Министерство науки и высшего образования РФ

ФГАОУ ВО Пермский национальный исследовательский

политехнический университет

Кафедра «Вычислительная математика, механика и биомеханика»

Отчет по лабораторной работе № 6

Вариант 13

тема «Циклы»

по дисциплине «Информатика»

Выполнил: студент группу ИСТ-22-1б Ипанов М.В.

Проверил: доцент каф. ВММБ Каменских А.А.

Пермь, 2022

**Содержание**

[Задание 1 3](#_Toc123992370)

[1.1. Постановка задачи 3](#_Toc123992371)

[1.2. Решение задачи, код программы 3](#_Toc123992372)

[1.3. Тестирование работы программы 4](#_Toc123992373)

[Задание 2 5](#_Toc123992374)

[2.1. Постановка задачи 5](#_Toc123992375)

[2.2. Решение задачи. Код программы 5](#_Toc123992376)

[2.3. Тестирование работы программы 6](#_Toc123992377)

[Задание 3 7](#_Toc123992378)

[3.1. Постановка задачи 7](#_Toc123992379)

[3.2. Решение задачи. Код программы 7](#_Toc123992380)

[3.3. Проверка решения 8](#_Toc123992381)

[Задание 4 9](#_Toc123992382)

[4.1. Постановка задачи 9](#_Toc123992383)

[4.2. Решение задачи. Код программы 9](#_Toc123992384)

[4.3. Проверка решения 10](#_Toc123992385)

[Задание 5 11](#_Toc123992386)

[5.1. Постановка задачи 11](#_Toc123992387)

[5.2. Решение задачи. Код программы 12](#_Toc123992388)

[5.3. Проверка решения 13](#_Toc123992389)

# Задание 1

## 1.1. Постановка задачи

Найти сумму первых N членов ряда и найти сумму членов ряда, которые меньше заданного с клавиатуры числа M:

**

## 1.2. Решение задачи, код программы

System.*out*.println("Введите x, n, m: ");  
d\_x = in.nextDouble();  
i\_n = in.nextInt();  
d\_m = in.nextDouble();  
System.*out*.println("Сумма первых n чисел = " + *num1\_1*(d\_x, i\_n));  
System.*out*.println("Сумма первых n чисел, которые меньше m = " + *num1\_2*(d\_x, d\_m));

public static double num1\_1(double x, int n) {  
 int i;  
 double n\_chlen, S = 0;  
 i = 1;  
 while (i <= n) {  
 n\_chlen = *pow*(-1, i - 1) \* *pow*(x, 2 \* i - 1) / (3 \* i - 1);  
 i += 1;  
 S += n\_chlen;  
 }  
 return S;  
}  
  
public static double num1\_2(double x, double m) {  
 int i;  
 double n\_chlen, S = 0;  
 i = 1;  
 while (true) {  
 n\_chlen = *pow*(-1, i - 1) \* *pow*(x, 2 \* i - 1) / (3 \* i - 1);  
 S += n\_chlen;  
 if (S < m) {  
 break;  
 }  
 i += 1;  
  
 if (i >= 10\_000) {  
 return 0;  
 }  
 }  
 return S;  
}

## 1.3. Тестирование работы программы

Таблица 1

Тестирование работы программы и проверка результатов решения

|  |  |  |
| --- | --- | --- |
| № п.п. | Решение Java | Проверка |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |

Сравнение решения в Java и перепроверке MS Excel показали правильность написанного кода.

# Задание 2

## 2.1. Постановка задачи

Дано натуральное число . Определить, сколько раз в нем встречается, введенная с клавиатуры, цифра . Вывести сообщение, является ли  максимальной цифрой в числе (например, дано число 123895, ввели с клавиатуры цифру 3, в нашем числе цифра 3 встречается 1 раз, цифра 3 не является ни максимальной, ни минимальной цифрой в числе 123895).

