ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ ____________________

ПРЕПОДАВАТЕЛЬ

| ассистент | | И.Д. Свеженин |
| --- | --- | --- |
| должность, уч. степень, звание | подпись, дата | инициалы, фамилия |

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

Функциональное программирование

по курсу: Кроссплатформенное программирование

РАБОТУ ВЫПОЛНИЛ

| СТУДЕНТ ГР. № | 4329 | | Д.С. Шаповалова |
| --- | --- | --- | --- |
| | | подпись, дата | инициалы, фамилия |

Санкт-Петербург 2025

Содержание

**1. Цель работы:**

Изучение и практическое применение синтаксиса и возможностей языка высокого уровня Kotlin с использованием парадигмы функционального программирования.

**2. Задание:**

Программа должна вычислять значение заданной функции путем разложения в ряд Маклорена с заданной точностью и с использованием стандартной функции класса Math. Аргумент функции и точность должны задаваться пользователем. Ввод и вывод информации можно осуществлять через командную строку. При разработке следует придерживаться принципов функционального программирования. Выбранный вариант – 7.

1. $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \ldots + \frac{x^n}{n!} + \ldots$

2. $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots + (-1)^{n-1}\frac{x^{2n-1}}{(2n-1)!} + \ldots$

3. $\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots + (-1)^{n-1}\frac{x^{2n-2}}{(2n-2)!} + \ldots$

4. $sh\, x = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \frac{x^9}{9!} + \ldots + \frac{x^{2n-1}}{(2n-1)!} + \ldots$

5. $ch\, x = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \frac{x^8}{8!} + \ldots + \frac{x^{2n-2}}{(2n-2)!} + \ldots$

6. $(1+x)^m = 1 + \frac{m}{1!}x + \frac{m(m-1)}{2!}x^2 + \frac{m(m-1)(m-2)}{3!}x^3 + \ldots$ (биномиальный ряд)

7. $\frac{1}{1+x} = 1 - x + x^2 - x^3 + x^4 - x^5 + \ldots + (-1)^{n-1}x^n + \ldots$

8. $\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \ldots + (-1)^{n-1}\frac{x^n}{n} + \ldots$

9. $arctg\, x = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \ldots + (-1)^{n-1}\frac{x^{2n-1}}{(2n-1)} + \ldots$

10. $\arcsin x = x + \frac{1}{2}\frac{x^3}{3} + \frac{1\cdot 3}{2^2\cdot 2!}\frac{x^5}{5} + \frac{1\cdot 3\cdot 5}{2^3\cdot 3!}\frac{x^7}{7} + \ldots$

Рисунок 1.1 – Таблица с вариантами функций.

**3. Краткое описание хода разработки, алгоритма работы программы и назначение используемых технологий**

Ход разработки:

1. Подключение функции класса Math.
2. Разработка главной функции main:
    a. Главный цикл вычислений n-ного члена ряда.
    b. Вычисление значения суммы членов ряда по упрощённой формуле.
3. Добавление вывода результата вычислений.
4. Добавление ввода значения x и точности.
5. Исправление ошибок в вычислении суммы членов ряда.

**Используемые технологии:**

- Kotlin – используется для работы написанного кода
- Math – используется для функции pow – возведения в степень (в других вариантах для вычисления стандартной функции)

**Описание алгоритма работы программы:**

*Вывод информации пользователю*

На экран выводится название задачи, формула функции

$$f(x) = 11 + x = 1 - x + x2 - x3 + \cdots$$

и область сходимости ряда ($|\ x\ | < 1$).

*Ввод данных*

- Пользователь вводит число x.
- Пользователь задаёт требуемую точность ε (маленькое положительное число, например 0.001).

*Проверка условия сходимости*

Если введённый x не удовлетворяет условию $|\ x\ | < 1$, программа сообщает об ошибке и завершается.

*Инициализация переменных для расчёта ряда*

- sum = 0.0 — накопленная сумма ряда.
- n = 0 — номер текущего члена ряда.
- term — значение текущего члена ряда.
- lastTerm — переменная для контроля величины последнего члена.

*Циклическое вычисление ряда Маклорена*

В цикле do ... while вычисляется очередной член ряда по формуле: $a_n = (-1)^n \cdot x^n$

Этот член прибавляется к сумме: sum += term.

Значение и номер члена выводятся в виде таблицы: номер, член ряда, промежуточная сумма.

Переменная n увеличивается на 1.

