Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №1**

**«Вычисление значения функции»**

**ПО ДИСЦИПЛИНЕ «Основы алгоритмизации и программирования»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

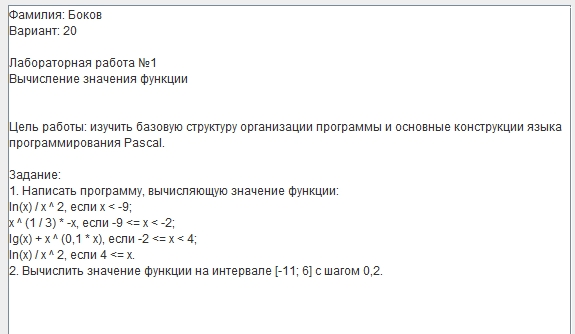
2025

В отчете должны отображаться:

1. Цель работы
2. Формулировка задания (с вариантом)
3. Описание алгоритма и ответы на вопросы
4. Схема алгоритма с комментариями
5. Код программы
6. Результат выполнения программы
7. Вывод
8. **Цель работы**

Целью работы является изучение базовой структуры организации программ и основных конструкций языка программирования Pascal

1. **Формулировка задания**



1. **Описание алгоритма и ответы на вопросы**

Определяем функцию task\_one(x: integer): real, которая:

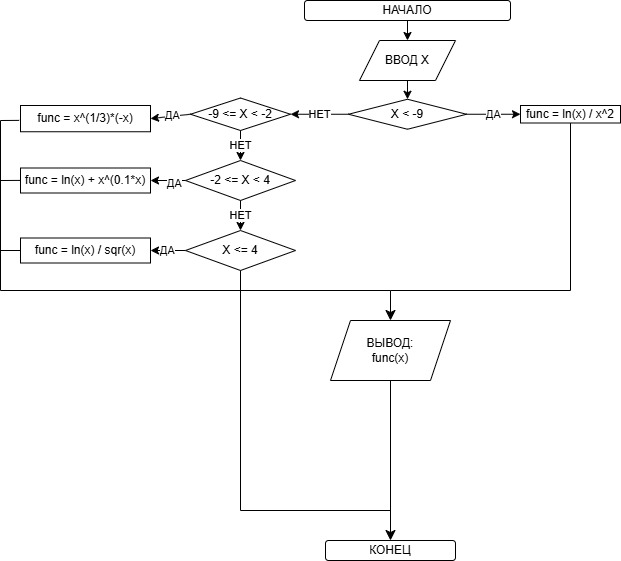
* Определяет лямбда-функцию func (которая будет соответствовать выбранной функции) с условными операторами внутри для выбора формулы в зависимости от x.
* Вычисляет y = func(x) и выводит его.
* Запрашивает у пользователя, нужно ли выполнить второе задание. Если ответ 'y', вызывает процедуру task\_two, передавая ей функцию func.

Определяем процедуру task\_two(start\_: real; end\_: integer; step: real; func: real -> real), которая:

* Выводит заголовок таблицы.
* Запускает цикл с предусловием x <= end\_, при каждом шаге увеличивает x на step\_. В цикле для каждого x вычисляет y = func(x) и выводит x и y

**Ответы на вопросы:**

1. Алгоритм — это четкая последовательность команд, выполнение которой приводит к решению поставленной задачи.
2. Виды алгоритмов:  
   линейный, разветвляющийся, циклический.
3. Условный (разветвляющийся) алгоритм — это алгоритм, в котором в зависимости от выполнения некоторого условия совершается либо одна, либо другая последовательность действий.
4. Виды условных алгоритмов: полный и неполный. В решении задачи использован полный условный алгоритм с несколькими ветвями для выбора формулы функции в зависимости от x.
5. Циклический алгоритм — это алгоритм, в котором некоторые команды повторяются несколько раз подряд до выполнения определенного условия или заданного количества повторений.
6. Виды циклических алгоритмов: с предусловием, с постусловием, с параметром. В решении задачи использован циклический алгоритм с предусловием (while) в процедуре task\_two для перебора значений x и вывода таблицы.
7. **Схема алгоритма**



1. **Код программы**

uses math;

procedure task\_two(start\_: real; end\_: integer; step: real; func: real -> real);

begin

  writeln('|   x   |   y   |');

  writeln('¦---------------¦');

  var x := start\_;

  while x <= end\_ do

  begin

    var y := func(x);

    writeln('¦', x:6:2, ' |', y:7:2, '¦');

    x += step;

  end;

  writeln('¦---------------¦');

end;

function task\_one(x: integer): real;

begin

  var func: real -> real;

  var doTaskTwo: string := 'n';

  if x < -9 then

  begin

    func := x -> ln(x) / (x\*x);

    writeln(func(x));

  end

  else if (x >= -9) and (x < -2) then

  begin

    func := x -> power(x, 1/3) \* x \* (-1);

    writeln(func(x));

  end

  else if (x >= -2) and (x < 4) then

  begin

    func := x -> ln(x) + power(x, 0.1\*x);

    writeln(func(x));

  end

  else if (4 <= x) then

  begin

    func := x -> ln(x) / sqr(x);

    writeln(func(x));

  end

  else

    func := x -> 0;

  writeln('Выполнить второе задание? (y/n, default - n)');

  readln(doTaskTwo);

  if doTaskTwo = 'y' then

    task\_two(-11, 6, 0.2, func);

end;

begin

  writeln(task\_one(5));

end.

1. **Результат выполнения программы**

Для x = 5 в первом задании вывод:  
0.064377516497364  
Во втором задании:

…¦ 5.00 | 0.06¦

¦ 5.20 | 0.06¦

¦ 5.40 | 0.06¦

¦ 5.60 | 0.05¦

¦ 5.80 | 0.05¦

¦ 6.00 | 0.05¦

1. **Вывод**

В ходе работы была разработана программа для вычисления определенной функции с использованием условных конструкций, лямбда-функций и процедур. Алгоритм позволяет выбирать формулу для любого x и выводить табличные данные. Обнаружена проблема с областью определения функций (ln(x) для x <= 0), что приводит к nan в таблице для отрицательных x в некоторых ветвях