## 2.2. Решение задачи. Код программы

System.*out*.println("Введите натуральное число и k: ");  
i\_x = in.nextInt();  
i\_k = in.nextInt();  
System.*out*.println("k встречается в числе " + *num2\_1*(i\_x, i\_k) + " раз");  
System.*out*.println("k " + (*num2\_2*(i\_x, i\_k) ? "Является" : "НЕ является") + " максимальной цифрой в числе");

public static int num2\_1(int x, int k) {  
 boolean flag;  
 int x\_n, count = 0, answer = 1;  
  
 while (x > 0) {  
 x\_n = x % 10;  
 x /= 10;  
 if (x\_n == k) {  
 count += 1;  
 }  
 }  
 return count;  
}  
  
public static boolean num2\_2(int x, int k) {  
  
 boolean flag = true;  
 int x\_n;  
  
 while (x > 0) {  
 x\_n = x % 10;  
 x /= 10;  
 if (x\_n > k) {  
 flag = false;  
 }  
 }  
 return flag;  
}

## 2.3. Тестирование работы программы

Для проверки решения задача с некоторыми входными данными была прорешена в MS Excel и составлена таблица соответствия решения в программе:

Таблица 1

Тестирование работы программы и проверка результатов решения

|  |  |  |
| --- | --- | --- |
| № п.п. | Решение Java | Проверка |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |

При тестировании программы на некоторых реальных и произвольных данных при правильном вводе данных ошибок выявлено не было. Сравнение ручной проверки и ответа, который выдал код показывают правильность написанного кода.

# Задание 3

## 3.1. Постановка задачи

Написать программы, которые вычисляют выражения:



## 3.2. Решение задачи. Код программы

System.*out*.println("Введите i (в задаче i = 8 или 5): ");  
i\_i = in.nextInt();  
System.*out*.println("Ответ 1) = " + *num3\_1*(i\_i));  
System.*out*.println("Ответ 2) = " + *num3\_2*(i\_i));  
System.*out*.println("Ответ 3) = " + *num3\_3*(i\_i));

public static double num3\_1(int in) {  
 double S = 0;  
 for (int i = 1; i <= in; i++) {  
 for (int j = 1; j <= i; j++) {  
 S += *pow*(i - j, 1.0 / 3.0);  
 }  
 }  
 return S;  
}  
  
public static double num3\_2(int in) {  
 double p = 1;  
 for (int i = 1; i <= in; i++) {  
 for (int j = 1; j <= i; j++) {  
 p \*= *pow*(j, i);  
 }  
 }  
 return p;  
}  
  
public static double num3\_3(int in) {  
 double S, p = 1;  
 for (int i = 1; i <= in; i++) {  
 S = 0;  
 for (int j = i; j <= 2 \* i - 1; j++) {  
 for (int k = i + j; k <= 2 \* (i + j); k++) {  
 S += (3 \* i - 2 \* (k + 0.2 \* j));  
 }  
 }  
 p \*= S;  
 }  
 return p;  
}

## 3.3. Проверка решения

Таблица 1

Тестирование работы программы и проверка результатов решения

|  |  |  |
| --- | --- | --- |
| № п.п. | Решение Java | Проверка |
| 1 |  |  |
| 2 |  |  |
| 3 |  | **Вроде верно** |

Тестирование работы программы и проверка результатов решения

В ходе небольшого количества проверок никаких ошибок выявлено не выявлено, программа работала корректно со всеми контрольными данными.

# Задание 4

## 4.1. Постановка задачи

Пользователь вводит целое десятичное число. Написать программу перевода целого десятичного числа из десятичной системы счисления в систему счисления с основанием, которое пользователь вводит с клавиатуры (вводим ограничения на системы счисления, в которых есть буквенное обозначение цифр, так же не может быть введено 0 и 1 как основание системы счисления). После перевода сделать проверку. Пользователь может выбирать систему счисления до бесконечности, т.е. необходимо предусмотреть внешний цикл с вопросом к пользователю о необходимости продолжать перевод из одной системы счисления в другую.

