – количество строк и столбцов в матрице, затем элементы матрицы построчно через пробел, затем числа i и j — номера столбцов, подлежащих обмену. Программа должна вывести указанную матрицу с замененными столбцами.

IV. На вход программе подаются два натуральных числа n и m . Напишите программу для создания матрицы размером $n \times m$, заполнив её

символами . и * в шахматном порядке. В левом верхнем углу должна стоять точка. Выведите полученную матрицу на экран, разделяя элементы пробелами.

V. Напишите программу для вычисления суммы двух матриц. На вход программе подаются два натуральных числа n и m – количество строк и столбцов в матрицах, затем элементы первой матрицы в виде таблицы, затем пустая строка, далее следуют элементы второй матрицы в виде таблицы. Программа должна вывести результирующую матрицу в виде таблицы, разделяя элементы символом пробела. Для вывода элементов матрицы отводите ровно 3 символа на каждый элемент. Для этого используйте строковый метод `ljust()`.

VI. Напишите программу для умножения матрицы на число. На вход программе подаются два натуральных числа n и m – количество строк и столбцов в матрицах, затем число, далее следуют элементы матрицы в виде таблицы. Программа должна вывести результирующую матрицу в виде таблицы, разделяя элементы символом пробела. Для вывода элементов матрицы отводите ровно 3 символа на каждый элемент. Для этого используйте строковый метод `ljust()`.

VII. На вход программе подаются два натуральных числа n и m . Напишите программу, которая создает матрицу размером $n \times m$, заполнив ее числами от 1 до $n \times m$ по столбцам, например (3×3):

```
1 4 7
2 5 8
3 6 9
```

Для вывода элементов матрицы, отводите ровно 3 символа на каждый элемент. Для этого используйте строковый метод `ljust()`.

ПРАКТИЧЕСКАЯ РАБОТА № 7.

Скалярное произведение векторов. Произведение вектора на матрицу. Умножение матрицы на матрицу

1) Скалярное произведение векторов

Python	C++
<pre>def print_mtx(a, u, v): [print(*[str(a[i][j]).ljust(5) for j in range(v)]) for i in range(u)] n, m = 3, 4 # ввод матрицы NxM значений от 1 до N*M по строкам A = [[i + j for j in range(0, m)] for i in range(1, n * m, m)] print_mtx(A, n, m) print() # ввод матрицы MxN значений от 1 до N*M по столбцам B = [[j + i for j in range(1, n * m, m)] for i in range(0, m)] print_mtx(B, m, n) print() C = [] for i in range(n): l = [] for j in range(n): g = 0 for k in range(m): g += A[i][k] * B[k][j] l.append(g) C.append(l) print_mtx(C, n, n)</pre>	<pre>#include <iostream> #include <vector> using namespace std; int main() { vector<int> a(2,1), b(2,3); a[1]=2; a.push_back(3);// a(1,2,3) b[1]=2; b.push_back(1);// b(3,2,1) int c=0; for (int i=0; i<3; i++) c+=a[i]*b[i]; cout<<"Scalar product of vectors a(1,2,3) and b(3,2,1) is "<<c<<endl; return 0; }</pre>

2) Произведение вектора на матрицу

Python	C++
<pre>def print_mtrx(a, u, v): [print(*[str(a[i][j]).ljust(5) for j in range(v)] for i in range(u)] n, m = 3, 4 # ввод матрицы NxM значений от 1 до N*M по строкам A = [[i + j for j in range(0, m)] for i in range(1, n * m, m)] print_mtrx(A, n, m) print() # ввод матрицы MxN значений от 1 до N*M по столбцам B = [[j + i for j in range(1, n * m, m)] for i in range(0, m)] print_mtrx(B, m, n) print() C = [] for i in range(n): l = [] for j in range(n): g = 0 for k in range(m): g += A[i][k] * B[k][j] l.append(g) C.append(l) print_mtrx(C, n, n)</pre>	<pre>#include <iostream> #include <vector> using namespace std; const int N=3; void printVec (vector <int> b) { for (int i=0; i<N; i++) { cout.width(4); cout<<b[i]; } cout<<endl<<endl; } void printMat (int ABC[N][N]) { for (int i=0; i<N; i++) { for (int j=0; j<N; j++) { cout.width(4); cout<<ABC[i][j]; } cout<<endl; } cout<<endl; } int main() { vector<int> a(2,1), c(3); a[1]=2; a.push_back(3);// a(1,2,3) printVec(a); int B[N][N]; int k=1; for (int i=0; i<N; i++) for (int j=0; j<N; j++) { B[i][j]=k; k++; } printMat(B); for (int i=0; i<N; i++) { c[i]=0; for (int j=0; j<N; j++) c[i]+=a[j]*B[j][i]; } printVec(c); return 0; }</pre>

