

Министерство науки и высшего образования Российской Федерации Федеральное
государственное бюджетное образовательное учреждение высшего образования
«Алтайский государственный технический университет им. И. И. Ползунова

Факультет информационных технологий

Кафедра прикладной информатики в области экономики

Отчет защищен с оценкой _____

Преподаватель С. В. Умбетов

«» _____ 2024 г.

Отчёт по лабораторной работе №2

по дисциплине «Алгоритмизация и Программирование»

«Линейный вычислительный процесс»

Студент группы _____ ПИЭ-22 _____ Н. А. Горкун

Группа _____ и., о., фамилия

Преподаватель _____ ассистент, к. т. н. _____ С. В. Умбетов

Должность, учетная степень _____ и.,о., фамилия

Барнаул 2024

Цели и задачи работы: изучение циклических алгоритмов, операторов цикла, программирование циклического вычислительного процесса.

Задание к работе: реализовать циклический вычислительный процесс.
Самостоятельно решить задачи в соответствии с индивидуальным вариантом

Задание принял:



Горкун Н. А.

Подпись

ФИО

Ход работы

Задание 1: дано целое число n и набор из n вещественных чисел. Вывести в том же порядке округленные значения всех чисел из данного набора (как целые числа), а также сумму всех округленных значений

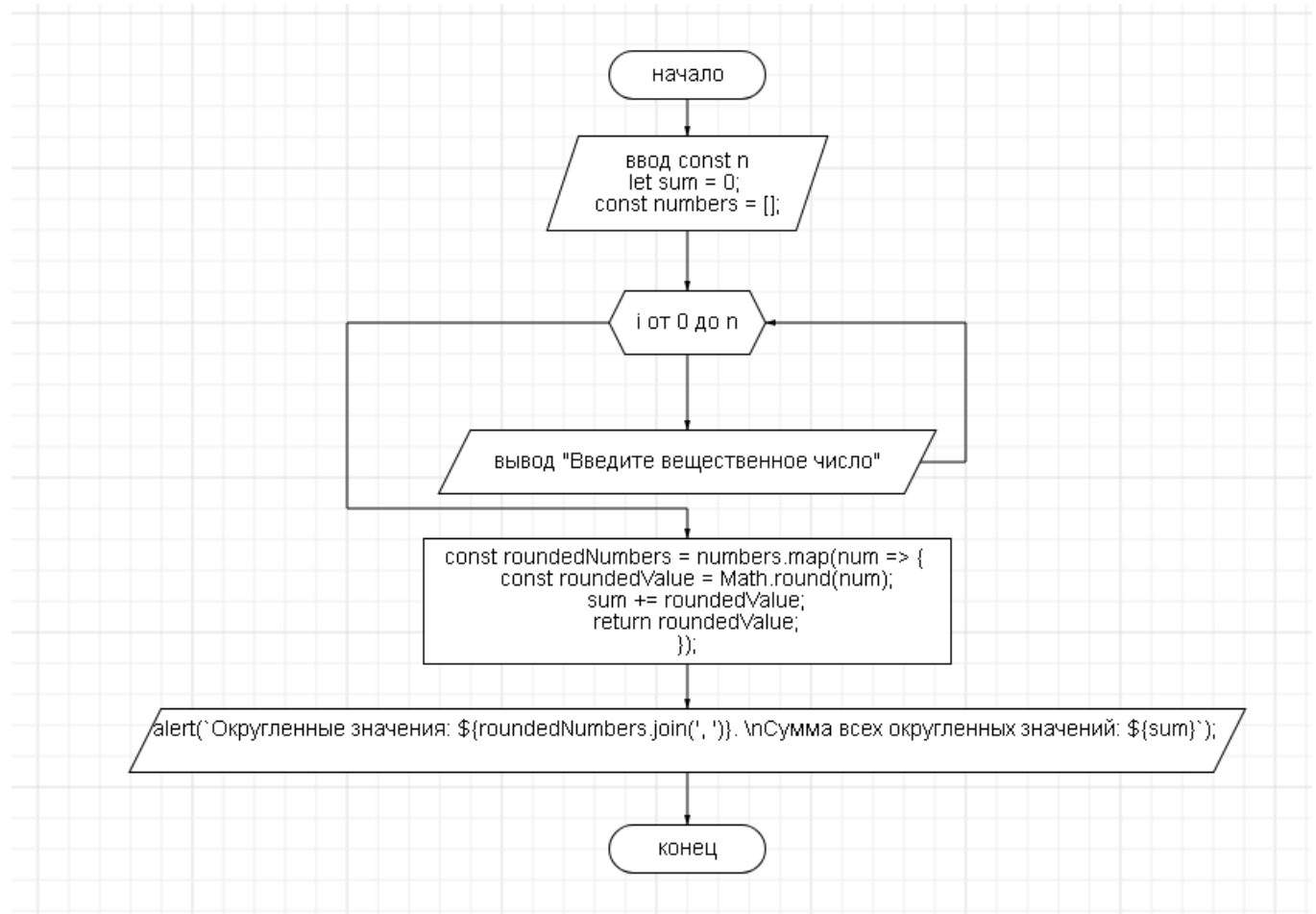


Рисунок 1 – Блок-схема выполнения кода программы

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4
5  </head>
6  <body>
7  <script>
8
9  const n = Number(prompt("Введите целое число n: "));
10 let sum = 0;
11
12 const numbers = [];
13 for (let i = 0; i < n; i++) {
14   numbers.push(Number(prompt(`Введите вещественное число ${i + 1}:`)));
15 }
16
17 const roundNumbers = numbers.map(num => {
18   const roundedValue = Math.round(num);
19   sum += roundedValue;
20   return roundedValue;
21 });
22
23 alert(`Округленные значения: ${roundNumbers.join(', ')}. \n Сумма всех округленных значений: ${sum}`);
24
25 </script>
26 </body>
27 </html>

```

Рисунок 2 – Код программы

Проведем тестирование написанной программы и проверим работу Visual Studio с помощью Paint. Ниже представлено сравнение работы консоли и результатов в Paint.

Результаты работы программы Visual Studio		Проверка результатов в Paint	
Входные данные	Выходные данные	Входные данные	Выходные данные
2;1.99;2.22	2,2,4	2;1.99;2.2 2	2,2,4
3;2.56;5.12;4.99	3;5;5;13	3;2.56;5.1	3;5;5;13
5;1.12;2.99;3.33; 4.44;5.55	1;3;3;4;5;17	5;1.12;2.9 9;3.33;4.4 4;5.55	1;3;3;4;5;17

Таблица 1 – Проверка результатов