Цикл продолжается, пока модуль последнего члена ряда больше заданной точности ε.

*Вычисление точного значения функции*

После выхода из цикла программа считает «стандартное» значение функции напрямую по формуле: $\mathrm{f}(x) = \frac{1}{(1+x)}$

*Сравнение результатов*

- Выводится итоговая сумма ряда.
- Выводится точное значение функции.
- Считается разница между ними:

$$\Delta = |\text{ сумма ряда } - f(x)\ |$$

*Вывод итогов*

На экран печатаются:

- введённые данные (x, ε),
- количество вычисленных членов ряда,
- итоговая сумма ряда,
- точное значение функции,
- разница между ними.

*Смысл алгоритма*

Программа приближённо вычисляет функцию $\mathrm{f}(x) = \frac{1}{(1+x)}$ с помощью ряда Маклорена, сравнивает результат с точным значением и показывает, насколько быстро и точно ряд сходится при заданном значении x.

**4. Скриншоты, иллюстрирующие результаты работы программы:**

В качестве демонстрации работы программы приведём несколько скриншотов с разными значениями x и точностью.

```
Введите x (дробное число, например 0.5): 0.5
Введите точность (маленькое число, например 0.001): 0.001

Вычисление ряда:
n   Член ряда   Сумма
------------------------
0   1,000000    1,000000
1   -0,500000   0,500000
2   0,250000    0,750000
3   -0,125000   0,625000
4   0,062500    0,687500
5   -0,031250   0,656250
6   0,015625    0,671875
7   -0,007813   0,664063
8   0,003906    0,667969
9   -0,001953   0,666016
10  0,000977    0,666992

========================================
РЕЗУЛЬТАТЫ:
Значение x: 0.5
Точность: 0.001
Количество членов ряда: 10
Сумма ряда: 0,66699219
Стандартная функция: 0,66666667
Разница: 0,00032552
========================================
```

Рисунок 2.1 – Результат работы программы, пример 1

Как мы видим из значения разницы в вычислении значения стандартной функции и суммы членов разложенного ряда, результаты почти сходятся, что говорит о правильности работы программы.

```
ВЫЧИСЛЕНИЕ ФУНКЦИИ С ПОМОЩЬЮ РЯДА МАКЛОРЕНА
Функция: (1/(1+x)) = (1 - x + x^2 - x^3 + ...)
Ряд сходится при |x| < 1

Введите x (дробное число, например 0.5): 0.23
Введите точность (маленькое число, например 0.001): 0.023

Вычисление ряда:
n   Член ряда   Сумма
------------------------
0   1,000000    1,000000
1   -0,230000   0,770000
2   0,052900    0,822900
3   -0,012167   0,810733

========================================
РЕЗУЛЬТАТЫ:
Значение x: 0.23
Точность: 0.023
Количество членов ряда: 3
Сумма ряда: 0,81073300
Стандартная функция: 0,81300813
Разница: 0,00227513
========================================
```

Рисунок 2.2 – Результат работы программы, пример 2

```
ВЫЧИСЛЕНИЕ ФУНКЦИИ С ПОМОЩЬЮ РЯДА МАКЛОРЕНА
Функция: (1/(1+x)) = (1 - x + x^2 - x^3 + ...)
Ряд сходится при |x| < 1

Введите x (дробное число, например 0.5): 0.23
Введите точность (маленькое число, например 0.001): 0.00023

Вычисление ряда:
n   Член ряда   Сумма
------------------------
0   1,000000    1,000000
1   -0,230000   0,770000
2   0,052900    0,822900
3   -0,012167   0,810733
4   0,002798    0,813531
5   -0,000644   0,812888
6   0,000148    0,813036

========================================
РЕЗУЛЬТАТЫ:
Значение x: 0.23
Точность: 2.3E-4
Количество членов ряда: 6
Сумма ряда: 0,81303581
Стандартная функция: 0,81300813
Разница: 0,00002768
========================================
```

Рисунок 2.3 – Результат работы программы, пример 3

Сравнив результаты работы с рисунков 2.2 и 2.3 можем увидеть, как влияет значение точности на вычисления – увеличивается количество членов, результат через ряд становится точнее.

```
ВЫЧИСЛЕНИЕ ФУНКЦИИ С ПОМОЩЬЮ РЯДА МАКЛОРЕНА
Функция: (1/(1+x)) * (1 - x + x^2 - x^3 + ...)
Ряд сходится при |x| < 1

Введите x (дробное число, например 0.5): 1
Введите точность (маленькое число, например 0.001): 1
Ошибка: x должен быть между -1 и 1
```