3) Умножение матрицы на матрицу

Python	C++
<pre>def print_mtrx(a, u, v): [print(*[str(a[i][j]).ljust(5) for j in range(v)] for i in range(u)] n, m = 3, 4 # ввод матрицы NxM значений от 1 до N*M по строкам A = [[i + j for j in range(0, m)] for i in range(1, n * m, m)] print_mtrx(A, n, m) print() # ввод матрицы MxN значений от 1 до N*M по столбцам B = [[j + i for j in range(1, n * m, m)] for i in range(0, m)] print_mtrx(B, m, n) print() C = [] for i in range(n): l = [] for j in range(n): g = 0 for k in range(m): g += A[i][k] * B[k][j] l.append(g) C.append(l) print_mtrx(C, n, n)</pre>	<pre>#include<iostream> using namespace std; const int N=3, M=5; template <int row, int col> void print (int (&ABC)[row][col], int u, int v) { for (int i=0; i<u; i++) { for (int j=0; j<v; j++) { cout.width(4); cout<<ABC[i][j]; } cout<<endl; } cout<<endl; } int main() { int A[N][M], B[M][N], C[N][N]; int k=1; for (int i=0; i<N; i++) for (int j=0; j<M; j++) { A[i][j]=k; k++; } print(A, N, M); k=1; for (int j=0; j<N; j++) { for (int i=0; i<M; i++) { B[i][j]=k; k+=1; } } print(B, M, N); for (int i=0; i<N; i++) for (int j=0; j<N; j++) { C[i][j]=0; for (int m=0; m<M; m++) C[i][j]+=A[i][m]*B[m][j]; } print(C, N, N); return 0; }</pre>

Задания

I.

Произведения вектора $B(1,2,3)$ (слева) на матрицу

$$\begin{array}{cc} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{array}$$

и произведение матрицы $\begin{array}{ccc} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{array}$ на тот же вектор (справа). Координаты

вектора и матрицу пользователь вводит с экрана.

II. Напишите программу для нахождения результирующей матрицы

при умножении $\begin{array}{cc} 4 & 2 \\ 9 & 0 \end{array}$ на $\begin{array}{cc} 3 & 1 \\ -3 & 4 \end{array}$

III. Напишите программу для нахождения результирующей матрицы

при умножении $\begin{array}{cc} 2 & 1 \\ -3 & 0 \\ 4 & -1 \end{array}$ на $\begin{array}{ccc} 5 & -1 & 6 \\ -3 & 0 & 7 \end{array}$

V. Напишите программу для вычисления произведения

$$(1 \ 2 \ 3) \times \begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix} \times \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

Какой размерности получится матрица? (231)

ПРАКТИЧЕСКАЯ РАБОТА № 8.

Численное интегрирование (методом прямоугольников, методом трапеций, метод Симпсона)

Обзор методов интегрирования.

1. Методы Ньютона-Котеса. Сюда относятся метод прямоугольников, трапеций, Симпсона.
2. Методы статистических испытаний (методы Монте-Карло). Здесь узлы сетки для квадратурного или кубатурного интегрирования выбираются с помощью датчика случайных чисел, ответ носит вероятностный характер. В основном применяются для вычисления кратных интегралов.
3. Сплайновые методы. Задается кусочный полином с условиями связи между отдельными полиномами посредством системы коэффициентов.
4. Методы наивысшей алгебраической точности. Обеспечивают оптимальную расстановку узлов сетки интегрирования и выбор весовых коэффициентов $w(x)$. Сюда относится метод Гаусса, Гаусса-Кристоффеля (вычисление несобственных интегралов) и метод Маркова.

1. Методы Ньютона-Котеса.

