

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ  
БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
Институт математики и информационных систем  
Факультет автоматики и вычислительной техники  
Кафедра электронных вычислительных машин

Отчёт по лабораторной работе №3  
по дисциплине  
«Программирование»  
по теме: 'Изучение базовых принципов организации процедур и функций'

Выполнил студент гр. ИВТб-1301-06-00 \_\_\_\_\_ /Габдулбариев Т.Р./  
Проверил доцент кафедры ЭВМ \_\_\_\_\_ /Долженкова М.Л./

Киров  
2025

## **Цель**

Цель лабораторной работы: освоить синтаксис построения процедур и функций, изучить способы передачи данных в подпрограммы, получить навыки организации минимального пользовательского интерфейса.

## **Решение задачи**

Задание 1. Реализовать программу вычисления площади фигуры, ограниченной кривой

$$1 * x^3 + (1) * x^2 + (-3) * x + (14) \text{ и осью ОХ (в положительной части по оси ОУ).}$$

2. Вычисление определенного интеграла должно выполняться численно, с применением метода трапеций.
3. Пределы интегрирования вводятся пользователем.
4. Взаимодействие с пользователем должно осуществляться посредством case-меню.
5. Требуется реализовать возможность оценки погрешности полученного результата.
6. Необходимо использовать процедуры и функции там, где это целесообразно.

Исходный код программы для решения задачи на языке Pascal:

```
uses System;

var
  a, b, Itrue, Itrap: real;
  step, menuPos: integer;
  errorFlag: integer;

const
  ROOT = -3.249; // Известный корень уравнения

procedure Error(a, b: real; step: integer; var flag: integer); forward;

procedure enterA(var a: real);
begin
  Console.Clear;
  Console.Write('Введите нижний предел интегрирования a: ');
  readln(a);
end;

procedure enterB(var b: real);
begin
  Console.Clear;
  Console.Write('Введите верхний предел интегрирования b: ');
  readln(b);
end;

procedure enterStep(var step: integer);
begin
  Console.Clear;
```

```

Console.WriteLine('Введите количество разбиений (n): ');
readln(step);
end;

function f(x: real): real;
begin
  f := x*x*x + x*x - 3*x + 14;
end;

function Fperv(x: real): real;
begin
  Fperv := (x*x*x*x)/4 + (x*x*x)/3 - (3*x*x)/2 + 14*x;
end;

function TrapIntegral(a, b: real; n: integer): real;
var
  h, sum: real;
  i: integer;
begin
  h := (b - a)/n;
  sum := (f(a) + f(b))/2;
  for i := 1 to n-1 do
    sum := sum + f(a + i*h);
  TrapIntegral := sum*h;
end;

procedure calcIntegral(a, b: real; step: integer);
var
  aInt: real;
begin

```

```

Console.Clear;

errorFlag := 0;
Error(a, b, step, errorFlag);

if errorFlag = 0 then
  exit;

if a < ROOT then
  aInt := ROOT
else
  aInt := a;

Itrap := TrapIntegral(aInt, b, step);
Itrue := Fperv(b) - Fperv(aInt);

writeln('Функция: f(x) = x^3 + x^2 - 3x + 14');
writeln('Корень уравнения: x = ', ROOT:0:3);
writeln('Интегрирование на интервале [', aInt:0:3, '; ', b:0:3, ']');
writeln;
writeln('Метод трапеций: ', Itrap:0:6);
writeln('Точное значение: ', Itrue:0:6);
readln;

end;

procedure calcError(a, b: real; step: integer);
var
  absErr, relErr, aInt: real;
begin
  Console.Clear;

```

```

errorFlag := 0;
Error(a, b, step, errorFlag);

if errorFlag = 0 then
    exit;

if a < ROOT then
    aInt := ROOT
else
    aInt := a;

Itrap := TrapIntegral(aInt, b, step);
Itrue := Fperv(b) - Fperv(aInt);

absErr := abs(Itrue - Itrap);
if Itrue <> 0 then
    relErr := absErr / abs(Itrue) * 100
else
    relErr := 0;

writeln('Корень уравнения: x = ', ROOT:0:3);
writeln('Интервал интегрирования: [', aInt:0:3, '; ', b:0:3, ']');
writeln('Абсолютная погрешность: ', absErr:0:6);
writeln('Относительная погрешность: ', relErr:0:4, '%');
readln;

end;

procedure Error(a, b: real; step: integer; var flag: integer);

