**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра математического обеспечения и применения ЭВМ**

**ОТЧЕТ**

**по практической работе №1**

**по дисциплине «Вычислительная математика»**

**Тема:**

**Исследование погрешностей.**

|  |  |
| --- | --- |
| Студент гр. 3344 | Гайфутдинов А.Р. |
| Преподаватель | Попова Е.В. |

Санкт-Петербург

2025

**Цель работы.**

Изучение особенностей вычисления чисел с плавающей точкой.

**Теоретические положения.**

Множество F чисел с плавающей запятой характеризуется четырьмя параметрами: основанием системы счисления b, точностью t и интервалом показателей [L, M]. Каждое число x с плавающей запятой, принадлежащее F, имеет следующее значение (1)

(1)

где целые числа удовлетворяют неравенствам и Целое число n называется показателем, а число - дробной частью. Если принять, что

, то *t* называется разрядностью мантиссы, а *m* - разрядностью порядка. Определенная таким образом мантисса оказывается в диапазоне Расположение представленных чисел на числовой оси уже не обладает свойством равномерности.

Действительная машинная реализация представлений чисел с плавающей точкой может отличатся в деталях от рассматриваемой идеальной, однако различия несущественны, и на практике их почти всегда можно игнорировать, анализируя основные проблемы ошибок округления. Величина является оценкой относительной точности плавающей арифметики, которая характеризуется посредством машинного эпсилон, т.е. наименьшего числа с плавающей точкой ε, такого, что 1+ε>1. Точное значение машинного эпсилон зависит не только то указанных выше параметров, но и от принятого способа округления. В вычислительных машинах используются различные системы чисел с плавающей точкой, причем в некоторых ЭВМ несколько систем. Так, для современных ПЭВМ характерно применение двух систем, которые называются обычной точностью и удвоенной точностью.

На множестве F определены арифметические операции в соответствии с тем, как они выполняются ЭВМ. Эти операции, в свою очередь моделируются в машине посредством приближений, называемых плавающими операциями. Для плавающих операций сложения, вычитания, умножения и деления существует возможность возникновения ошибок округления, переполнения и появления машинного нуля. Следует отметить, что операции плавающего сложения и умножения коммутативны, но не ассоциативны, и дистрибутивный закон для них также не выполняется. Невыполнение указанных алгебраических законов, имеющих фундаментальное значение для математического анализа, приводит к сложности анализа плавающих вычислений и возникающих при этом ошибок.

**Задание.**

1. С помощью языка программирования Java, онлайн-компилятора Trinket исследовать величину машинного эпсилон при заданных в варианте начальных значениях на определенных технических устройствах, указать эти технические устройства, привести скриншоты.
2. С помощью метода границ найти приближенное значение функции зависящей от двух аргументов, и величину абсолютной погрешности функции, если , . Предварительно определить область определения функции, отобразить эту область графически. (Если границы аргументов не попадают в область определения – замените их, аргументируя).
3. Для нечетных номеров: по неточному значению все цифры которого верны в строгом смысле, определить предельное значение относительной погрешности. Для четных номеров: неточное число из предыдущего варианта (для 2-го из 1-го) имеет относительную погрешность . Определить количество верных знаков в широком смысле неточного числа.
4. Исследовать ошибки, связанные с числами с плавающей точкой.

А) В программе TestSum (в онлайн компиляторе замените имя класса) суммируется последовательность чисел, которая начинается с 0.01 и заканчивается 1.0. Числа в последовательности увеличиваются на 0.01 следующим образом: 0.01 + 0.02 + 0.03 и так далее.

public class TestSum {

public static void main(String[] args) {

// Инициализировать sum

float sum, i;

sum = 0;

// Прибавить 0.01, 0.02, ..., 0.99, 1 к sum

for (i = 0.01f; i <= 1.0f; i = i + 0.01f)

sum += i;

// Отобразить sum

System.out.println("Сумма равна " + sum); }}

Сравните результат с точным значением.

Б) В программе замените тип float на double для повышения точности. Объясните полученный результат.