1) Метод трапеций

```
from math import pi, sin
def f(z):
    return sin(z)
n = 574
a, b = 0, pi
h = (b - a) / n
s = 0
for i in range(0, n):
    x1 = a + i * h
    x2 = a + (i + 1) * h

    s += (f(x1) + f(x2)) / 2

print(s*h)
```

```
#include<math.h>
using namespace std;
double f(double z)
{
    return sin(z);
}

int main()
{
    int n = 574;
    double a = 0, b = M_PI, h=(b-a)/n, x1,
x2, s=0;
    for (int i=0; i<n; i++)
    {
        x1 = a + i*h;
        x2 = a + (i+1)*h;

        s += (f(x1) + f(x2))/2;

    }

    cout<<s*h;

    return 0;
}
```

2) Метод прямоугольников

<pre>from math import pi, sin def f(z): return sin(z) n = 406 a, b = 0, pi h = (b - a) / n s = 0 for i in range(0, n): s += f(a + (i + 0.5) * h) print(s * h)</pre>	<pre>#include<iostream> #include<math.h> using namespace std; double f(double z) { return sin(z); } int main() { int n = 406; double a = 0, b = M_PI, h=(b-a)/n, s=0; for (int i=0; i<n; i++) { s += f(a+(i+0.5)*h); } cout<<s*h; return 0; }</pre>
---	--

3) Метод Симпсона

<pre>from math import pi, sin def f(z): return sin(z) n = 11 a, b = 0, pi h = (b - a) / n s = 0 for i in range(0, n): x1 = a + i * h x2 = a + (i + 1) * h s += 1 / 6.0 * (f(x1) + 4 * f(0.5 * (x1 + x2)) + f(x2)) print(s*h)</pre>	<pre>#include<iostream> #include<math.h> using namespace std; double f(double z) { return sin(z); } int main() { int n = 11; double a = 0, b = M_PI, h=(b-a)/n, x1, x2, s=0; for (int i=0; i<n; i++) { x1 = a + i*h; x2 = a + (i+1)*h; s += 1/6.0*(f(x1) + 4.0*f(0.5*(x1+x2)) + f(x2)); } }</pre>
---	--

	<pre>cout<<s*h; return 0; }</pre>
--	---

Задания

I. Вычислите трижды разными методами (методом трапеций, прямоугольников и методом Симпсона) определенный интеграл:

$$\int_{-1}^1 \frac{x dx}{\sqrt{5-4x}}$$

Ответ, полученный при аналитическом расчете: 1,6666666666666666666666666666667.

II. Вычислите трижды разными методами (методом трапеций, прямоугольников и методом Симпсона) определенный интеграл:

$$\int_0^{0,75} \frac{dx}{(x+1)\sqrt{x^2+1}}$$

Ответ, полученный при аналитическом расчете: 0,52255045738047987978609863603808.

III. Вычислите трижды разными методами (методом трапеций, прямоугольников и методом Симпсона) определенный интеграл:

$$\int_0^1 \frac{\arcsin(\sqrt{x})}{\sqrt{x(1-x)}} dx$$

Ответ, полученный при аналитическом расчете: 2,467401100272339654708622749969.

IV. Вычислите трижды разными методами (методом трапеций, прямоугольников и методом Симпсона) определенный интеграл:

$$\int_{\frac{1}{2}}^2 \left(1+x-\frac{1}{x}\right) e^{x+1/x} dx$$

Ответ, полученный при аналитическом расчете: 18,273740941055210157105263926752.

V. Вычислите трижды разными методами (методом трапеций, прямоугольников и методом Симпсона) определенный интеграл:

$$\int_0^\pi \frac{x \sin(x)}{1+\cos^2(x)} dx$$

Ответ, полученный при аналитическом расчете: 2,467401100272339654708622749969.

ПРАКТИЧЕСКАЯ РАБОТА № 9.

Метод Гаусса (численное интегрирование в языках Python и C++) при заданных гауссовых точках и коэффициентах

4. Методы наивысшей алгебраической точности

1) метод Гаусса при заданных точках, коэффициентах и границах интегрирования от -1 до 1