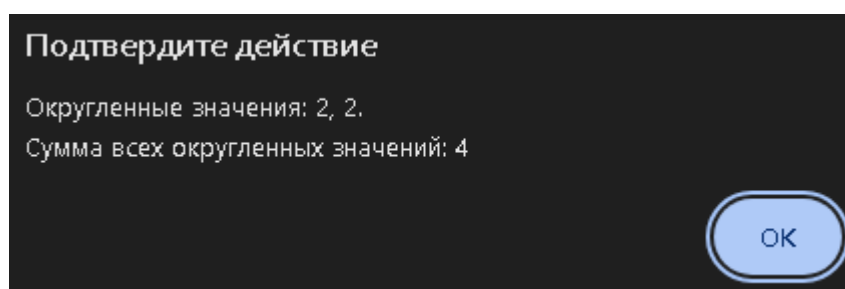


Рисунок 3 – Результат работы программы

1,99; 2,22
 $2 + 2 = 4$

Рисунок 4 – Проверка в Paint

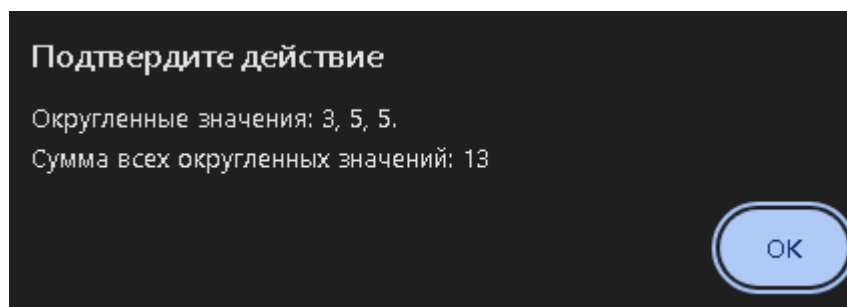


Рисунок 5 – Результат выполнения работы программы

2,56; 5,12; 4,99
 $3 + 5 + 5 = 13$

Рисунок 6 – Проверка в Paint

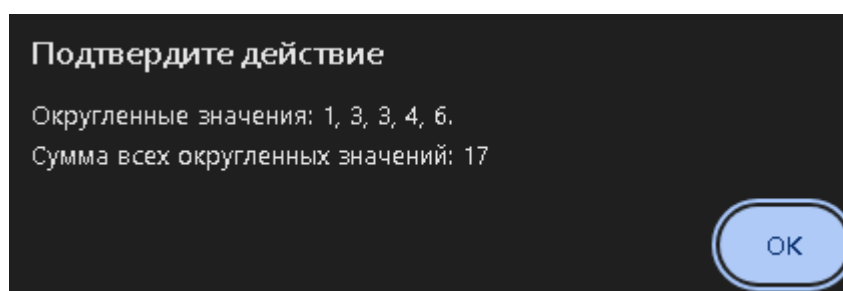


Рисунок 7 – Результат выполнения работы программы

1,12; 2,99; 3,33; 4,44; 5,55
 $1 + 3 + 3 + 4 + 6 = 17$

Рисунок 8 – Проверка в Paint

Задание 2: дано вещественное число цена 1 кг конфет вывести стоимость 1.2, 1.4, ..., 2 кг конфет.

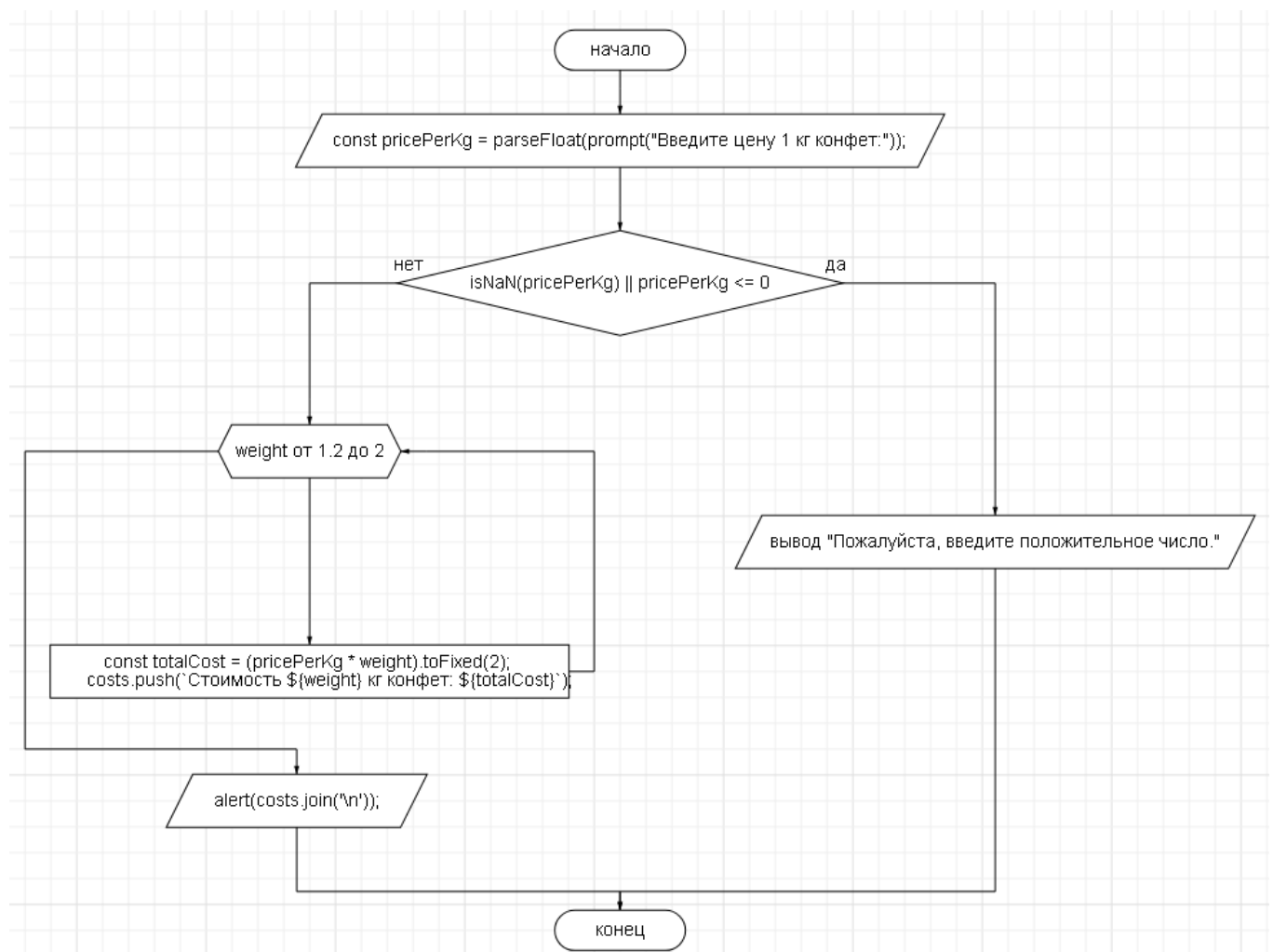


Рисунок 9 – Блок-схема выполнения кода программы

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4
5  </head>
6  <body>
7  <script>
8
9  const pricePerKg = parseFloat(prompt('Введите цену 1 кг конфет: '));
10 if (isNaN(pricePerKg) || pricePerKg <= 0) {
11     alert('Пожалуйста, введите положительное число.')
12 } else {
13     let costs = [];
14     for (let weight = 1.2; weight <=2; weight += 0.2) {
15         const totalCost = (pricePerKg * weight).toFixed(2);
16         costs.push(`Стоимость ${weight} кг конфет: ${totalCost}`);
17     }
18     alert(costs.join('\n'));
19 }
20
21 </script>
22 </body>
23 </html>

