

**Московский государственный технический
университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Основы программирования»

Отчет по лабораторной работе №6
«Численное интегрирование функций»

Выполнил:
студент группы ИУ5-15Б
Трифонов Дмитрий
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Папшев И.С.
Подпись и дата:

Москва, 2021 г.

Постановка задачи

1. Численное интегрирование функции с заданной точностью методом прямоугольников.

$$\int_a^b f(x) dx$$

Вычислить определённый интеграл в пределах от **a** до **b** для четырех функций $f_1 = x$, $f_2 = \sin(22 * x)$, $f_3 = x^4$ и $f_4 = \arctg(x)$.

Вычисление интеграла оформить в виде функции `IntRect`.

Вычисления выполнить для пяти значений точности: 0.01, 0.001, 0.0001, 0.00001 и 0.000001.

Исследовать быстродействие алгоритма в зависимости от подынтегральной функции и требуемой точности (быстродействие алгоритма можно оценить числом элементарных прямоугольников **n**, при котором достигнута требуемая точность интегрирования).

Результаты представить в виде 5 таблиц, по одной таблице для каждого значения точности. В каждой таблице выводить данные для всех четырех функций.

Для печати таблицы результатов использовать приведенную в приложении 2 функцию

```
void PrintTabl(I_print i_prn[],int k)
```

Здесь `i_prn[]` – массив структур типа `I_print` размерностью `k`.

Для ее использования нужно создать и заполнить массив из структур типа `I_print` и передать его в функцию в качестве параметра.

Вид таблицы приведен в Приложении 1.

Для использования в функции `PrintTabl()` символов псевдографики замените в ней кодировочную таблицу Windows 1251 на кодировочную таблицу MS Dos 866 (так как это глобальная переменная, то перед выходом из функции нужно восстановить кодировочную таблицу Windows 1251).

2. Выполнить п.1, используя для интегрирования метод трапеций. Вычисление интеграла оформить в виде функции `IntTrap`.

Для печати таблиц результатов использовать ту же функцию, что и в методе прямоугольников.

Разработка алгоритма

Вывод результатов на экран производится функцией `printIntResults`, которая принимает на вход функцию, по которой будет искомый интеграл (метод прямоугольников или метод трапеций). Эта функция заполняет соответствующими значениями массив результатов `I_print`. Сам же вывод результатов потом совершается через функцию `PrintTabl`.

Входные данные:

Границы вычислений — `int a, b`

Начальная точность — `int n`

`double eps`

`double (*searchType) (TPF, int)` — указатель на функцию для поиска интеграла (метод прямоугольников / трапеций)

Промежуточные данные:

`int startingN` – переменная, в которой сохраняется значение `N`, установленное в начале выполнения команды.

`char* funcNames[]` - названия функций для вывода в таблицу

`TPF funcs[]` - массив функций, интеграл которых требуется вычислить

`I_print results[5]` - массив для заполнения полученными результатами для последующего их вывода на экран (функция `void printIntResults(double (*searchType) (TPF, int))`)

