МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Физико-технический факультет

Кафедра вычислительной техники и электроники (ВТиЭ)

Лабораторная работа № 01

**Интегрированные среды разработки программного обеспечения**

Выполнил студент 595 гр.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.В. Осипенко

Проверил: к.т.н,, доцент каф. ВТиЭ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ю.Г. Скурыдин

Лабораторная работа защищена

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Барнаул 2019

1. **Формулировка задачи**

Создать программу, производящую вычисления по формуле:

1. **Постановка задачи**

На вход программы поступает вещественное число в диапазоне +/- 1.7E-308 до 1.7E+308, такое что это число в кубе не было равно противоположному по знаку значению константы b () . На выходе выдается в результате вычисления вещественное число по приведенной формуле:

1. **Математическая модель**

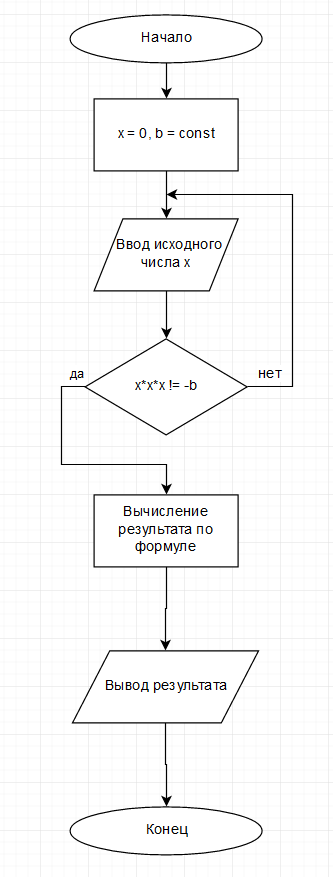
Для построения математической модели рассмотрим пример. Произведем вычисления, передав программе на вход число 15(при b = 5). Воспользуемся распространенным методом – посчитаем получившееся выражение на калькуляторе. В итоге получим число: -8433.17 .

1. **Описание алгоритма**

Начало

* 1. Ввод исходного числа x
  2. Проверка, что неравно b (иначе возврат к пункту 4.1)
  3. Вычисление результата по формуле
  4. вывод результат на экран
  5. Завершение программы

Конец

1. **Опорный граф (блок-схема) алгоритма**
2. **Проект программы с определением замкнутых программных единиц и необходимых структур данных**

В данном случае программа состоит из одного блока – блока ввода и вывода. Структуры данных не используются.

1. **Текст программы:**

**на языке С++**

**#include** <iostream>// стандартная библиотека ввода и вывода потока

**#include** <cmath>// математическая библиотека

**#include** <conio.h>// консольная библиотека ввода и вывода

**using namespace std**;// использование стандартного пространства имен

**int** main(){// главная функция

**int** x = 0;

**const int** b = 5;// объявления переменных x и b

**do**{// основной блок программы

**cout** << "Pleas, enter variable x in range from +/- 1.7E-308 to 1.7E+308: ";

**cin** >> x;// Призыв ввести значения для переменной

**if** (x\*x\*x == -b) **continue**;// Проверка условия

**cout** << "Your result is " << 3.14\*(x\*x) - (x\*x\*x)\*log(sqrt(abs((x\*x) + (2\*sin(x))/(b+(x\*x\*x))))) << endl;// подсчет результата и вывод его в консоль

**break**;//выход

}**while(1);**

getch();// запрос любой клавиши

**return** 0;// вернуть ноль если программа завершилась в штатном режиме

}

**на языке Pascal**

program lab1;// name of program

uses math;// using lib math

var // variable

x:real;

const //constants

b = 5;

pi = 3.14;

begin // start main block of algorithm

writeln('Pleas, enter variable x in range from +/- 1.7E-308 to 1.7E+308:');

read(x); // чтение числа x

write('Your result is: ');

writeln(pi\*(x\*x) - (x\*x\*x)\*ln(sqrt(abs((x\*x) + (2\*sin(x))/(b+(x\*x\*x))))));

// вывод результата

end.

1. **Проверка работоспособности (тестирование) программы**

Для проверки работоспособности программ по запросу введем число 15. На выходе программы должны выдать число -8433.17.

1. **Сравнительный анализ и оценка эффективности работы программ на разных языках программирования**

Ввиду простоты программ различий в скорости работы нет или оно минимально.

1. **Формулировка задачи**

В трехзначном числе x зачеркнули его последнюю цифру. Когда в оставшемся двузначном числе переставили цифры, а затем приписали к ним слева последнюю цифру числа x, то получилось число 654. Найти число x.

1. **Постановка задачи**

На вход программа ничего не поступает. Произвести преобразования над трехзначным целочисленным числом. На выход поступает итоговое трехзначное целочисленное число.

1. **Математическая модель**

Нам дано число 654. Нам нужно вычеркнуть первую цифру в этом числе. Для этого поделим число на 100: 654 / 100 = 6 , 54 в остатке(запомним его). Далее нужно поменять местами цифры числа 54, для этого поделим число на 10 и затем прибавим к результату остаток от деления, умноженный на 10: 54 / 10=5, 4 в остатке; 5 + 4 \* 10 = 45. Теперь необходимо в конец приписать число, которое мы вычеркнули в самом начале(6). Чтобы это сделать умножим наше число на 10 и прибавим 6: 45 \* 10 = 450; 450 + 6 = 456. 456 наш итоговый ответ.

1. **Описание алгоритма**

Начало

* 1. Делим 654 на 100 целочислено и с остатком
  2. Делим 54 на 10 целочислено и с остатком
  3. Прибавляем 4\*10
  4. Умножаем на 10 и прибавляем 6
  5. Выводим ответ

Конец

1. **Опорный граф (блок-схема) алгоритма**

Конец

Вывод результата

Начало

Делим 654 на 100 целочислено и с остатком

Делим 54 на 10 целочислено и с остатком

Прибавляем 4\*10

Умножаем на 10 и прибавляем 6

1. **Проект программы с определением замкнутых программных единиц и необходимых структур данных**

В данном случае программа состоит из одного основного блока. Структуры данных – нет.

1. **Текст программы:**

**на языке С++**

#include <iostream>

int main()

{

int var = 654;//Наше число

// Делим число на 100, запоминая остаток

int temp = var/100;

var %= 100;

// Делим число на 10

int b = var % 10;

var /= 10;

// Прибавляем остаток от прошлого деления, умноженный на 10

var += b \* 10;

// Умнажаем на 10 и прибавляем результат от первого деления(654/100)

var = var\*10 + temp;

// Выводим результат

printf("Result %d",var);

return 0;

}

**на языке Pascal**

var

temp,x,b: integer;

begin

x := 654;//Наше число

// Делим число на 100, запоминая остаток

temp := x div 100;

x := x mod 100;

// Делим число на 10

b := x mod 10;

x := x div 10;

// Прибавляем остаток от прошлого деления, умноженный на 10

x :=x + b \* 10;

// Умнажаем на 10 и прибавляем результат от первого деления(654/100)

x := x\*10 + temp;

// Выводим результат

writeln(x);

end.

1. **Проверка работоспособности (тестирование) программы**

Для проверки работоспособности программ запустим их и увидим число 456.

1. **Сравнительный анализ и оценка эффективности работы программ на разных языках программирования**

Ввиду простоты программ различий в скорости работы нет или оно минимально.