```

```

begin
  flag := 1;
  if (step <= 0) or (b <= a) then
begin
  writeln('Ошибка: требуется b > a и n > 0.');
  writeln('Нажмите Enter для возврата в меню...');

  readln;
  flag := 0;
  exit;
end;

if b <= ROOT then
begin
  writeln('Ошибка: интервал полностью слева от корня.');
  writeln('Корень уравнения x = ', ROOT:0:3);
  writeln('Нажмите Enter для возврата в меню...');

  readln;
  flag := 0;
  exit;
end;

```

procedure showMenu;

```

begin
  Console.Clear;
  writeln('Используйте стрелки ↑/↓ и Enter');
  writeln('Корень уравнения: x = ', ROOT:0:3);
  writeln;

  for var i := 1 to 6 do

```

```

begin

    if i = menuPos then
        Console.ForegroundColor := ConsoleColor.Cyan
    else
        Console.ForegroundColor := ConsoleColor.White;

    case i of
        1: writeln('Ввод нижнего предела a      Текущее: ', a:0:3);
        2: writeln('Ввод верхнего предела b      Текущее: ', b:0:3);
        3: writeln('Ввод числа шагов n          Текущее: ', step);
        4: writeln('Вычислить интеграл');
        5: writeln('Оценить погрешность');
        6: writeln('Выход');

    end;
end;

Console.ForegroundColor := ConsoleColor.White;
end;

procedure MoveMenuUp;
begin
    dec(menuPos);
    if menuPos < 1 then menuPos := 6;
end;

procedure MoveMenuDown;
begin
    inc(menuPos);
    if menuPos > 6 then menuPos := 1;
end;

```

```

begin
  a := 0; b := 0; step := 0; menuPos := 1;
  errorFlag := 1;

  while true do
    begin
      showMenu;
      var key := Console.ReadKey(true);
      case key.Key of
        ConsoleKey.UpArrow: MoveMenuUp;
        ConsoleKey.DownArrow: MoveMenuDown;
        ConsoleKey.Enter:
          case menuPos of
            1: enterA(a);
            2: enterB(b);
            3: enterStep(step);
            4: calcIntegral(a,b,step);
            5: calcError(a,b,step);
            6: exit;
          end;
    end;
  end;
end.