В) Измените программу, используя целочисленный счетчик для прибавления всех чисел к sum.

Г) Измените программу, прибавляя эти же числа от наибольшего к наименьшему. Выведите все результаты в таблицу и дайте им объяснения.

**Вариант 6.**

|  |  |  |  |
| --- | --- | --- | --- |
| 6 | eps = 6 |  |  |

**Выполнение работы.**

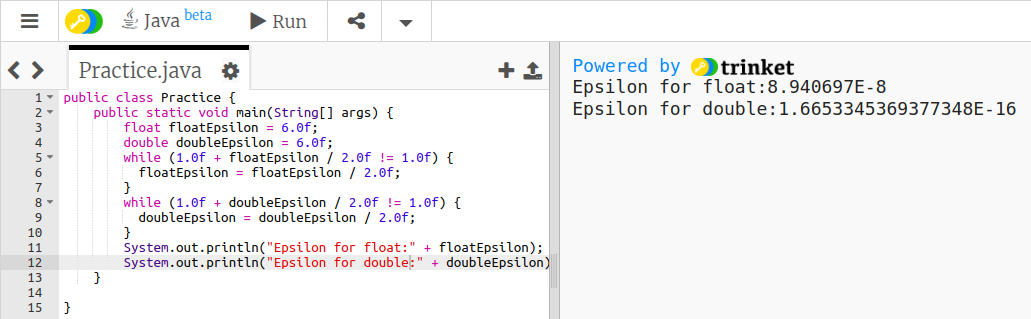
***1) Исследование величины машинного эпсилон.***

Операционная система : Linux x64, v: 5.15.0-1026-aws.

Компилятор : trinket.

Версия Java : 1.8.0\_352.

Машинный эпсилон — это наименьшее число ε, для которого выполняется условие: 1 + ε > 1, где ε — число с плавающей точкой.

Расчёт машинного эпсилон для типов данных double и float:

Результаты:

Для float: **8.940697E×10−8**

Для double: **1.6653345369377348×10−16**

***2) Вычисление приближенного значения функции с помощью метода границ.***

С помощью метода границ вычисляется приближенное значение функции u, при ограничениях -0,879 < x < 0,936, -1,234 < y < 0,126.

Область определения функции ограничена следующими неравенствами:

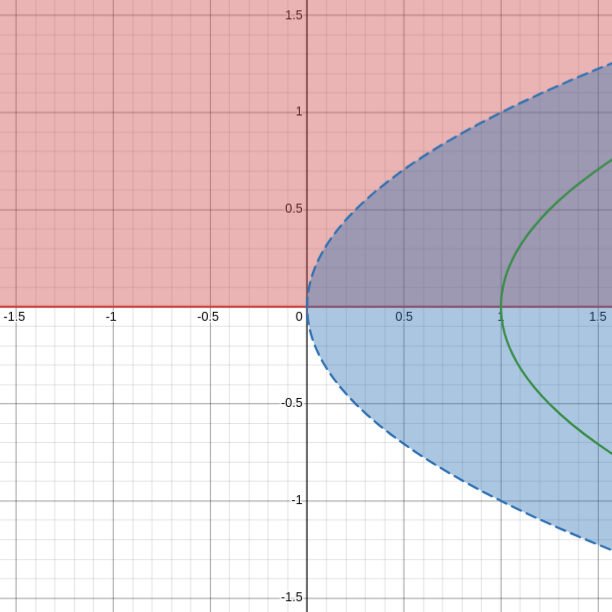
1. y >= 0 : ограничения ввиду аргумента квадратного корня (y).

2. x - y^2 > 0 : ограничения ввиду аргумента натурального логарифма (x-y^2).