Python	C++
<pre> from math import * def f(z): return z**2 n = int(input('Задайте число точек Гаусса для интегрирования n (1<n<6) ')) if n == 2: x=[-1 / sqrt(3), 1 / sqrt(3)] w=[1, 1] elif n == 3: x = [-sqrt(3 / 5), 0, sqrt(3 / 5)] w = [5 / 9, 8 / 9, 5 / 9] elif n == 4: x = [-sqrt(3 / 7 + 2 / 7 * sqrt(6 / 5)), -sqrt(3 / 7 - 2 / 7 * sqrt(6 / 5)), sqrt(3 / 7 - 2 / 7 * sqrt(6 / 5)), sqrt(3 / 7 + 2 / 7 * sqrt(6 / 5)),] w = [(18 - sqrt(30)) / 36, (18 + sqrt(30)) / 36, (18 + sqrt(30)) / 36, (18 - sqrt(30)) / 36] elif n == 5: x = [-1 / 3 * sqrt(5 + 2 * sqrt(10 / 7)), -1 / 3 * sqrt(5 - 2 * sqrt(10 / 7)), 0, 1 / 3 * sqrt(5 - 2 * sqrt(10 / 7)), 1 / 3 * sqrt(5 + 2 * sqrt(10 / 7))] w = [(322 - 13 * sqrt(70)) / 900, (322 + 13 * sqrt(70)) / 900, 128 / 225, (322 + 13 * sqrt(70)) / 900, (322 - 13 * sqrt(70)) / 900] else: print('Ошибка ввода: n должно быть 1<n<6.', 'Попробуйте еще раз', sep='\n', end='\n') print(x) print(w) s = 0 for i in range(n): s += w[i] * f(x[i]) print(f'Интеграл на отрезке [-1, 1] равен {s}') </pre>	<pre> #include<iostream> #include<math.h> using namespace std; double f(double z) { return pow(z, 2); } void print_vec(double v[], int n) { for (int i=0; i<n; i++) { cout<< v[i]<<" "; } cout<<endl; } main() { setlocale(0, ""); int n; cout<<"Задайте число точек Гаусса для интегрирования n (1<n<6) "<<endl; cin>>n; double x[n], w[n]; switch (n) { case 2: x[0]=-1. / sqrt(3); x[1] = 1. / sqrt(3); w[0] = 1.; w[1] = 1.; break; case 3: x[0]=-sqrt(3. / 5); x[1] = 0; x[2] = sqrt(3. / 5); w[0] = 5. / 9; w[1] = 8. / 9; w[2] = 5. / 9; break; case 4: x[0] = -sqrt(3. / 7 + 2. / 7 * sqrt(6. / 5)); x[1] = -sqrt(3. / 7 - 2. / 7 * sqrt(6. / 5)); x[2] = sqrt(3. / 7 - 2. / 7 * sqrt(6. / 5)); x[3] = sqrt(3. / 7 + 2. / 7 * sqrt(6. / 5)); w[0] = (18 - sqrt(30)) / 36.; w[1] = (18 + sqrt(30)) / 36.; w[2] = (18 + sqrt(30)) / 36.; </pre>

	<pre> w[3] = (18 - sqrt(30)) / 36.; break; case 5: x[0] = -1 / 3. * sqrt(5 + 2 * sqrt(10 / 7.)); x[1] = -1 / 3. * sqrt(5 - 2 * sqrt(10 / 7.)); x[2] = 0.; x[3] = 1 / 3. * sqrt(5 - 2 * sqrt(10 / 7.)); x[4] = 1 / 3. * sqrt(5 + 2 * sqrt(10 / 7.)); w[0] = (322 - 13 * sqrt(70)) / 900.; w[1] = (322 + 13 * sqrt(70)) / 900.; w[2] = 128. / 225; w[3] = (322 + 13 * sqrt(70)) / 900.; w[4] = (322 - 13 * sqrt(70)) / 900.; break; default: cout<<"Ошибка ввода: n должно быть 1<n<6. Попробуйте еще раз."<<endl; break; } print_vec(x, n); print_vec(w,n); double s = 0; for (int i=0; i<n; i++) { s += w[i] * f(x[i]); } cout<<"Интеграл на отрезке [-1, 1] равен "<<s; return 0; } </pre>
--	---

2) метод Гаусса при заданных точках, коэффициентах и произвольных границах интегрирования от а до b

Python	C++
<pre> def f(z): return sin(z) n = int(input('Задайте число точек Гаусса для интегрирования n (1<n<6) ')) if n == 2: x=[-1 / sqrt(3), 1 / sqrt(3)] w=[1, 1] elif n == 3: x = [-sqrt(3 / 5), 0, sqrt(3 / 5)] w = [5 / 9, 8 / 9, 5 / 9] elif n == 4: x = [-sqrt(3 / 7 + 2 / 7 * sqrt(6 / 5)), -sqrt(3 / 7 - 2 / 7 * sqrt(6 / 5)), sqrt(3 / 7 - 2 / 7 * sqrt(6 / 5)), sqrt(3 / 7 + 2 / 7 * sqrt(6 / 5)),] w = [(18 - sqrt(30)) / 36, (18 + sqrt(30)) / 36, (18 + sqrt(30)) / 36, (18 - sqrt(30)) / 36] elif n == 5: x = [-1 / 3 * sqrt(5 + 2 * sqrt(10 / 7)), -1 / 3 * sqrt(5 - 2 * sqrt(10 / 7)), 0, 1 / 3 * sqrt(5 - 2 * sqrt(10 / 7)), 1 / 3 * sqrt(5 + 2 * sqrt(10 / 7))] w = [(322 - 13 * sqrt(70)) / 900, (322 + 13 * </pre>	<pre> #include<iostream> #include<math.h> using namespace std; double f(double z) { return sin(z); } void print_vec(double v[], int n) { for (int i=0; i<n; i++) { cout<< v[i]<<" "; } cout<<endl; } main() { setlocale(0, ""); int n; double a = 0, b = M_PI; cout<<"Задайте число точек Гаусса для интегрирования n (1<n<6) "<<endl; cin>>n; double x[n], w[n]; </pre>

