Институт информационных технологий и управления

в технических системах

Кафедра информационных технологий и компьютерных систем

ОТЧЕТ

по лабораторной работе № 7

**«ИССЛЕДОВАНИЕ ЦИКЛИЧЕСКИХ АЛГОРИТМОВ»**

по дисциплине «Программирование. Базовые процедуры обработки информации»

Выполнил студент группы ИВТ/б-11д

Орлов И.В.

Проверил доцент Петров И.И.

Севастополь

2020

**1. ЦЕЛЬ РАБОТЫ**

Целью данной работы является изучение назначения и приемов использования операторов циклов Java, исследование циклических алгоритмов и программ, применяемых для решения типовых задач вычисления значения функции по рекуррентной формуле при заданном значении аргумента и табулирования функции на заданном интервале.

2. **ПОСТАНОВКА ЗАДАЧИ**

1) Ознакомиться с теоретическими сведениями о рекуррентных формулах арифметических корней, приведенными в пункте 3.4 методических указаний, а также с возможностями построения циклических программ в Java.

2) Разработать методы для вычисления функции арифметического корня по рекуррентной формуле согласно варианту задания и для табулирования этой функции на заданном интервале.

Вариант задания V вычислен по формуле

V = (N%14 != 0) ? N%14 : 14; ,

где N – номер студента в списке группы:

1%14=1;

Данные варианта задания приведены в таблицах 2.1 и 2.2

Таблица 2.1 – Реализуемые функции

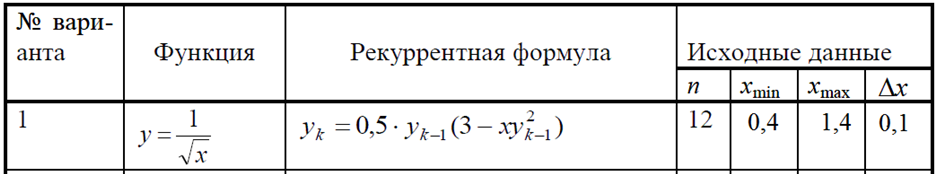
****

Таблица 2.2 – Виды циклов, подлежащих реализации

|  |  |  |
| --- | --- | --- |
| Номер  варианта | Вычисление корня по рекуррентной формуле | Табулирование функции |
| 1 | while | for |

**3. АНАЛИЗ ЗАДАЧИ**

Разрабатываемый класс (программа табулирования функции на заданном интервале) должен включать несколько методов.

*1) Первый метод* должен осуществлять вычисление значения функции (арифметического корня) средствами класса Math.

*2) Второй метод* должен осуществлять вычисление арифметического корня по рекуррентной формуле для заданного значения *х*.

Метод реализует циклический алгоритм, в теле которого пересчитывается по рекуррентной формуле значение  при *k* = 1,2,3,…,n. Согласно теории рекуррентных вычислений с ростом *k* значение  приближается к точному значению . В программе пересчет  должен начаться с  (н**ачальное приближение: )** и повториться *n* раз. Значения *x* и *n* должны передаваться в метод в качестве параметров. На каждом шаге цикла должен осуществляться вывод  и ошибки вычисления по рекуррентной формуле .

Метод нужно запустить на выполнение при двух значениях *х* из диапазона , заданного в таблице 2.1.

Для каждого значения *х* необходимопроанализировать полученные значения последовательных приближений , вычисленных по рекуррентной формуле, к точному значению *у*. Сколько итераций потребовалось для получения значения , равного *y* с точностью до 6 знака? От чего зависит значение необходимого числа итераций? **Если вычисления по рекуррентной формуле не сходятся к конкретному значению после заданного числа итераций, то возможно, что используемая рекуррентная формула не подходит для заданного интервала. Тогда нужно попробовать подобрать другой интервал или уточнить рекуррентную формулу**

*3)Третий метод* должен осуществлять табулирование функции арифметического корня на заданном интервале.

Под табулированием функции понимают вычисление значения функции при всех значениях аргумента *х*, изменяющихся с шагом в пределах от  до .

Для вычисления значения функции третий метод будет вызывать первый метод и второй метод. Вывод промежуточных значений  во втором методе предлагается исключить (***закомментировать***). Третий метод должен осуществлять вывод точного значения функции  (считаем точным значение функции, вычисленное средствами модуля Math) и приближенного

значения  (вычисленного по рекуррентной формуле), а также ошибки  для каждого значения *х*.

4) Четвертый метод − main() − должен осуществлять табулирование функции последовательно на нескольких интервалах с различным значением . Для выполнения табулирования он должен дважды вызвать *третий метод. В первый раз в вызываемый метод нужно передать* Запустите значения *п*, , , , заданные в таблице 2.1, во второй раз – другие значения, чтобы показать, как будет работать (и будет ли работать) рекуррентная формула для того или иного интервала.