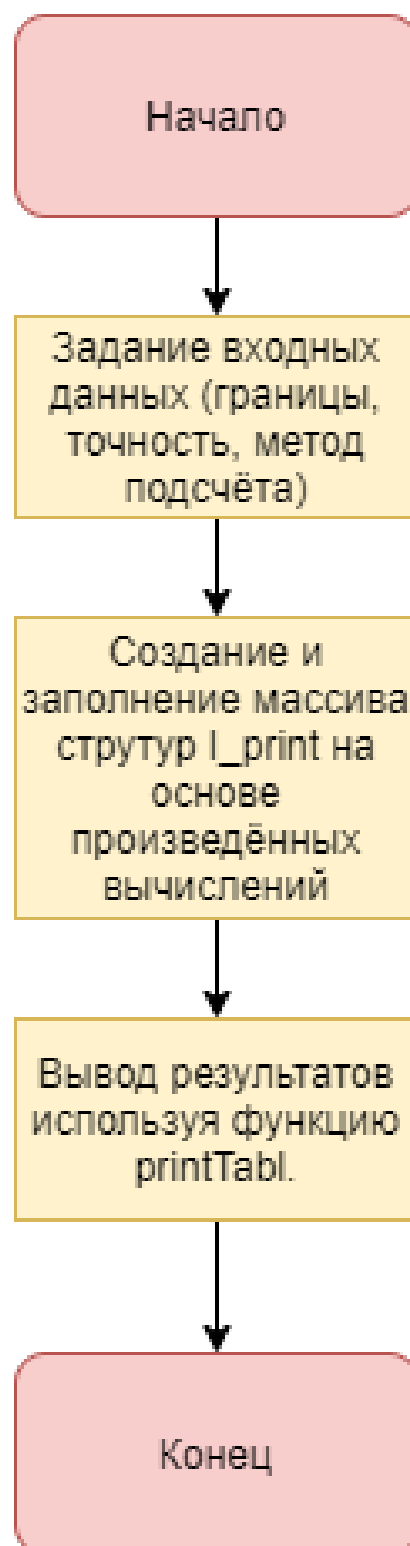
А также данные для методов прямоугольника и трапеции `double dy, y, ln` – для трапеций и `double s, dx` для прямоугольников.

Выходные данные:

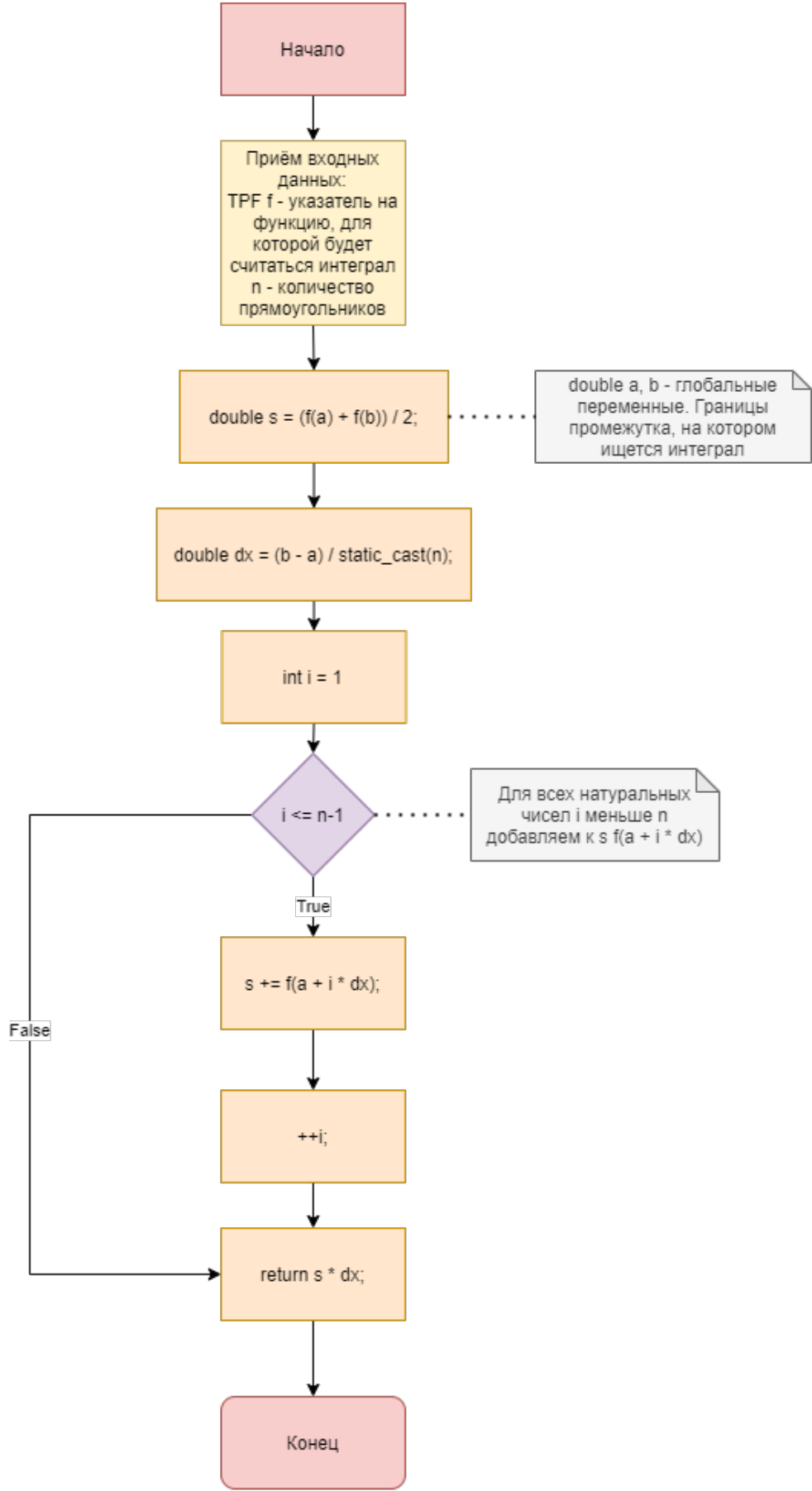
`double intRect(TPF f, int n)` и `double intTrap(TPF f, int n)` возвращают значение интеграла для заданной функции и шага `n`.