<pre> sqrt(70)) / 900, 128 / 225, (322 + 13 * sqrt(70)) / 900, (322 - 13 * sqrt(70)) / 900] else: print('Ошибка ввода: n должно быть 1<n<6. Попробуйте еще раз') a, b = 0, pi # границы интегрирования print(x) print(w) s=0 for i in range(n): s += w[i] * f((b - a) / 2 * x[i] + (a + b) / 2) print(f'Интеграл на отрезке [{a}, {b}] равен {(b- a)/2*s}') </pre>	<pre> switch (n) { case 2: x[0] = -1. / sqrt(3); x[1] = 1. / sqrt(3); w[0] = 1.; w[1] = 1.; break; case 3: x[0] = -sqrt(3. / 5); x[1] = 0; x[2] = sqrt(3. / 5); w[0] = 5. / 9; w[1] = 8. / 9; w[2] = 5. / 9; break; case 4: x[0] = -sqrt(3. / 7 + 2. / 7 * sqrt(6. / 5)); x[1] = -sqrt(3. / 7 - 2. / 7 * sqrt(6. / 5)); x[2] = sqrt(3. / 7 - 2. / 7 * sqrt(6. / 5)); x[3] = sqrt(3. / 7 + 2. / 7 * sqrt(6. / 5)); w[0] = (18 - sqrt(30)) / 36.; w[1] = (18 + sqrt(30)) / 36.; w[2] = (18 + sqrt(30)) / 36.; w[3] = (18 - sqrt(30)) / 36.; break; case 5: x[0] = -1 / 3. * sqrt(5 + 2 * sqrt(10 / 7.)); x[1] = -1 / 3. * sqrt(5 - 2 * sqrt(10 / 7.)); x[2] = 0.; x[3] = 1 / 3. * sqrt(5 - 2 * sqrt(10 / 7.)); x[4] = 1 / 3. * sqrt(5 + 2 * sqrt(10 / 7.)); w[0] = (322 - 13 * sqrt(70)) / 900.; w[1] = (322 + 13 * sqrt(70)) / 900.; w[2] = 128. / 225; w[3] = (322 + 13 * sqrt(70)) / 900.; w[4] = (322 - 13 * sqrt(70)) / 900.; break; default: cout<<"Ошибка ввода: n должно быть 1<n<6. Попробуйте еще раз."<<endl; break; } print_vec(x, n); print_vec(w,n); double s = 0; for (int i=0; i<n; i++) { s += w[i] * f((b - a) / 2 * x[i] + (a + b) / 2); } cout<<"Интеграл на отрезке ["<<a<< ", "<<b<<"] равен "<<(b-a)/2*s; return 0; } </pre>
--	---

Задания

I. Вычислите определенный интеграл методом Гаусса, используя 2, 3, 4 и 5 точек Гаусса:

$$\int_0^3 \arcsin \left(\sqrt{\frac{x}{1+x}} \right) dx$$

Ответ, полученный при аналитическом расчете:
2,4567393972175136910894115028668.

II. Вычислите определенный интеграл методом Гаусса, используя 2, 3, 4 и 5 точек Гаусса:

$$\int_0^{2\pi} \frac{dx}{\sin^4(x) + \cos^4(x)}$$

Ответ, полученный при аналитическом расчете:
8,8857658763167324940317619801214.

III. Вычислите определенный интеграл методом Гаусса, используя 2, 3, 4 и 5 точек Гаусса:

$$\int_0^{\pi} e^x \cos^2(x) dx$$

Ответ, полученный при аналитическом расчете:
13,284415579667561403437451820769

IV. Вычислите определенный интеграл методом Гаусса, используя 2, 3, 4 и 5 точек Гаусса:

$$\int_0^1 x(2-x^2)^{12} dx$$

Ответ, полученный при аналитическом расчете:
12,115384615384615384615384615385

V. Вычислите определенный интеграл методом Гаусса, используя 2, 3, 4 и 5 точек Гаусса:

$$\int_0^{2\pi} \frac{dx}{(2 + \cos(x))(3 + \cos(x))}$$

Ответ, полученный при аналитическом расчете:
1,406157259389252577680216020254

ПРАКТИЧЕСКАЯ РАБОТА № 10.