**Следует отметить, что в выражениях, в которых используются операции над вещественными значениями, нельзя смешивать различные типы данных, иначе можно получить неправильный результат.**

**4. ТЕСТОВЫЕ ПРИМЕРЫ И РЕЗУЛЬТАТЫ ИХ ОБРАБОТКИ ВРУЧНУЮ**

*В данном пункте сначала нужно привести результаты расчета на калькуляторе (или с помощью табличного процессора Excel) значений заданной функции вычисления корня* ***в двух точках, принадлежащих заданному интервалу*** *(тех точках, на которых мы будем проверять работу метода вычисления приближенного значения функции (funIter)).*

*Затем нужно привести результаты табулирования функции на заданном интервале с заданным шагом, полученные с помощью табличного процессора Excel. Если при заданном шаге точек, в которых вычисляется значение функции, окажется мало, шаг нужно уменьшить. Например:*

Результаты табулирования функции  на интервале [10, 100] с шагом 10 изображены на рисунке 4.1.



Рисунок 4.1 – Результаты табулирования

функции  на интервале [10, 100] с шагом 10

График функции , построенный по результатам табулирования функции на интервале *[10, 100]* с шагом *10* приведен на рисунке 4.2. График построен при помощи табличного процессора Excel.



Рисунок 3.8 – График функции , построенный по

результатам табулирования на интервале *[10,100]* с шагом 10

**5. СХЕМЫ АЛГОРИТМОВ**

В приведенных ниже алгоритмах использованы виды циклов, заданные таблицей 2.2. Схема алгоритма вычисления корня по рекуррентной формуле изображена на рисунке 5.1. Схема алгоритма табулирования функции изображена на рисунке 5.2.

Рисунок 5.1 – Схема алгоритма вычисления корня

по рекуррентной формуле

Рисунок 5.2 – Схема алгоритма табулирования функции

**6.ТЕКСТ ПРОГРАММЫ**

Согласно пункту 2.8.1. методических указаний к лабораторным работам, текст программы должен быть отформатирован следующим образом:

1. шрифт: arial, 12 пт, полужирный;
2. абзац:

* выравнивание по левому краю;
* уровень – основной текст;
* отступ слева – 0;
* отступ справа – 0;
* интервал перед – 0;
* интервал после – 0;
* первая строка – нет (отступа или выступа),
* междустрочный интервал – множитель, значение – 1,2;

1. комментарии (обязательно должны присутствовать в тексте программы) выделить курсивом и синим цветом.

**7. СВЕДЕНИЯ ОБ ОТЛАДКЕ ПРОГРАММЫ И ПРОВЕРКЕ**

**ЕЕ РАБОТОСПОСОБНОСТИ**

Ниже приведены результаты запуска метода funIter () для двух точек из заданного интервала. Результаты вызова метода funIter (*10.0,10*) приведены на рисунке 7.1. Приближенное значение функции с точностью до шестого знака было вычислено на *пятой* итерации. Результаты вызова метода funIter (*15.0,10*) приведены на рисунке 7.2. Приближенное значение функции с точностью до шестого знака было вычислено на *пятой* итерации.



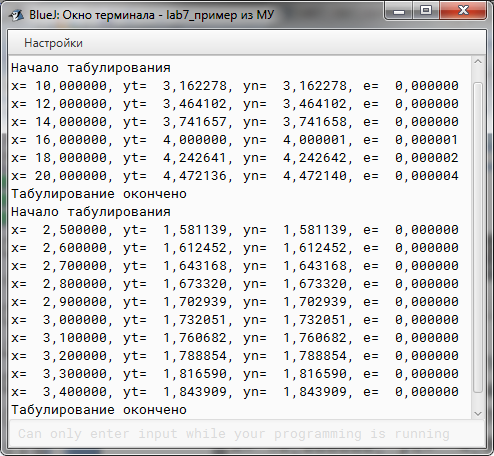
*Результаты должны быть представлены именно в такой форме*

Рисунок 7.1 − Результаты вызова метода funIter (*10.0,10*)

Рисунок 7.2 − Результаты вызова метода funIter (*15.0,10*)

Для точек из заданного интервала метод funIter () выдал ожидаемый результат, что позволяет сделать вывод о его работоспособности.

Результаты выполнения программы приведены на рисунке 7.3 (без вывода приближений, полученных в методе funIter(), – соответствующий оператор вывода «закомментирован»).



*Вставить свои результаты*

Рисунок 7.3 – Результаты выполнения программы

При табулировании функции на заданном интервале методы вычисления заданной функции средствами класса Math и с помощью итерационной формулы выдали ожидаемый результат, что позволяет сделать вывод о работоспособности программы. *Далее следует прокомментировать результаты второго вызова метода табулирования функции (для другого интервала).*

**ВЫВОД**

В ходе выполнения лабораторной работы были изучены операторы цикла языка Java: for, while и do while, а также разработана программа табулирования функции, вызывающая следующие методы:

* метод вычисления функции корня в заданной точке с помощью функции класса Math;
* метод приближенного вычисления функции корня в заданной точке с помощью рекуррентной формулы, действительной для заданного интервала (использовался цикл *while*);
* метод табулирования функции на заданном интервале (использовался цикл *for*).

**В методах осуществляется передача параметров по значению, т.к. все параметры принадлежат простым типам.**