```

Рисунок 9 – Код программы

Проведем тестирование написанной программы и проверим работу Visual Studio с помощью Excel. Ниже представлено сравнение работы консоли и результатов в Excel.

Результаты работы программы Visual Studio		Проверка результатов в Excel	
Входные данные	Выходные данные	Входные данные	Выходные данные
10	Стоимость 1.2 кг конфет: 12.00 Стоимость 1.4 кг конфет: 14.00 Стоимость 1.5999999999999999 кг конфет: 16.00 Стоимость 1.7999999999999998 кг конфет: 18.00 Стоимость 1.9999999999999998 кг конфет: 20.00	10	Стоимость 1.2 кг конфет: 12.00 Стоимость 1.4 кг конфет: 14.00 Стоимость 1.5999999999999999 кг конфет: 16.00 Стоимость 1.7999999999999998 кг конфет: 18.00 Стоимость 1.9999999999999998 кг конфет: 20.00
20	Стоимость 1.2 кг конфет: 24.00 Стоимость 1.4 кг конфет: 28.00 Стоимость 1.5999999999999999 кг конфет: 32.00 Стоимость 1.7999999999999998 кг конфет: 36.00 Стоимость 1.9999999999999998 кг конфет: 40.00	20	Стоимость 1.2 кг конфет: 24.00 Стоимость 1.4 кг конфет: 28.00 Стоимость 1.5999999999999999 кг конфет: 32.00 Стоимость 1.7999999999999998 кг конфет: 36.00 Стоимость 1.9999999999999998 кг конфет: 40.00
30	Стоимость 1.2 кг конфет: 36.00 Стоимость 1.4 кг конфет: 42.00 Стоимость 1.5999999999999999 кг конфет: 48.00 Стоимость 1.7999999999999998 кг конфет: 54.00 Стоимость 1.9999999999999998 кг конфет: 60.00	30	Стоимость 1.2 кг конфет: 36.00 Стоимость 1.4 кг конфет: 42.00 Стоимость 1.5999999999999999 кг конфет: 48.00 Стоимость 1.7999999999999998 кг конфет: 54.00 Стоимость 1.9999999999999998 кг конфет: 60.00