Метод Гаусса (численное интегрирование в языках Python и C++) при рассчитанных гауссовых точках и коэффициентах

4. Методы наивысшей алгебраической точности

1) метод Гаусса при рассчитанных точках, коэффициентах и заданных границах интегрирования от -1 до 1

Python	C++
<pre> from math import * def f(z): return z**2 def initial_x(n): x = [] for i in range(n): x.append(cos(pi * (4 * i + 3) / (4 * n + 2))) return x def PLx(x, n): derP = [] for i in range(n): Px = [1, x[i]] for j in range(2, n + 1): Px.append((2 * j - 1) / j * x[i] * Px[j - 1] - (j - 1) / j * Px[j - 2]) derP.append(n / (1 - x[i]**2) * (Px[n - 1] - x[i] * Px[n])) x[i] = x[i] - Px[n] / derP[i] return derP, x def wi(n, x, derP): wi = [] for i in range(n): wi.append(2 / ((1 - x[i]**2) * derP[i]**2)) return wi n = int(input('Задайте число точек Гаусса для интегрирования ')) k = int(input('Количество итераций для вычисления гауссовых точек ')) x = initial_x(n) for i in range(k): derP, x = PLx(x, n) </pre>	<pre> #include<iostream> #include<math.h> #include<vector> using namespace std; void print_vec(vector<double> v, int n) { for (int i=0; i<n; i++) { cout<< v[i]<<" "; } cout<<endl; } double f(double z) { return pow(z,2); } vector<double> initial_x(int n) { vector <double> x(n); for (int i=0; i<n; i++) { x[i]=cos(M_PI*(4*(i+1)-1)/(4*n+2)); } return x; } void PLx (int n, int k, vector<double> &derP, vector<double> &x) { vector<double> Px(n+1); x=x0; for (int m=0; m<k; m++) { for (int i=0; i<n; i++) { Px[0] = 1; Px[1] = x[i]; for (int j=2; j<n+1; j++) { Px[j]=(2.*j-1)/j*x[i]*Px[j-1]-(j- 1.)/j*Px[j-2]; } derP[i]=n/(1.-x[i]*x[i])*(Px[n-1]-x[i]*Px[n]); } } } </pre>

```

print(x)
w = wi(n, x, derP)
print(w)
s=0
for i in range(n):
    s += w[i] * f(x[i])

print(f'Интеграл на отрезке [-1, 1] равен {s}')

```

```

x[i] = x[i] - Px[n]/derP[i];
    }
}
vector<double> wi(vector<double> x, vector<double>
derP, int n)
{
    vector<double> w(n);

    for (int i=0; i<n; i++)
    {
        w[i]= 2./((1-x[i]*x[i])*pow(derP[i],2));
    }
    return w;
}

int main()
{
    setlocale(0,"");
    int n;
    cout<<"Задайте число точек Гаусса для
интегрирования n (1<n<6)"<< endl;
    cin>>n;

    int k;
    cout<<"Количество итераций для вычисления
гауссовых точек "<< endl;
    cin>>k;

    vector<double> x=initial_x(n);
    vector<double> derP(n);
    PLx (n, k, derP, x);

    vector<double> w = wi(x, derP, n);
    print_vec(x, n);
    print_vec(w, n);

    double s=0;
    for (int i=0; i<n; i++)
    {
        s += w[i]*f(x[i]);
    }
    cout<<"Интеграл на отрезке [-1, 1] равен "
<<s<<endl;
    return 0;
}

```

2) метод Гаусса при рассчитанных точках, коэффициентах и произвольных границах интегрирования от a до b