Рисунок 3.1 – Попытка ввести плохие значения

Если попытаться ввести x целым числом, а не дробным, как требуется для схождения ряда, то программа предупредит пользователя о неправильном вводе.

```
ВЫЧИСЛЕНИЕ ФУНКЦИИ С ПОМОЩЬЮ РЯДА МАКЛОРЕНА
Функция: (1/(1+x)) = (1 - x + x^2 - x^3 + ...)
Ряд сходится при |x| < 1

Введите x (дробное число, например 0.5): 0.5
Введите точность (маленькое число, например 0.001): 1

Вычисление ряда:
n   Член ряда    Сумма
------------------------
0   1,000000     1,000000

========================================
РЕЗУЛЬТАТЫ:
Значение x: 0.5
Точность: 1.0
Количество членов ряда: 0
Сумма ряда: 1,00000000
Стандартная функция: 0,66666667
Разница: 0,33333333
========================================
```

Рисунок 3.2 – Попытка ввести плохую точность

Если же ввести не дробную точность, а целую, то результат вычислений будет просто не точным настолько, насколько большое число ввели. Чем число меньше – тем точнее вычисления.

```
ВЫЧИСЛЕНИЕ ФУНКЦИИ С ПОМОЩЬЮ РЯДА МАКЛОРЕНА
Функция: (1/(1+x)) * (1 - x + x^2 - x^3 + ...)
Ряд сходится при |x| < 1

Введите x (дробное число, например 0.5): втпо
Exception in thread "main" java.lang.NumberFormatException Create breakpoint : For input string: "втпо" <3 internal lines>
    at java.base/java.lang.Double.parseDouble(Double.java:971)
    at MainKt.main(Main.kt:11)
    at MainKt.main(Main.kt)
```

Рисунок 3.3 – Попытка ввести не число

В случае, если пользователь ввёл вместо числа что-угодно другое, программа прекратит свою работу, завершившись с ошибкой.

**5. Вывод:**

В данной работе был изучен синтаксис языка высокого уровня – Kotlin, написана программа, вычисляющая значение заданной функции путем разложения в ряд Маклорена с заданной точностью и с использованием стандартной функции класса Math. Аргумент функции x и точность вычислений вводится пользователем в консоль, вывод выполняется также в консоль.

ПРИЛОЖЕНИЕ А

Листинг программы:

```
import kotlin.math.pow

fun main() {
    println("ВЫЧИСЛЕНИЕ ФУНКЦИИ С ПОМОЩЬЮ РЯДА МАКЛОРЕНА")
    println("Функция: (1/(1+x)) = (1 - x + x^2 - x^3 + ...)")
    println("Ряд сходится при |x| < 1")
    println()

    // Ввод данных
    print("Введите x (дробное число, например 0.5): ")
    val x = readLine()!!.toDouble()

    print("Введите точность (маленькое число, например 0.001): ")
    val epsilon = readLine()!!.toDouble()

    // Проверка условия сходимости
    if (x >= 1 || x <= -1) {
        println("Ошибка: x должен быть между -1 и 1")
        return
    }

    // Вычисление ряда Маклорена
    var sum = 0.0
    var n = 0
    var term: Double
    var lastTerm: Double

    println("\nВычисление ряда:")
    println("n\tЧлен ряда\tСумма")
    println("------------------------")

    do {
        // Вычисляем n-й член ряда: (-1)^(n-1) * x^n
        term = (-1.0).pow(n) * x.pow(n)
        sum += term
        lastTerm = term

        println("$n\t${"%.6f".format(term)}\t${"%.6f".format(sum)}")
        n++

    } while (Math.abs(lastTerm) > epsilon) // Продолжаем пока член ряда
больше точности

    // Вычисление стандартной функции для сравнения
    val standard = 1.0 / (1.0 + x)  // Упрощенная функция для сравнения

    // Вывод результатов
    println("\n" + "=".repeat(40))
    println("РЕЗУЛЬТАТЫ:")
    println("Значение x: $x")
    println("Точность: $epsilon")
    println("Количество членов ряда: ${n-1}")
    println("Сумма ряда: ${"%.8f".format(sum)}")
    println("Стандартная функция: ${"%.8f".format(standard)}")

    val difference = Math.abs(sum - standard)
    println("Разница: ${"%.8f".format(difference)}")
    println("=".repeat(40))
}
```