Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт Информационных технологий и анализа данных  
  
Отделение прикладной математики и информатики

ВВЕДЕНИЕ В ЭТАПЫ РАЗРАБОТКИ ПРОГРАММ.

ОТЧЕТ  
по лабораторной работе № 2  
Вариант №15

по дисциплине «программирование»

Выполнил

Студент группы ЭВМб-23-1 \_\_\_\_\_\_\_\_\_\_\_\_Г. М. Распутин  
  
Принял

Доцент \_\_\_\_\_\_\_\_\_\_\_\_\_ А. Б. Столбов

Иркутск 2023

# Введение в этапы разработки программ. Задание А.

## Условие задачи.

*15.* Вычислить значение функции F(x), разложенной в ряд и сходящейся в указанной области.

Функция и область сходимости: 

Разложение в ряд: 

## **Таблица внешних спецификаций.**

Таблица 1.1 – Внешние спецификации

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Имя | Назначение | Тип | Вх/Вых. | Диапазон |
| 1 | eps | Точность или погрешность, используемая для первого типа вычисления. | Вещественное число. | Вход | 0.1E-20 < eps < 1 |
| 2 | x | Изначальное число, используемое для дальнейших расчётов. | Вещественное число. | Вход | X ∈ (-0.1E8; -1) ∪ (1; 0.1E8) |
| 3 | k | Количество итераций для цикла for, который используется во втором типе расчётов. | Целое число. | Вход | 0 < k < 0.1E5 |
| 4 | sum | Итоговая сумма, полученная в ходе решения. | Вещественное число. | Выход | 1.3E-38< sum <1.3E38 |

## **3. Алгоритмизация**.

Блок-схема алгоритма представлена на рис. 1.1

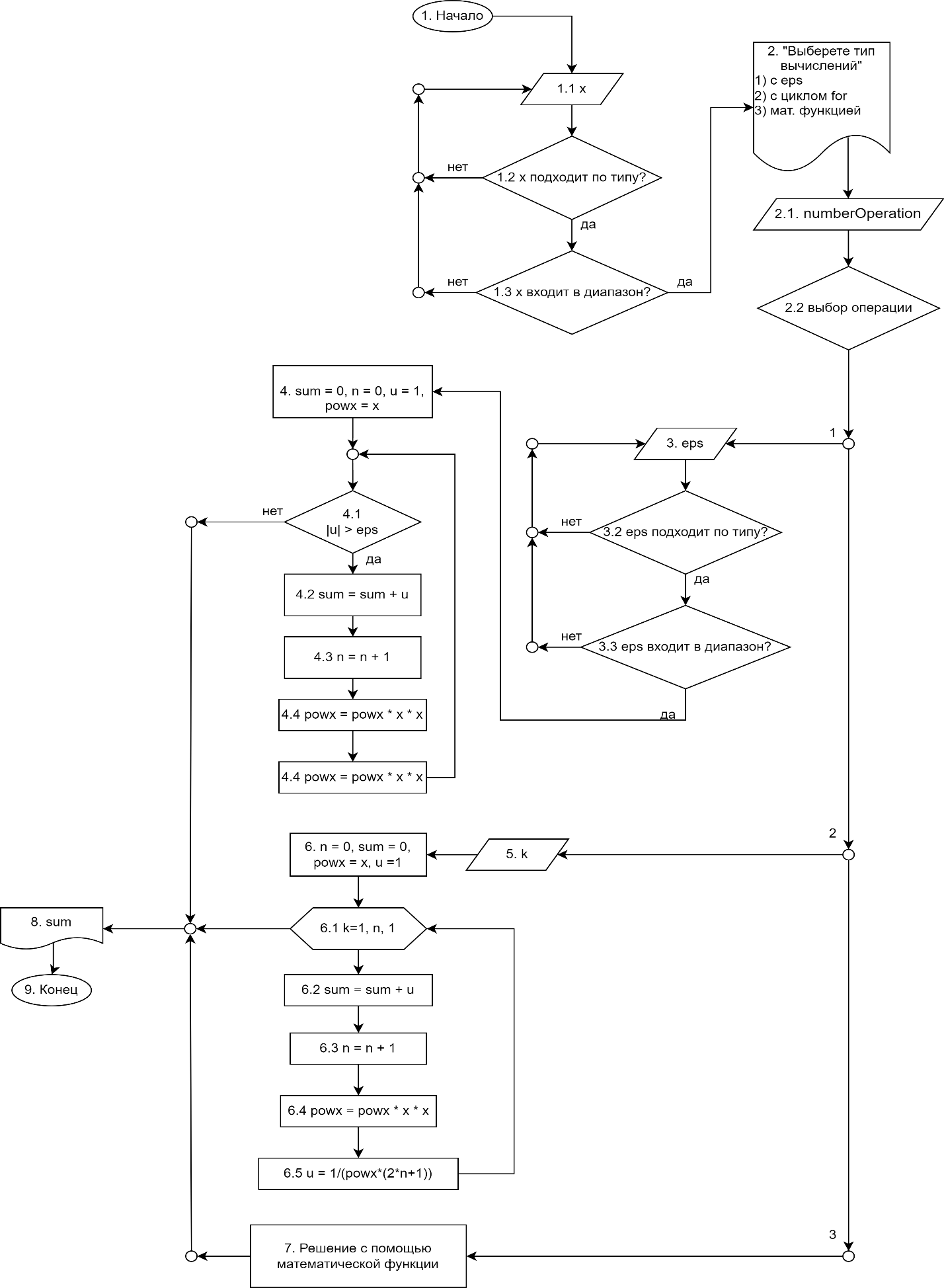


Рисунок 1.1 – Блок-схема алгоритма

## 4. Проектирование тестов

Таблица 1.2 – Таблица тестов.