«Подбор шага» для нужной точности обеспечивает `double getInt(TPF f, double (*searchType) (TPF, int))`, которая возвращает итоговое значение интеграла для заданных функций.

Общая схема программы:



Уточнённая схема функции IntRect:



Текст программы

...riy\Documents\study\programming basics\Lab6\Lab6\Lab6.cpp

1

```
1 #include <iostream>
2 #include <string>
3 #include <iomanip>
4 #include <conio.h>
5 #include <cmath>
6 using namespace std;
7
8 typedef double (*TPF)(double);
9 struct I_print { //данные для печати результатов интегрирования
10     char* name; //название функции
11     double i_sum; //значение интегральной суммы
12     double i_toch; //точное значение интеграла
13     int n; //число разбиений области интегрирования
14     //при котором достигнута требуемая точность
15 };
16 double f1(double x) { return x; }
17 double f2(double x) { return sin(22 * x); };
18 double f3(double x) { return x * x * x * x; };
19 double f4(double x) { return atan(x); };
20 double f5(double x) { return cos(x); }
21
22 double eps = 0.1, a = -1, b = 3;
23 int n = 2, startingN = 2;
24 char* funcNames[] = { (char*)"x", (char*)"sin(22x)", (char*)"x^4",
25     (char*)"atan(x)", (char*)"cos(x)"};
26 TPF funcs[] = { f1, f2, f3, f4, f5 };
27
28 double getRightInt(TPF f) {
29     if (f == f1)
30         return (b * b - a * a) / 2.0;
31     else if (f == f2)
32         return (cos(a * 22.0) - cos(b * 22.0)) / 22.0;
33     else if (f == f3)
34         return (b * b * b * b * b - a * a * a * a) / 5.0;
35     else if (f == f4)
36         return b * atan(b) - a * atan(a) - (log(b * b + 1) - log(a * a + 1)) /
37         2.0;
38     else if (f == f5)
39         return sin(b) - sin(a);
40 }
41
42 double intRect(TPF f, int n) {
43     double s = s + (f(a) + f(b)) / 2;
44     double dx = (b - a) / static_cast<double>(n);
45     for (int i = 1; i <= n - 1; ++i) {
46         s += f(a + i * dx);
47     }
48     return s * dx;
49 }
50
51 double intTrap(TPF f, int n) {
```

```
48     double dy = (b - a) / n;
49     double y = f(a) + f(b);
50     for (int i = 1; i < n; i++) { y += 2 * (f(a + dy * i)); }
51     double In = ((b - a) / (2 * n) * y);
52     return In;
53 }
54 double getInt(TPF f, double (*searchType) (TPF, int)) {
55     while (abs(searchType(f, n * 2) - searchType(f, n)) / 3.0 > eps)
56         n *= 2;
57     return searchType(f, n * 2);
58 }
59
60 void PrintTabl(I_print i_prn[], int k)
61 {
62     system("chcp 866>nul");
63     const int m = 4; //число столбцов таблицы
64     int wn[m] = { 12, 18, 18, 10 }; //ширина столбцов таблицы
65     char* title[m] = { (char*)"Function", (char*)"Integral", (char*)"IntSum",
66         (char*)"N " };
67     int size[m];
68     for (int i = 0; i < m; i++)
69         size[i] = strlen(title[i]);
70     //шапка таблицы
71     cout << char(218) << setfill(char(196));