## 4.2. Решение задачи. Код программы

System.*out*.println("Введите x и основани СС в которую нужно перевести: ");  
i\_x = in.nextInt();  
i\_k = in.nextInt();  
if (i\_k >= 2 && i\_k <= 10) {  
 i\_x = *num4\_2*(i\_x, i\_k);  
 System.*out*.println("Ответ - " + i\_x);  
 System.*out*.println("Проверка - " + *num4\_3*(i\_x, i\_k));  
} else {  
 System.*out*.println("error");  
}  
  
public static int num4\_2(int x, int k) {  
 String S = "";  
 int x\_s;  
 while (x > 0) {  
 x\_s = x % k;  
 x /= k;  
 S = S + x\_s;  
 }  
 S = *reverseString*(S);  
 x = Integer.*parseInt*(S);  
 return x;  
}  
  
public static String reverseString(String str) {  
 char[] array = str.toCharArray();  
 String result = "";  
 for (int i = array.length - 1; i >= 0; i--) {  
 result = result + array[i];  
 }  
 return result;  
}  
  
public static int num4\_3(int x, int k) {  
 int x\_s, i = 0, S = 0;  
 while (x > 0) {  
 x\_s = x % 10;  
 x /= 10;  
 S += x\_s \* *pow*(k, i);  
 i += 1;  
 }  
 return S;  
}

## 4.3. Проверка решения

Таблица 1

Тестирование работы программы и проверка результатов решения

|  |  |  |
| --- | --- | --- |
| № п.п. | Решение Java | Проверка MS Excel |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |

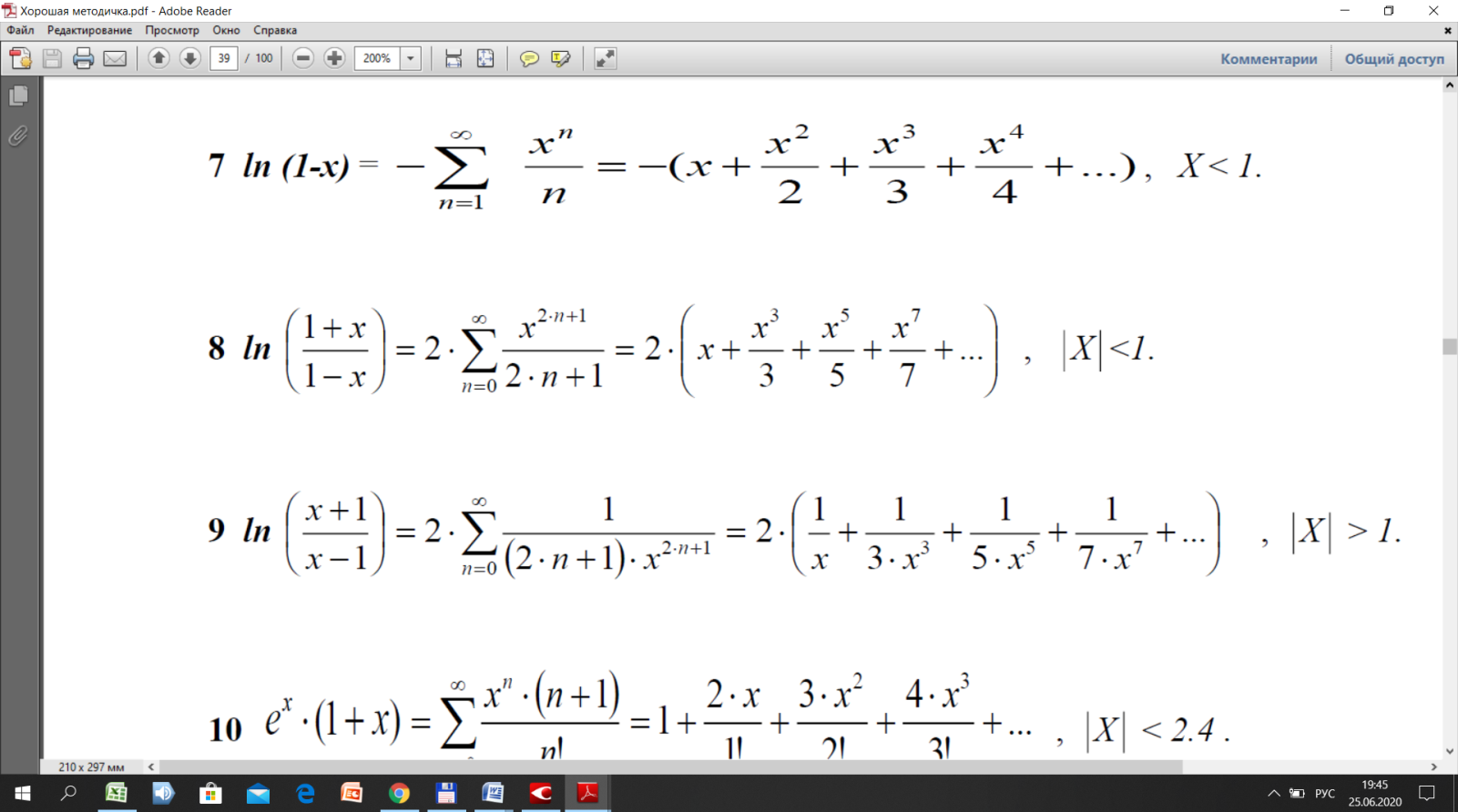
Тестирование работы программы и проверка результатов решения

