

Лабораторная работа 3.

Организация циклов с контрольным выводом итераций

Цель

1. Знакомство с операторами циклов и итерационными процессами.
2. Организация пользовательских классов
3. Освоение вывода данных.

Постановка задачи

Согласно варианту задания, найдите сумму ряда с общим членом a_n и точностью $|a_n| \leq \varepsilon$. Организуйте вывод построчно на экран следующих значений: номера итерации, значения текущего члена ряда и суммы.

Контроль сходимости ряда должен осуществляться с помощью переменных целого типа: Stop1, Stop2, Stop3, в которые пользователь вводит значения контрольных итераций. После попадания на контрольную итерацию делать остановку вычислений. Спросить у пользователя о необходимости продолжения вычислений. Если пользователь согласился – продолжить цикл. Предусмотреть средства борьбы с возможным заикливанием приложения.

Рекомендации:

При составлении программы с типами переменных невысокой точности необходимо учесть возможность переполнения результатов вычислений, когда число выходит за пределы допустимые для описанного типа. Один из способов, позволяющий избежать такой ситуации – это производить вычисления последующего члена ряда через предыдущий. Для этого необходимо получить множитель по формуле:

$$k_n = \frac{a_{n+1}}{a_n}.$$

Член ряда a_0 считаем вручную, а затем вычисления члена ряда организовываем следующим образом:

$$\begin{aligned} a_1 &= k_0 \cdot a_0, \\ a_2 &= k_1 \cdot a_1, \\ &\dots\dots\dots \\ a_n &= k_{n-1} \cdot a_{n-1}, \\ a_{n+1} &= k_n \cdot a_n \end{aligned}$$

Более простой способ – это переход к данным повышенной точности, однако, он менее эффективен, чем предыдущий.

Итерационный цикл организуйте при помощи оператора цикла, считая, что необходимое количество итераций заранее неизвестно.

В цикле организовать проверку на попадание в контрольную итерацию. Вывести на экран вопрос о продолжении расчета, с вариантами ответа для пользователя. Предусмотреть возможность, как продолжения расчета, так и возможность досрочного выхода из цикла с помощью **return** или **break**.

Метод расчета последующего члена ряда через предыдущий организовать отдельным статическим классом. В него же включить отдельно метод вывода на экран расчетных данных и метод расчета суммы ряда с определенной точностью, согласно варианту.

Варианты задания:

1. $a_n = (-1)^{n-1} / n^n$, $\varepsilon = 10^{-25}$.
2. $a_n = 1/2^n + 1/3^n$, $\varepsilon = 2 \cdot 10^{-15}$.
3. $a_n = (2n-1) / 2^n$, $\varepsilon = 5 \cdot 10^{-16}$.
4. $a_n = 1 / ((3n+1)!)$, $\varepsilon = 5 \cdot 10^{-10}$.
5. $a_n = 10^n / n!$, $\varepsilon = 2 \cdot 10^{-17}$.
6. $a_n = (n!) / (2n)!$, $\varepsilon = 5 \cdot 10^{-8}$.
7. $a_n = n! / n^n$, $\varepsilon = 10^{-17}$.
8. $a_n = (-1)^n n! / (n^n)^2$, $\varepsilon = 2 \cdot 10^{-27}$.
9. $a_n = 3^n / (3n)!$, $\varepsilon = 5 \cdot 10^{-15}$.
10. $a_n = (n+2)! / (n^n)^2$, $\varepsilon = 10^{-26}$.

Контрольные вопросы для сдачи лабораторной работы:

1. Какие типы циклов имеются в языке Java?
2. Опишите параметры цикла **for**.
3. Как можно выйти из цикла?
4. Что такое переполнение и исчезновение числа, и как с этим бороться?
5. Как в Java реализуется возведение в степень?
6. Что такое статические классы и методы?