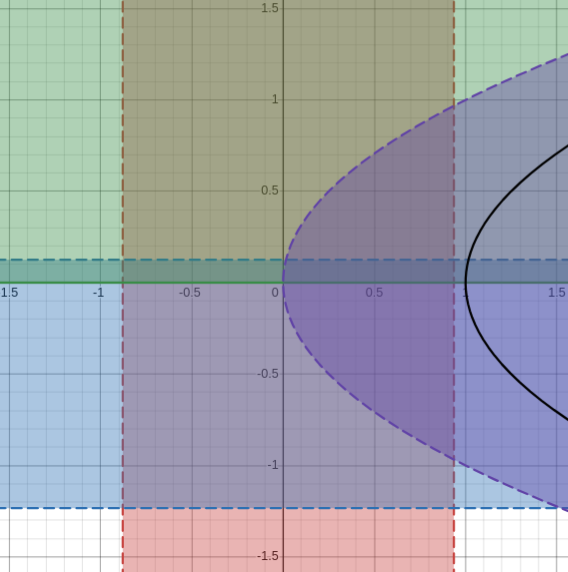
3. x - y^2 != 1 : ограничение ввиду запрета делить на ноль (x - y^2) (логарифм равен 0 при значении аргумента = 1).

Область определения представлена пересечением областей красного

синего цветов с “выколотой” инвертированной параболой:



Общий вид области определения функции с учётом ограничений, описанных в задании (-0,879 < x < 0,936, -1,234 < y < 0,126), поедставлен пересечением красной, зелёной, синкей и фиолетовой областей:



Приближение данного участка плоскости с указанием координат граничных точек:



Расссмотрим нижнюю границу для x и y, равную 0,1 и 0,01 соответственно, ввиду обнуления значения x и кривой, ограничивающей участок слева.

С применением метода границ определим приближенное значение функции на выбранном участке.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Выражение | Значение нижней границы значения выражения в общем виде  (НГ) | Значение верхней границы  значения выражения в общем виде (ВГ) | Нижняя граница значения рассматриваемого выражения | Верхняя граница значения рассматриваемого выражения |
| x | НГ(x) | ВГ(x) | 0,1 | 0,936 |
| y | НГ(y) | ВГ(y) | 0,01 | 0,126 |
| y^2 | НГ(y)^2 | ВГ(y)^2 | 0,01 ^ 2 = 0,0001 | 0,126 ^ 2 = 0,015876 |
| x-y^2 | НГ(x)-ВГ(y^2) | ВГ(x)-НГ(y^2) | 0,1 - 0,015876 = 0,084124 | 0,936 - 0,0001 = 0,9359 |
| e^(-x) | e^(-ВГ(x)) | e^(-НГ(x)) | e ^ ( -0,936 ) = 0,392193476 | e ^ ( -0,1 ) = 0,904837418 |
| √y | √ НГ(y) | √ ВГ(y) | √ 0,01 = 0,1 | √ 0,126 = 0,354964787 |
| ln(x-y^2) | ln(НГ(x-y^2)) | ln(ВГ(x-y^2)) | ln( 0,084124 ) = −2,475463378 | ln( 0,9359 ) = −0,066246646 |
| e^(-x)  +  √y | НГ(e^(-x))  +  НГ(√y) | ВГ(e^(-x))  +  ВГ(√y) | 0,392193476 + 0,1 = 0,492193476 | 0,904837418 + 0,354964787 = 1,259802205 |
| (e^(-x)+√y)  /  (ln(x-y^2)) | НГ(e^(-x)+√y)  /  ВГ(ln(x-y^2) | ВГ(e^(-x)+√y) /  НГ(ln(x-y^2) | 0,492193476 / ( - 0,066246646 ) = **−7,429711626** | 1,259802205 / ( - 2,475463378 ) = **−0,508915711** |

Полученные верхняя и нижняя границы значения функций:

ВГu = **- 0,508915711**

НГu = **- 7,429711626**

Приближенное значение функции и величина абсолютной погрешности:

u = (ВГu + НГu) / 2 = ( - 7,429711626 - 0,508915711 ) / 2 = **−3,969313668**

Δu = (ВГu - НГu) / 2 = ( - 0,508915711 + 7,429711626 ) / 2 = **3,460397958**

***3) Определение количества верных в широком смысле знаков.***

Дано неточное число x = 5,71 , относительная погрешность равна 0,0056. Определить количество верных знаков в широком смысле неточного числа.