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Назначение теста | Входные данные | Выходные данные |
| 1 | Простой тест работы вычисления с помощью погрешности. | X = 100, numberOperation = 1,  eps = 0.00003 | sum = 0.020001 |
| 2 | Простой тест работы вычисления с помощью цикла for. | X = 100, numberOperation = 2,  K = 25 | sum = 0.020001 |
| 3 | Простой тест работы вычисления с помощью математической функции. | X = 100, numberOperation = 3 | sum = 0.0200007 |
| 4 | Тест ввода числа с помощью экспоненты. | X = -0.3E5, numberOperation = 1, eps = 0.3E-5 | sum = -6.666667e-05 |
| 5 | Тест ввода числа без помощи экспоненты. | X = -30000, numberOperation = 1, eps = 0.000003 | sum = -6.666667e-05 |

## 5. Проверка правильности алгоритма с помощью таблицы тестов.

Тест 1: блоки 1, 1.1, 1.2, 1.3, 2, 2.1, **2.2**, 3, 3.1, 3.2, 4, 4.1, 4.2, 4.3, 4.4, 4.5…4.1, 8, 9

Тест 2: блоки 1, 1.1, 1.2, 1.3, 2, 2.1, **2.2**, 5, 6, 6.1, 6.2, 6.3, 6.4, 6,1…8, 9

Тест 3: блоки 1, 1.1, 1.2, 1.3, 2, 2.1, **2.2**, 7, 8, 9

Тест 4: блоки 1, 1.1, 1.2, 1.3, 2, 2.1, **2.2**, 3, 3.1, 3.2, 4, 4.1, 4.2, 4.3, 4.4, 4.5…4.1, 8, 9

Тест 5: блоки 1, 1.1, 1.2, 1.3, 2, 2.1, **2.2**, 3, 3.1, 3.2, 4, 4.1, 4.2, 4.3, 4.4, 4.5…4.1, 8, 9

Тест 6: блоки 1, 1.1, 1.2, 1.3, 2, 2.1, **2.2**, 3, 3.1, 3.2, 4, 4.1, 4.2, 4.3, 4.4, 4.5…4.1, 8, 9

## 6. Кодирование алгоритма или запись алгоритма на языке С++ (ссылка на код: <https://replit.com/@GieorghiiRasput/Lab2A#workA.cpp> )

#include <cmath>

#include <iostream>

#include <sstream>

#include <stdlib.h>

using namespace std;

int main() {

cout << "calculator function ln((x + 1) / (x - 1))\n";

float x, eps, sum = 0, powx, u = 1;

int n = 0;

int k = 0;

float dx1 = -0.1e8, dx2 = 0.1e8, deps1 = 0.1e-20, deps2 = 1;

// eps - погрешность

// sum - начальная сумма

// k - количество итераций для for

// powx - накопление степеней x в знаменателе

// n - количество итераций

// u - очередное слагаемое

// dx1, dx2 - диапозон значений x

// deps1, deps2 - диапозон значений eps

//циклическая проверка ввода Х

cout << "x in range (-10000000; -1) \* (1; 10000000)\n";

cout << "enter x" << endl;

while (1) {

stringstream ss;

string buferNum;

cin >> buferNum;

ss << buferNum;

if (ss >> x) {

try {

if (!(dx1 <= x and x <= dx2) or (x >= -1 and x <= 1)) {

throw 1;

}

cout << "x = " << x << "\n";

break;

}

catch (int err) {

cout << "error! number is not in range\n";

}

}

else {

cout << "Error with type\n";

}

}

powx = x;

cout << "Enter the number of the desired way to solve the example:\n1.\twith "

"eps\n2.\twith for\n3.\twith math func\n\n";

int numberOperation;

cin >> numberOperation;

if (numberOperation == 1) // eps

{

//вычисление с помощью eps. 1.1 пункт.