72     for (int j = 0; j < m - 1; j++)
73         cout << setw(wn[j]) << char(194);
74     cout << setw(wn[m - 1]) << char(191) << endl;
75     cout << char(179);
76     for (int j = 0; j < m; j++)
77         cout << setw((wn[j] - size[j]) / 2) << setfill(' ') << ' ' << title[j]
78         << setw((wn[j] - size[j]) / 2) << char(179);
79     cout << endl;
80     for (int i = 0; i < k; i++)
81     { //заполнение таблицы
82         cout << char(195) << fixed;
83         for (int j = 0; j < m - 1; j++)
84             cout << setfill(char(196)) << setw(wn[j]) << char(197);
85
86         cout << setw(wn[m - 1]) << char(180) << setfill(' ') << endl;
87
88         int field_l = ((wn[0] - strlen(i_prn[i].name)) / 2);
89
90         cout << char(179) << setw((wn[0] - strlen(i_prn[i].name)) / 2) << ' ' <<
91         << i_prn[i].name << setw((wn[0] - strlen(i_prn[i].name)) / 2);
92         if ((wn[0] - strlen(i_prn[i].name)) % 2)
93             cout << ' ';
94         cout << char(179);
```

```

95     cout << setw(wn[1] - 1) << setprecision(10) << i_prn[i].i_toch << char(179)
    (179)
96     << setw(wn[2] - 1) << i_prn[i].i_sum << char(179)
97     << setw(wn[3] - 1) << i_prn[i].n << char(179) << endl;
98 }
99 //низ таблицы
100 cout << char(192) << setfill(char(196));
101 for (int j = 0; j < m - 1; j++)
102     cout << setw(wn[j]) << char(193);
103 cout << setw(wn[m - 1]) << char(217) << endl;
104 //восстановление первоначальных значений
105 cout << setprecision(6) << setfill(' ');
106 system("chcp 1251>nul");
107 }
108 void printIntResults(double (*searchType) (TPF, int)) {
109     cout << "Область интегрирования функций: " << a << " <= x <= " << b <<
    endl;
110     cout << "Точность вычисления: " << eps << endl;
111     I_print results[5];
112     for (int i = 0; i < 5; ++i) {
113         I_print p;
114         p.name = funcNames[i];
115         p.i_toch = getRightInt(funcs[i]);
116         p.i_sum = getInt(funcs[i], searchType);
117         p.n = n;
118         n = startingN;
119         results[i] = p;
120     }
121     PrintTabl(results, 5);
122     while (!_kbhit());
123 }
124 short menu() {
125     cout << "=====Menu===== " << endl;
126     cout << "| 1 - Вычислить интеграл методом прямоугольника | " << endl;
127     cout << "| 2 - Вычислить интеграл методом трапеции | " << endl;
128     cout << "| 3 - Ввести значение eps | " << endl;
129     cout << "| 4 - Ввести границы вычислений (a, b) | " << endl;
130     cout << "| 5 - Задать начальную точность n | " << endl;
131     cout << "| 6 - Выход | " << endl;
132     cout << "===== " << endl;
133     cout << "Выберите номер: " << endl;
134     int choice;
135     cin >> choice;
136     while (cin.fail()) {
137         cout << "Ошибка ввода. Повторите ввод\n";
138         cin.clear();
139         cin.ignore(10, '\n');
140         cin >> choice;
141     }