Значащая цифра данного разряда является верной в широком смысле, если абсолютная погрешность не превышает единицы данного разряда.

Абсолютная погрешность:

∆ x = δ(x) \* x

∆ x = 0,0056 \* 5,71 = **0,03**

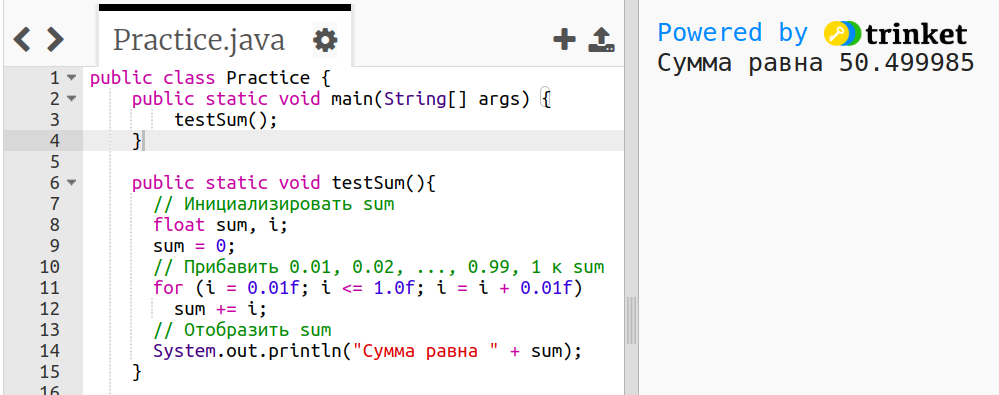
Абсолютная погрешность превышает единицу разряда 10^(-2), однако не превышает единицу более старших разрядов.

Следовательно, в данном случае верных знаков в широком смысле всего **2**: 5 и 7.

***4) Исследование ошибок, связанных с числами с плавающей точкой.***

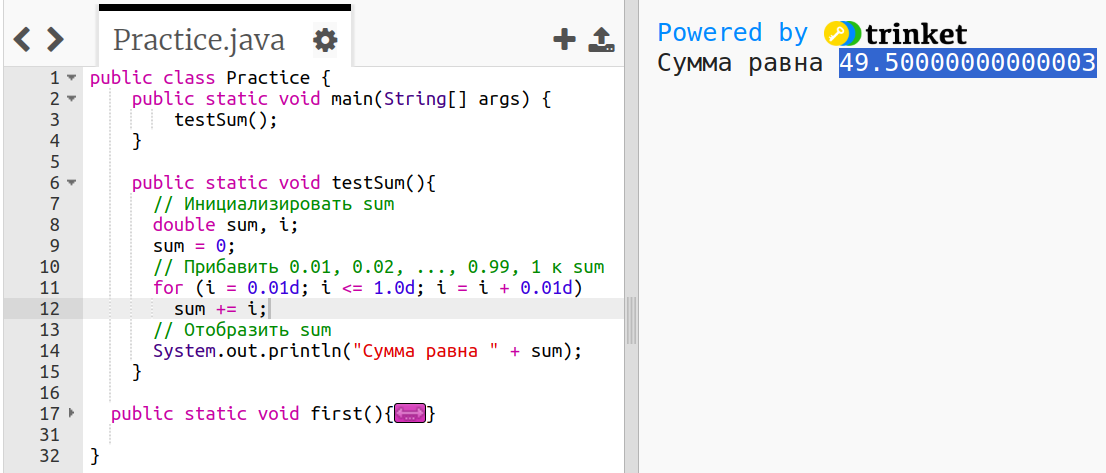
*А) Результат работы исходной программы:*