```

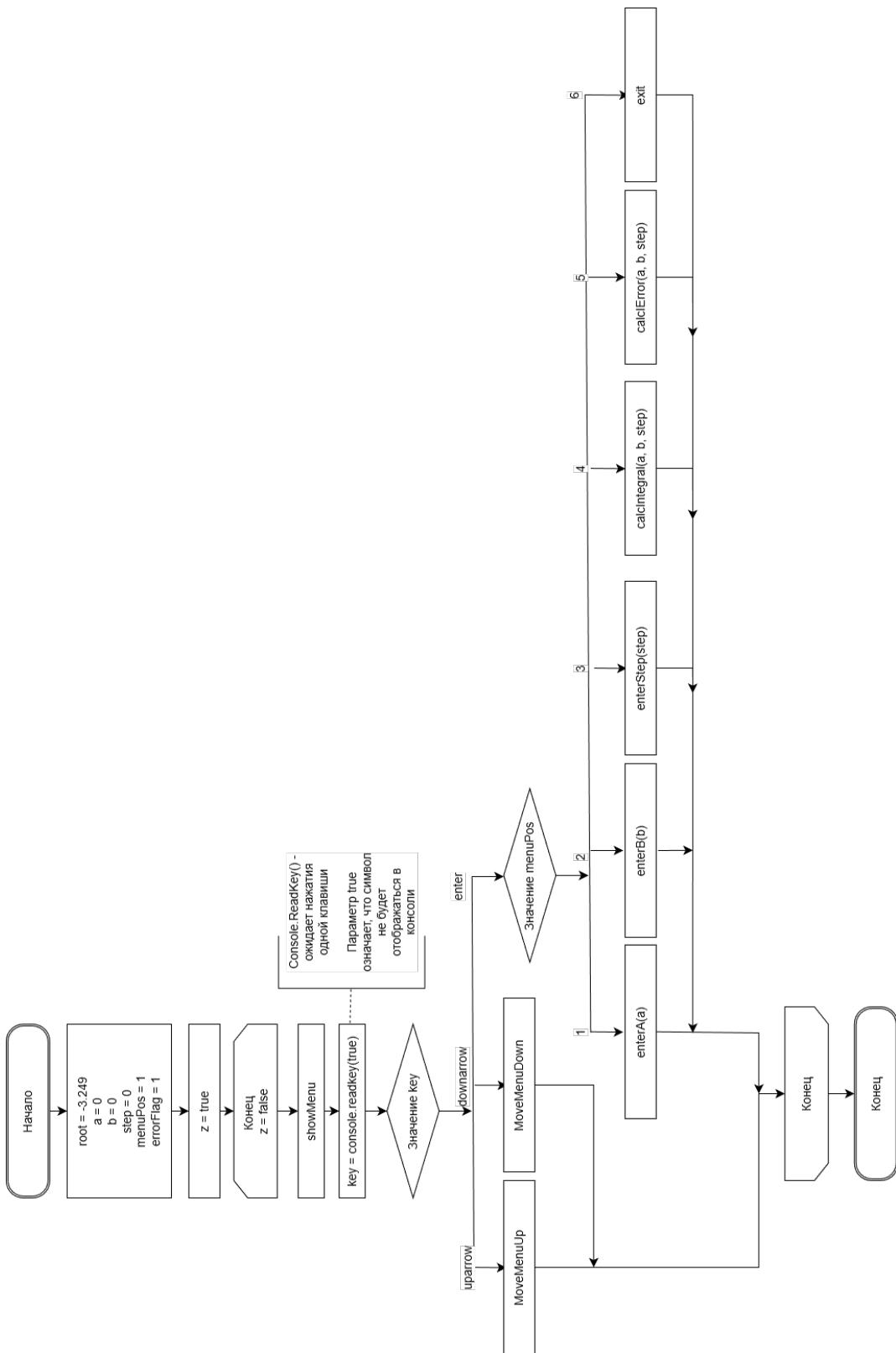


Рис. 1: Основная программа

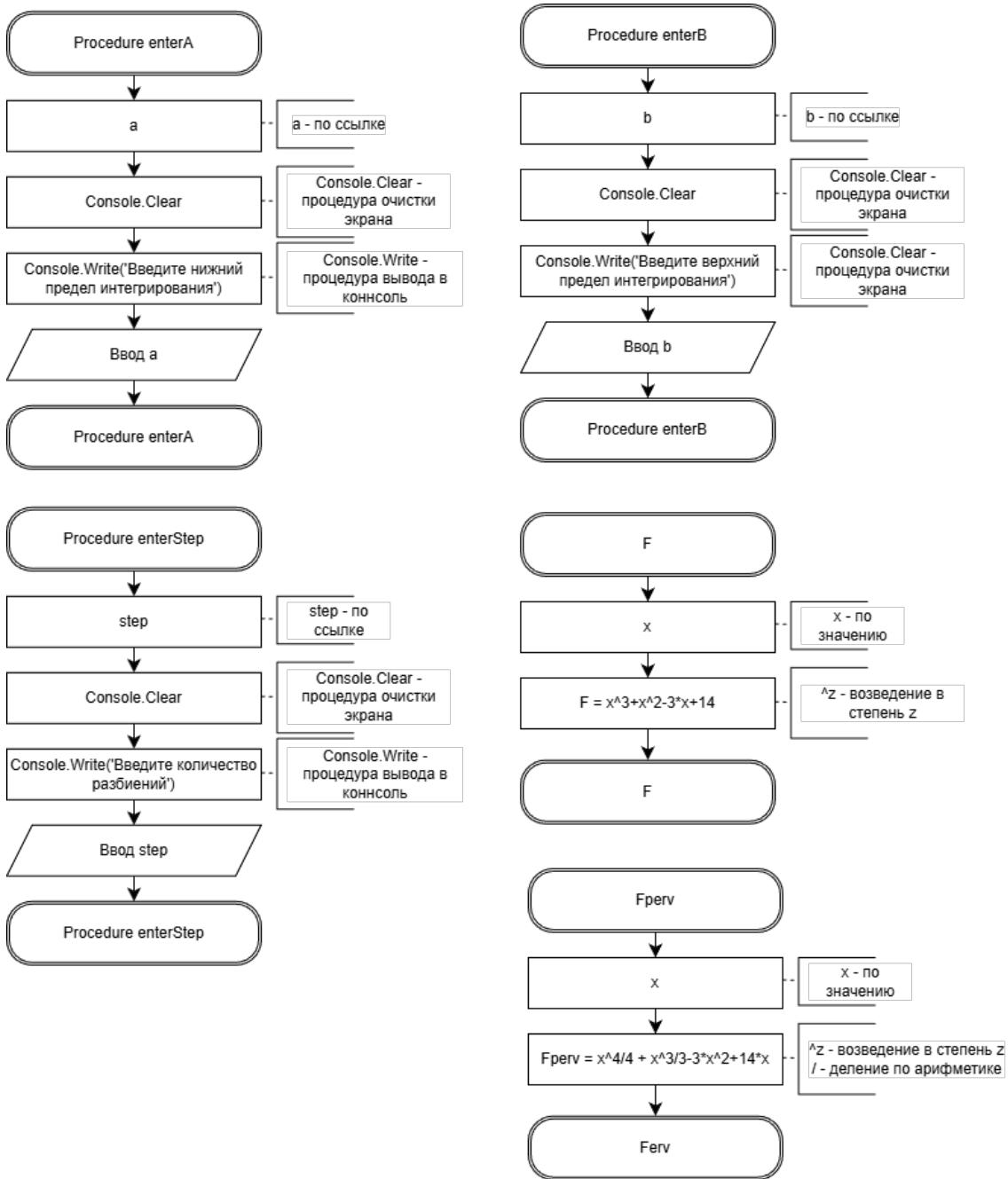


Рис. 2: Ввод пределов и