Таблица 2 – Проверка результатов

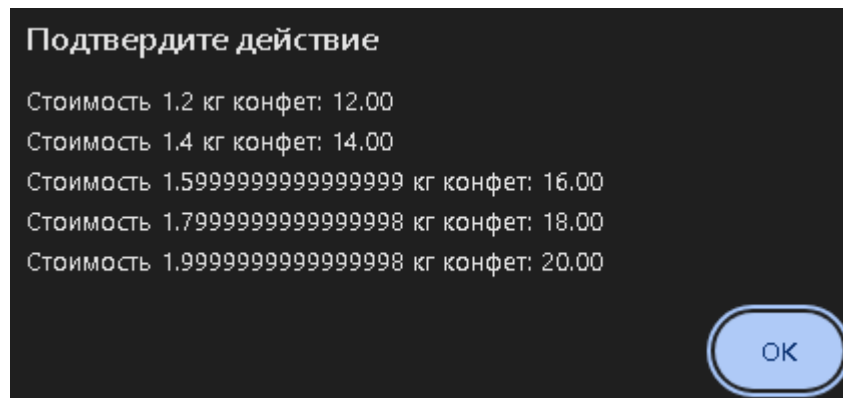


Рисунок 10 – Результат выполнения программы

	H	I
	10	12
		14
		16
		18
		20

Рисунок 11 – Проверка в Excel

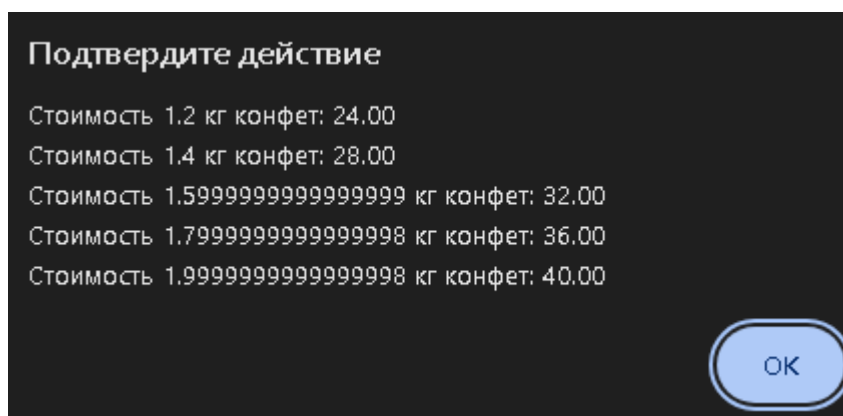


Рисунок 12 – Результат выполнения кода программы

	H	I
	20	24
		28
		32
		36
		40

Рисунок 13 – Проверка в Excel



Рисунок 14 – Результат выполнения кода программы

Н	I
30	24
	42
	48
	54
	60

Рисунок 15 – Проверка в Excel

Вывод: в ходе данной работы я использовал цикл `for` для перебора значений в нужном мне диапазоне. Также мне понадобилась `isNaN` — это функция в JavaScript, которая определяет, является ли переданное значение "не числом" (Not-a-Number). Функция `isNaN` возвращает `true`, если ее аргумент не может быть преобразован в число, в противном случае она возвращает `false`. Еще необходимо было наличие `toFixed(2)` - это метод, который используется в JavaScript для округления числа до определенного количества знаков после запятой. В скобках вы указываете количество знаков после запятой, до которого вы хотите округлить число, в моем случае до двух.