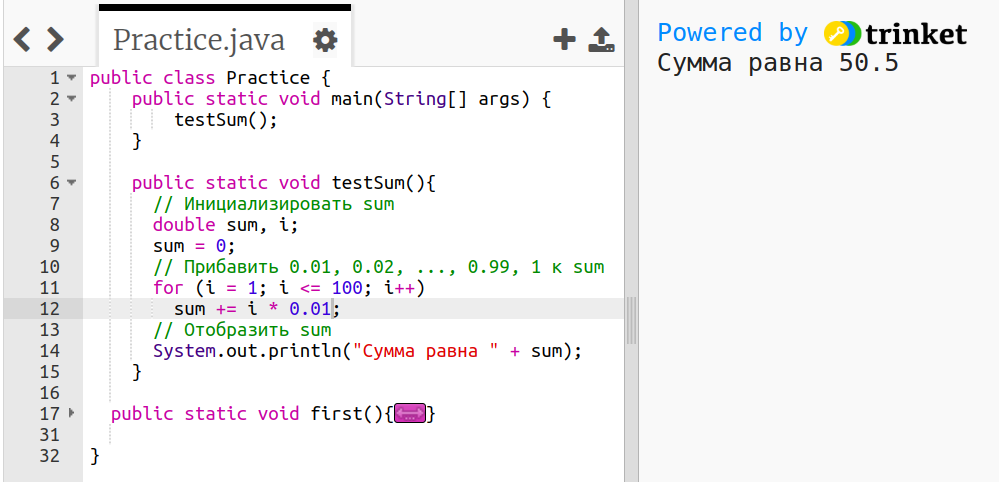
**sum = 50.499985**

 Разница между ожидаемым и полученным значением обуславливается ошибками округления на каждой итерации цикла.

*Б) Результат работы программы с типом счётчика double:*

**sum = 49.50000000000003**

 Это значение также отличается от ожидаемого результата. Разница между значениями, полученными с использованием float и double типов для хранения обусловлена разным количеством байт для хранения информации. Для float количество байт равно 8, для double - 16. Большее количество выделяемой памяти для хранения не даёт повышения точности, что означает, что причина такой погрешности кроется в выбранном алгоритме и аппаратной реализации операций сложения и сравнения. В данном случае, скорее всего, происходит пропуск последней итерации цикла.

 *В) Результат работы программы с типом счётчика int:*

**sum = 50,5**

Данное значение соответствует ожидаемому результату. Это связано с дискретным характером чисел int и единственностью их представления, что обеспечивает корректность вычислений без ошибок округления.

*Г) Результат работы программы с типом счётчика double и итерированием от большего к меньшему:*

**sum = 50.48999999999995**

 Результат отличается от ожидаемого (50,5) ввиду накопления погрешности при сложении чисел с плавающей точкой. Эта погрешность, в свою очередь, обусловлена неточностью хранения данных переменными типа double.

|  |  |  |
| --- | --- | --- |
| Модификация программы | Результат | Комментарий |
| A) Тип счётчика - float, итерирование от меньшего к большему | 50.499985 | Ошибка связана с накоплением ошибок округления данных float по ходу выполнения программы. |
| Б) Тип счётчика - double, итерирование от меньшего к большему | 49.50000000000003 | Ошибка связана с пропуском последней итерации добавления единицы, специфично для double. |
| В) Тип счётчика - int, итерирование от меньшего к большему | 50,5 | Дискретный счётчик обеспечивает безопасность от непредсказуемого поведения и ошибок округления, благодаря чему программа выдаёт корректный результат. |
| Г) тип счётчика - double, итерирование от большего к меньшему | 50.48999999999995 | Крайне высокая точность для числа с плавающей точкой, обусловлено отсутствием пропуска последней итерации и большей точностью double по сравнению с float. |

**Вывод.**

В ходе выполнения работы был исследован машинный эпсилон, обусловленный аппаратной частью ЭВМ, на которой была запущена программа на Java.

С помощью метода границ было найдено приближенное значение функции и величина ее абсолютной погрешности.

Для указанных в задании данных было определено количество верных знаков числа в широком смысле.

Были изучены особенности работы с числами с плавающей точкой.