подсчет функции, и ее первообразной

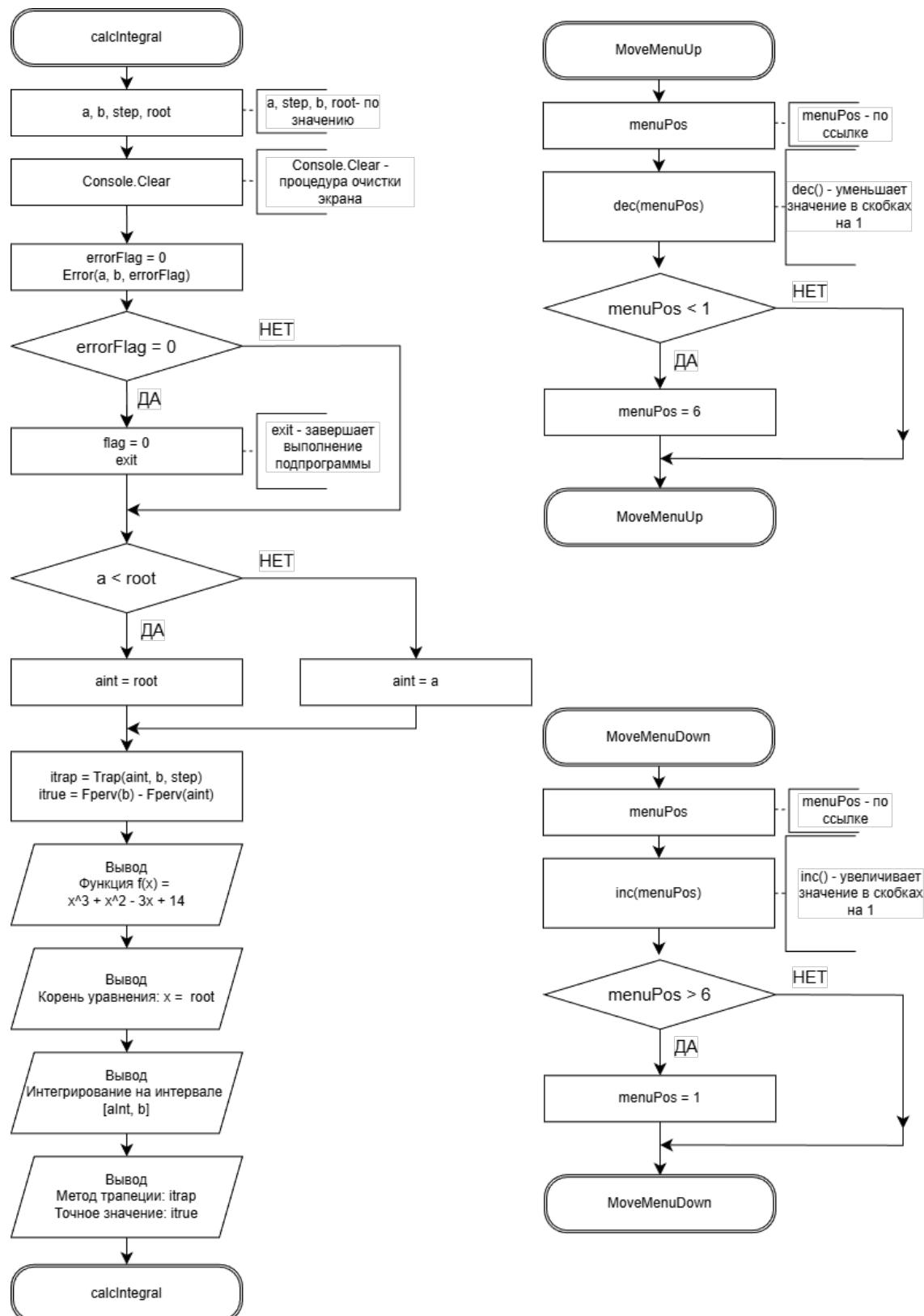


Рис. 3: Пункт подсчет интеграла и процедуры управления на стрелки