Python	C++
<pre> from math import * def f(z): return sin(z) def initial_x(n): x = [] for i in range(n): x.append(cos(pi * (4 * i + 3) / (4 * n + 2))) return x def PLx(x, n): derP = [] for i in range(n): Px = [1, x[i]] for j in range(2, n + 1): Px.append((2 * j - 1) / j * x[i] * Px[j - 1] - (j - 1) / j * Px[j - 2]) derP.append(n / (1 - x[i]**2) * (Px[n - 1] - x[i] * Px[n])) x[i] = x[i] - Px[n] / derP[i] return derP, x def wi(n, x, derP): wi = [] for i in range(n): wi.append(2 / ((1 - x[i]**2) * derP[i]**2)) return wi n = int(input('Задайте число точек Гаусса для интегрирования ')) k = int(input('Количество итераций для вычисления гауссовых точек ')) x = initial_x(n) a, b = 0, pi # границы интегрирования for i in range(k): derP, x = PLx(x, n) print(x) w = wi(n, x, derP) print(w) s=0 for i in range(n): s += w[i] * f((b - a) / 2 * x[i] + (a + b) / 2) print(f'Интеграл на отрезке [{a}, {b}] равен {(b- a)/2*s}') </pre>	<pre> #include<iostream> #include<math.h> #include<vector> using namespace std; void print_vec(vector<double> v, int n) { for (int i=0; i<n; i++) { cout<< v[i]<<" "; } cout<<endl; } double f(double z) { return sin(z); } vector<double> initial_x(int n) { vector <double> x(n); for (int i=0; i<n; i++) { x[i]=cos(M_PI*(4*(i+1)-1)/(4*n+2)); } return x; } void PLx (int n, int k, vector<double> &derP, vector<double> &x) { vector<double> Px(n+1); for (int m=0; m<k; m++) { for (int i=0; i<n; i++) { Px[0] = 1; Px[1] = x[i]; for (int j=2; j<n+1; j++) { Px[j]=(2.*j-1)/j*x[i]*Px[j-1]-(j- 1.)/j*Px[j-2]; } derP[i]=n/(1.-x[i]*x[i])*(Px[n-1]-x[i]*Px[n]); x[i] = x[i] - Px[n]/derP[i]; } } } vector<double> wi(vector<double> x, vector<double> derP, int n) { vector<double> w(n); for (int i=0; i<n; i++) </pre>

	<pre> { w[i]= 2./((1-x[i]*x[i])*pow(derP[i],2)); } return w; } int main() { setlocale(0,""); int n; cout<<"Задайте число точек Гаусса для интегрирования n (1<n<6)"<< endl; cin>>n; int k; cout<<"Количество итераций для вычисления гауссовых точек "<< endl; cin>>k; vector<double> x=initial_x(n); vector<double> derP(n); PLx (n, k, derP, x); vector<double> w = wi(x, derP, n); print_vec(x, n); print_vec(w, n); double a = 0, b = M_PI; double s=0; for (int i=0; i<n; i++) { s += w[i]* f((b - a) / 2 * x[i] + (a + b) / 2); } cout<<"Интеграл на отрезке [-1, 1] равен " <<(b-a)/2*s<<endl; return 0; } </pre>
--	--

Задания

I. Вычислите определенный интеграл методом Гаусса, используя 2, 6 и 7 точек Гаусса:

$$\int_{-1}^1 \frac{xdx}{\sqrt{5-4x}}$$

Ответ, полученный при аналитическом расчете: 1,66666666666666666666666666666667.

II. Вычислите определенный интеграл методом Гаусса, используя 3, 7 и 9 точек Гаусса:

$$\int_0^{0,75} \frac{dx}{(x+1)\sqrt{x^2+1}}$$

Ответ, полученный при аналитическом расчете:
0,52255045738047987978609863603808.

III. Вычислите определенный интеграл методом Гаусса, используя 4, 7 и 9 точек Гаусса:

$$\int_0^1 \frac{\arcsin(\sqrt{x})}{\sqrt{x(1-x)}} dx$$

Ответ, полученный при аналитическом расчете:
2,467401100272339654708622749969.

IV. Вычислите определенный интеграл методом Гаусса, используя 4, 7 и 9 точек Гаусса:

$$\int_{\frac{1}{2}}^2 \left(1 + x - \frac{1}{x}\right) e^{x+1/x} dx$$

Ответ, полученный при аналитическом расчете:
18,273740941055210157105263926752.

V. Вычислите определенный интеграл методом Гаусса, используя 2, 7 и 8 точек Гаусса:

$$\int_0^\pi \frac{x \sin(x)}{1 + \cos^2(x)} dx$$

Ответ, полученный при аналитическом расчете:
2,467401100272339654708622749969.