/\*........................................ВВОД ПЕРЕМЕННОЙ

\* EPS......................................\*/

cout << "eps in range (0;1)\n";

cout << "enter eps" << endl;

//циклическая проверка ввода eps

while (1) {

stringstream ss;

string buferNum;

cin >> buferNum;

ss << buferNum;

if (ss >> eps) {

try {

if (!(deps1 < eps and eps <= deps2)) {

throw 1;

}

cout << "eps = " << eps << "\n";

break;

}

catch (int err) {

cout << "error! number is not in range\n";

}

}

else {

cout << "Error with type\n";

}

}

/\*........................................КОНЕЦ ВВОДА ПЕРЕМЕННОЙ

\* EPS......................................\*/

int exp;

cout << "select display mode\n";

cout << "1.\tscientific\n2.\tfixed\n";

cin >> exp;

cout << "calculating the amount using eps:" << endl;

if (exp == 1) {

cout.setf(ios\_base::scientific);

}

if (exp == 2) {

cout.setf(ios\_base::fixed);

}

do {

u = 1 / (powx \* (2 \* n + 1));

sum += u;

cout << "n = " << n << "\t"

<< "sum = " << sum \* 2 << "\t"

<< "u = " << u << endl;

powx = powx \* x \* x;

n += 1;

} while (abs(u) > eps);

}

else if (numberOperation == 2) // K

{

/\*........................................ВВОД ПЕРЕМЕННОЙ

\* К......................................\*/

cout << "Enter k by character. If you want to stop typing, write the "

"symbol f \n";

int k = 0;

while (1) {

char c;

cin >> c;

if (c >= '0' && c <= '9') {

k = k \* 10 + (c - '0');

cout << "current k: " << k << endl;

}

else if (c == 'f')

break;

}

/\*........................................КОНЕЦ ВВОДА ПЕРЕМЕННОЙ

\* К................................\*/

//Вычисление суммы первых n членов ряда. 1.2 пункт.

cout << "calculating the amount using for:" << endl;

for (n = 0, sum = 0, powx = x, u = 1; n <= k; n++) {

u = 1 / (powx \* (2 \* n + 1));

sum += u;

cout << "n = " << n << "\t"

<< "sum = " << sum \* 2 << "\t"

<< "u = " << u << endl;

powx = powx \* x \* x;

}

}

else if (numberOperation == 3) {

//использование математической формулы. 1.3 пункт.

cout << "calculating the amount using math func:" << endl;

cout << log((x + 1) / (x - 1));

cout << endl;

}

cout << "again? Y/N\n";

string again\_wow;

cin >> again\_wow;

if (again\_wow == "Y") {

system("cls");

main();

}

return (0);

}

# Введение в этапы разработки программ. Задание Б.

## Условие задачи.

1. Текст содержит только цифры в возрастающем порядке;

## **Кодирование алгоритма или запись алгоритма на языке С++** (ссылка на код: <https://replit.com/@GieorghiiRasput/Lab2B#main.cpp>)

//15. Текст содержит только цифры в возрастающем порядке;

#include <iostream>

using namespace std;

int main() {

char i;

char prev = '0';

cout << "Enter text. End - enter\n";

do {

i = getchar();

if (!(i == '0' || i == '1' || i == '2' || i == '3' || i == '4' || i == '5' || i == '6' || i == '7' || i == '8' || i == '9' || i == '\n' ))

{

cout << "The text contains not only numbers in ascending order\n";

return(0);

}

if (prev > i && i != '\n')

{

cout << "The text contains not only numbers in ascending order\n";

return(0);

}

prev = i;

} while (i != '\n');

cout << "The text contains only numbers in ascending order";

return(0);

}

# Введение в этапы разработки программ. Задание В.

## Условие задачи.

1. Найти все натуральные числа, не превосходящие заданного N и делящиеся на каждую из своих цифр.

## **Таблица внешних спецификаций**

Таблица 3.1 – Внешние спецификации

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Имя | Назначение | Тип | Вх/Вых. | Диапазон |
| 1 | n | Число, которое не должны превосходить натуральные числа, делящиеся на каждую из своих цифр. | Натуральное число | Вход | 0 < N < 0.3E10 |
| 2 | num | Натуральные числа. | Натуральное число | Выход | 0 < num < 0.3E10 |

## **Проектирование тестов.**

Таблица 3.2 – Таблица тестов.

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Назначение теста | Входные данные | Выходные данные |
| 1 | Вывод всех подходящих чисел от 1 до 50 | N = 50 | 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 15, 22, 24, 33, 36, 44, 48 |
| 2 | Вывод всех подходящих чисел от 1 до 100. | N = 100 | 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 15, 22, 24, 33, 36, 44, 48, 55, 66, 77, 88, 99 |

## **4. Кодирование алгоритма или запись алгоритма на языке С++** (ссылка на код: <https://replit.com/@GieorghiiRasput/Lab2C#main.cpp>)

//15. Найти все натуральные числа, не превосходящие заданного N и делящиеся на каждую из своих цифр.

#include <iostream>

#include <string>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int n;

bool lier = true;

cout << "Найти все натуральные числа, не превосходящие заданного N и делящиеся на каждую из своих цифр.\n";

cout << "Введите число N\n";

cin >> n;

for (int num = 1; num <= n; num++)

{

lier = true;

string s = to\_string(num);

for (const int i : s) {

if (i - 48 == 0) { lier = false; break; }

if ((num % (i - 48) != 0)) { lier = false; break; }

}

if (lier == true) cout << num << ", ";

}

return(0);

}