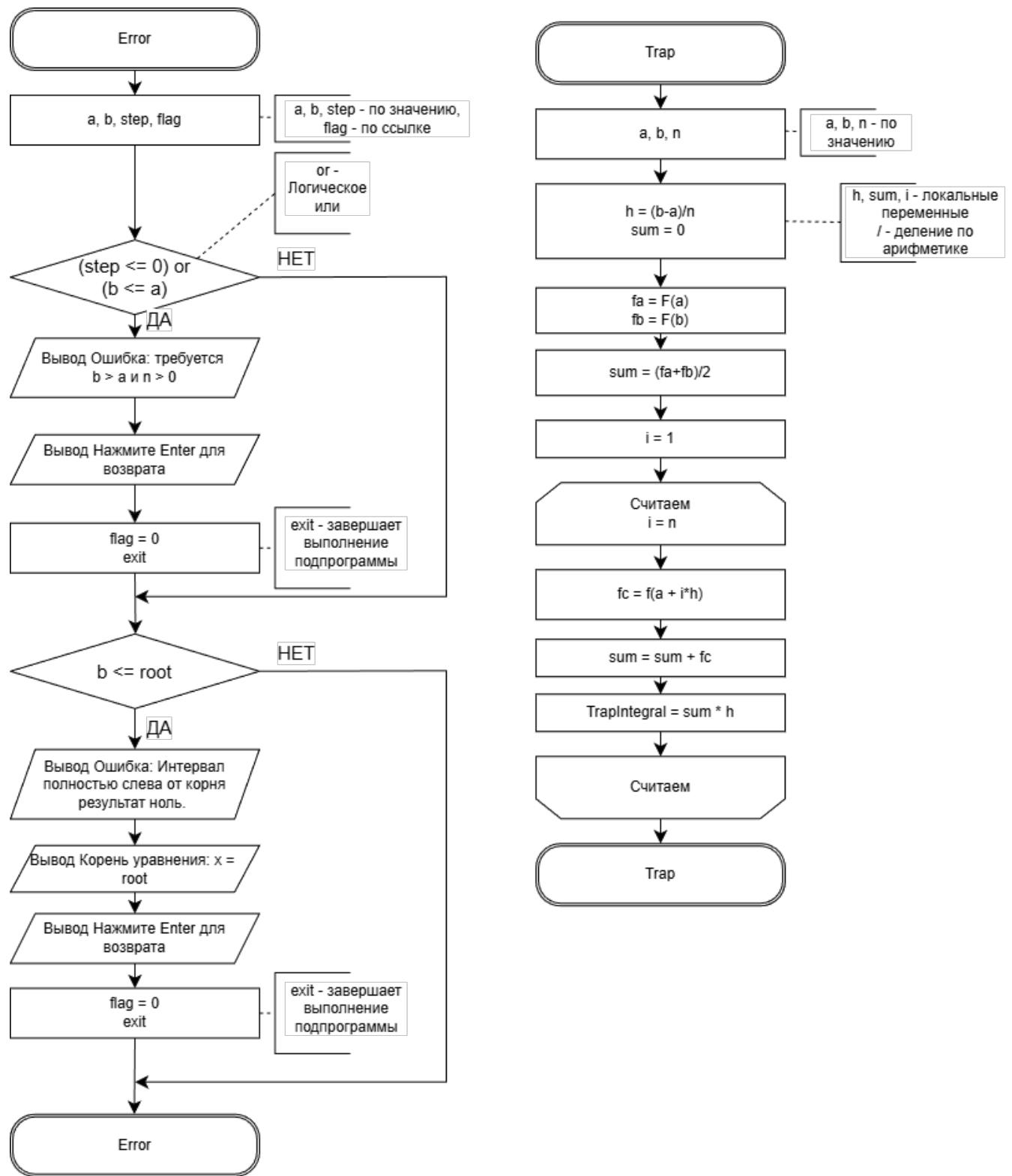


Рис. 4: Проверка на корректность данных и подсчет интеграла методом трапеции

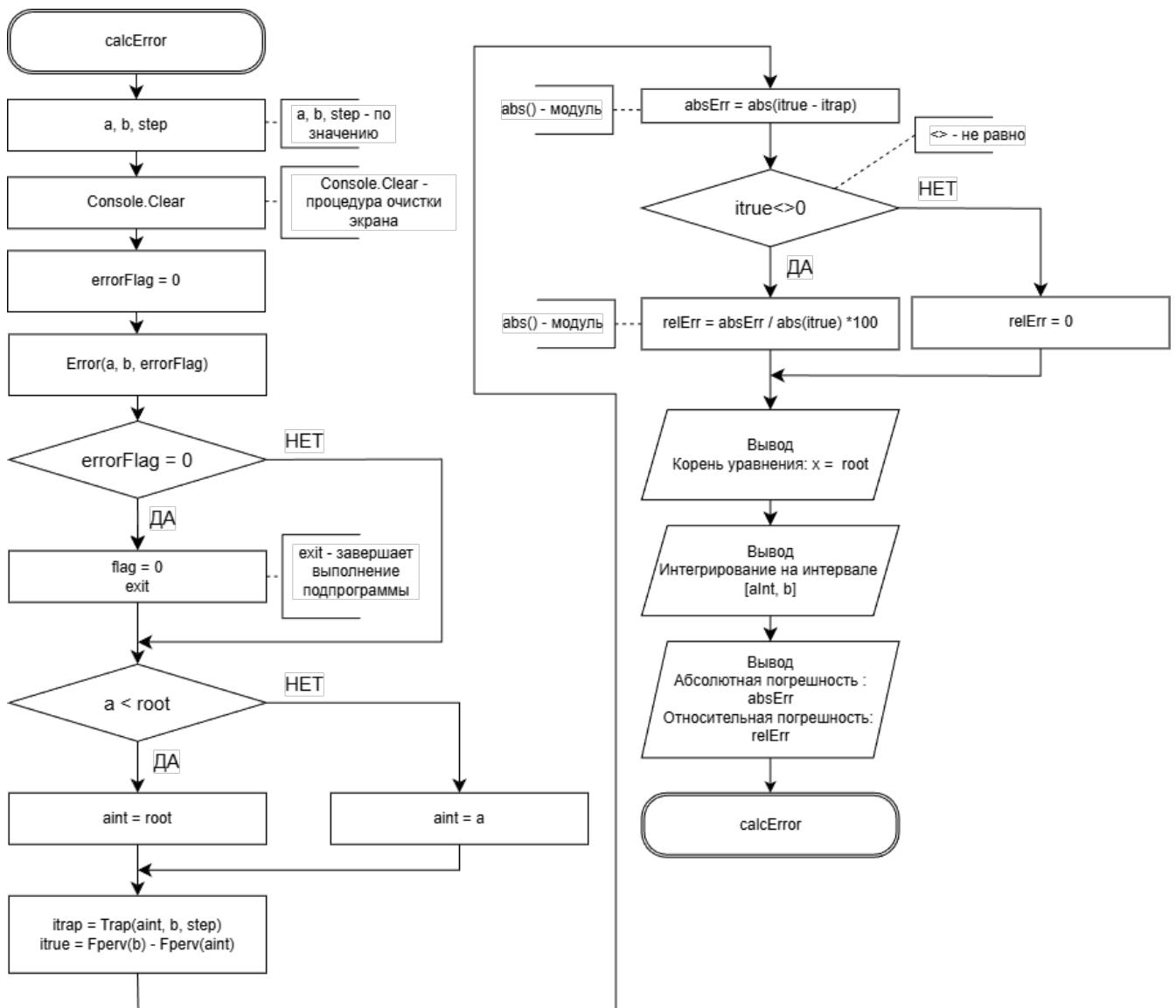