В ходе небольшого количества проверок никаких ошибок выявлено не выявлено, программа работала корректно со всеми контрольными данными.

# Задание 5

## 5.1. Постановка задачи

Организовать цикл для нахождения функции , через ряд Маклорена с погрешностью 10-2 – 10-4, остановка итерационной процедуры , где  – это погрешность. Вывести значение суммы ряда и итоговую погрешность для всех вариантов остановки итерационной процедуры.



## 5.2. Решение задачи. Код программы

System.*out*.println("Введите неизвестную (-1 < x < 1): ");  
d\_x = in.nextDouble();  
if (*abs*(d\_x) >= 1) {  
 System.*out*.println("error");  
 break;  
}  
*num5\_1*(d\_x);

public static double num5\_1(double x) {  
 double mistake\_st = 0.0001, mistake\_end = 0.01;  
 double step, difference, difference\_st, S = 0, ln = *log*((1 + x) / (1 - x));  
 difference\_st = *abs*(*log*((1 + x) / (1 - x)) - 2 \* x);  
 int n = 0;  
 boolean flag = false;  
 while (true) {  
 step = *pow*(x, 2 \* n + 1) / (2 \* n + 1);  
 S += 2 \* step;  
 difference = *abs*(ln - S);  
 if (flag == false) {  
 if (difference >= mistake\_st & difference <= mistake\_end) {  
 flag = true;  
 System.*out*.println("Шаг: " + n + " S = " + S + " (E) = " + difference);  
 }  
 } else {  
 if (difference >= mistake\_st) {  
 System.*out*.println("Шаг: " + n + " S = " + S + " (E) = " + difference);  
 } else {  
 break;  
 }  
 }  
 if (n > 1000 & flag == false) {  
 System.*out*.println("Значение не найдено. Не попало в данную погрешность измерения (" + difference\_st + " <=> " + difference + ")");  
 break;  
 }  
 n += 1;  
 }  
 return S;  
}

## 5.3. Проверка решения

Проверка проводилась с помощью сайта <https://mathsolver.microsoft.com>

Который является универсальным калькулятором.

Тестирование работы программы и проверка результатов решения

Таблица 1

|  |  |  |
| --- | --- | --- |
| № п.п. | Решение Java | Проверка |
| 1 |  | Так и задумано |
| 2 |  |  |
| 3 |  |  |

При тестировании программы на некоторых реальных и произвольных данных при правильном вводе данных ошибок выявлено не было. Сравнение проверки и ответа, который выдал код показывают правильность написанного кода.

Конец!