Список используемых источников

1. К.Ю. Поляков, Е.А. Еремин С и С++ [Электронный ресурс]. – Режим доступа: https://kpolyakov.spb.ru/download/ch10-8_c_cpp.pdf
2. К.Ю. Поляков, Е.А. Еремин Python [Электронный ресурс]. – Режим доступа: https://kpolyakov.spb.ru/download/ch10-8_python.pdf
3. "Поколение Python": курс для начинающих [Электронный ресурс]. – Режим доступа: <https://stepik.org/course/58852/syllabus>
4. Введение в программирование (C++) [Электронный ресурс]. – Режим доступа: <https://stepik.org/course/363/syllabus>
5. The Finite Element Method for Problems in Physics [Электронный ресурс]. – Режим доступа: <https://www.coursera.org/learn/finite-element-method?>
6. Gaussian quadrature [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Gaussian_quadrature (Дата обращения: 19.11.2021).
7. Gauss–Legendre quadrature [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Gauss–Legendre_quadrature.
8. Многочлены Лежандра [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Многочлены_Лежандра.
9. Python [Электронный ресурс]. – Режим доступа: <https://www.python.org/>
10. Wing Python IDE [Электронный ресурс]. – Режим доступа: <https://wingware.com/>
11. Зубарева, М.О. Gauss-method [Электронный ресурс]. – Режим доступа: <https://github.com/ZubarevaMari/Gauss-method>.

1. Указатели

Указатели для переменных

```
#include <iostream>
int main()
{
    int a, *b;
    a=4;
    b=&a;
    std::cout << a << std::endl;
    std::cout << *b << std::endl;
    a=5;
    std::cout << *b << std::endl;
    return 0;
}
```

Указатели для вектора

```
#include <iostream>
#include <vector>
int main()
{
    std::vector <double> vec1 (2, 3.1), *vec2;
    vec2=&vec1;
    std::cout << vec1.size() << " " << vec2->size() << std::endl;
    vec1.resize(3, 2.1);
    std::cout << vec1.size() << " " << vec2->size() << std::endl;
    return 0;
}
```

2. Итератор

```
#include <iostream>
#include <vector>
int main()
{
    std::vector <double> vec(2, 3.1);
    for (std::vector <double> :: iterator it=vec.begin(); it!=vec.end(); it++)
    {
        std::cout << *it << std::endl;
    }
}
```

```
    return 0;
}
```

Указатели для векторов и итератор

```
#include <iostream>
#include <vector>
int main()
{
    std::vector<double> phys (2,3.1), *ukaz;
    ukaz=&phys;
    std::cout << phys.size() << " " << ukaz -> size() << std::endl;
    //phys.resize(3,2.1);
    //std::cout << phys.size() << " " << ukaz -> size() << std::endl;
    for (std::vector<double> :: iterator it=phys.begin(); it !=phys.end(); it++)
    {
        std::cout << *it << std::endl;
    }
    return 0;
}
```

3. Классы

Triangle

```
#include<iostream>

class Triangle{
public:
    Triangle (double a, double b);
    double area();
    double base;
private:
    double height;
};
Triangle:: Triangle (double a, double b){
    base=a;
    height=b;
};
double Triangle::area(){
    return 0.5*base*height;
}

int main()
{
    Triangle tri1 (1.,5.), tri2(3.,2.);
```

```

std::cout<<tri1.area()<<"  "<<tri2.area()<<std::endl;
tri1. base=3.;
std::cout<<tri1.area()<<"  "<<tri2.area()<<std::endl;
return 0;
}

```

Kvadrat

```

#include<iostream>
class kvadrat{
    public:
        kvadrat (double a);
        double area();
        double perimeter();
        double leingth;
};
kvadrat:: kvadrat (double a){
    leingth=a;
}
double kvadrat::area(){
    return leingth*leingth;
}
double kvadrat::perimeter(){
    return leingth*4;
}
int main()
{
kvadrat kva1(1.), kva2(5.);
std::cout<< kva1.area()<<"  "<< kva2.area()<<std::endl;
kva1.leingth=3.;
std::cout<< kva1.area()<<"  "<< kva2.area()<<std::endl;
return 0;
}

```

4 .h файл

Triangle.h (должен находиться в той же папке, где расположен файл основной программы)

```

class Triangle{
    public:
        Triangle (double a, double b);
        double area();
        double base;
    private:

```

```
        double height;
};
Triangle:: Triangle (double a, double b){
    base=a;
    height=b;
}
double Triangle::area(){
    return 0.5*base*height;
}
```

Trian (файл основной программы)

```
#include<iostream>
#include"Triangle.h"

int main()
{
Triangle tri1 (1.,5.), tri2(3.,2.);
std::cout<<tri1.area()<<"  "<<tri2.area()<<std::endl;
tri1. base=3.;
std::cout<<tri1.area()<<"  "<<tri2.area()<<std::endl;
return 0;
}
```