Рис. 5: Подсчет погрешности

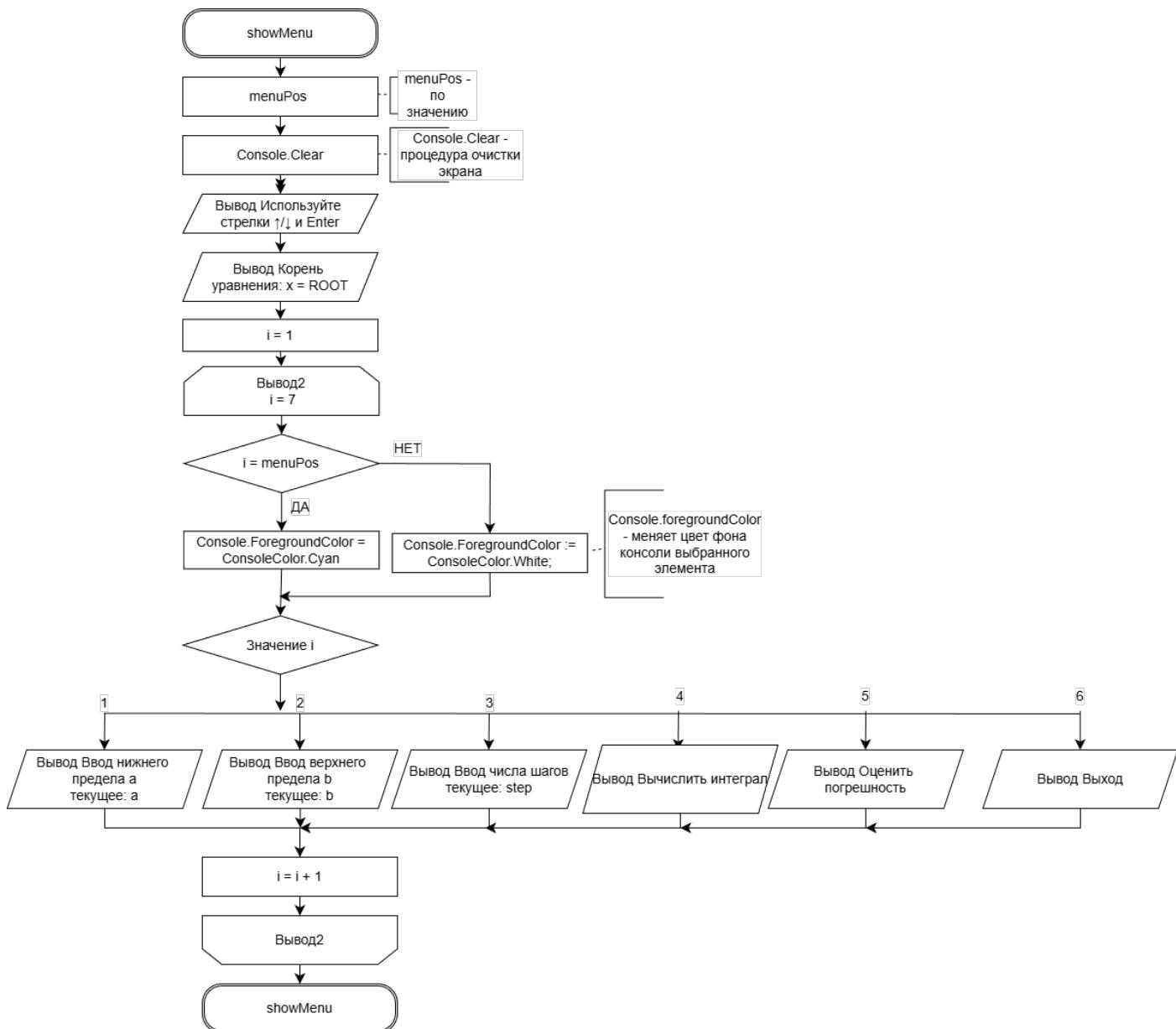


Рис. 6: Вывод меню

## Вывод

Мы освоили синтаксис построения процедур и функций, изучили способы передачи данных в подпрограммы, получили навыки организации минимального пользовательского интерфейса.