```




```
142     return choice;
143 }
144 void getEps() {
145     cout << "Выберите точность вычислений(eps)" << endl;
146     cout << "Текущее значение: " << eps << endl;
147     cin >> eps;
148     if (eps <= 0)
149         eps = 0.000001;
150 }
151 void getAB() {
152     cout << "Введите a и b через пробел:" << endl;
153     cin >> a >> b;
154     if (a > b)
155         swap(a, b);
156 }
157 void getN() {
158     cout << "Введите n > 1:" << endl;
159     cin >> startingN;
160     if (startingN < 2) {
161         cout << "Введено некорректное значение n." << endl;
162         cout << "Используется n = 2" << endl;
163         startingN = 2;
164     }
165 }
166 void showWrongKeyPressed() {
167     cout << "Wrong key pressed" << endl;
168     cout << "Press any key to continue" << endl;
169     while (!_kbhit());
170 }
171 void setupOutput() {
172     system("chcp 1251 > nul");
173     cout << fixed << showpoint;
174 }
175
176 int main () {
177     setupOutput();
178
179     while (true) {
180         short choice = menu();
181         system("cls");
182         switch (choice)
183         {
184             case 1:
185                 printIntResults(intRect);
186                 break;
187             case 2:
188                 printIntResults(intTrap);
189                 break;
190             case 3:
```

```
191         getEps();
192         break;
193     case 4:
194         getAB();
195         break;
196     case 5:
197         getN();
198         break;
199     case 6:
200         return 0;
201     default:
202         showWrongKeyPressed();
203     }
204     system("cls");
205 }
206 }
207
```

Анализ результатов


Метод прямоугольников:

 C:\Users\Dmitriy\Documents\study\programming basics\Lab6\Debug\Lab6.exe

Область интегрирования функций: $-1.000000 \leq x \leq 3.000000$

Точность вычисления: 0.000010

Function	Integral	IntSum	N
x	4.0000000000	4.0000000000	2
sin(22x)	-0.0000142441	0.0000948646	8
x^4	48.8000000000	48.8000089010	2048
atan(x)	2.1570201976	2.1570120596	128
cos(x)	0.9825909929	0.9825859951	256

 Выбрать C:\Users\Dmitriy\Documents\study\programming basics\Lab6\Debug\Lab6.exe

Область интегрирования функций: $-1.000000 \leq x \leq 3.000000$

Точность вычисления: 0.000001

Function	Integral	IntSum	N
x	4.0000000000	4.0000000000	2
sin(22x)	-0.0000142441	-0.0000136786	64
x^4	48.8000000000	48.8000005563	8192
atan(x)	2.1570201976	2.1570196890	512
cos(x)	0.9825909929	0.9825906805	1024

Метод трапеций:

C:\Users\Dmitriy\Documents\study\programming basics\Lab6\Debug\Lab6.exe

Область интегрирования функций: $-1.000000 \leq x \leq 3.000000$

Точность вычисления: 0.000010

Function	Integral	IntSum	N
x	4.0000000000	4.0000000000	2
sin(22x)	-0.0000142441	0.0000948646	8
x^4	48.8000000000	48.8000089010	2048
atan(x)	2.1570201976	2.1570120596	128
cos(x)	0.9825909929	0.9825859951	256

C:\Users\Dmitriy\Documents\study\programming basics\Lab6\Debug\Lab6.exe

Область интегрирования функций: $-1.000000 \leq x \leq 3.000000$

Точность вычисления: 0.000001

Function	Integral	IntSum	N
x	4.0000000000	4.0000000000	2
sin(22x)	-0.0000142441	-0.0000136786	64
x^4	48.8000000000	48.8000005563	8192
atan(x)	2.1570201976	2.1570196890	512
cos(x)	0.9825909929	0.9825906805	1024