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 Информационных систем в экономике

Отчет защищен с оценкой \_\_\_\_\_\_\_\_

Преподаватель \_\_\_\_\_*С. В. Умбетов*

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Отчёт по лабораторной работе №2

по дисциплине «Алгоритмизация и Программирование»

«Основы работы с языком JS»

**09.03.03**

**Студент группы ПИЭ-23 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Милованов Д. А.**

**Преподаватель ассистент к. т. н.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С. В. Умбетов**

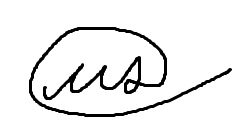
**Барнаул 2023**

**Лабораторная работа №1**

**Основы работы с языком JS**

**Цели и задачи работы:** познакомиться с основами JS, разобраться с принципами построения программы, алгоритмизации и запуска программ.

**Задание к работе:** написать программу для решения простейших математических задач. Программ должна выполнять операции: сложения, вычитания, умножения, деления и нахождения остатка от целочисленного деления.



**Задание принял:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Милованов Д. А.

Составим блок-схему, отображающую работу программы.



Рисунок 1 – Блок-схема работы калькулятора

Напишем программу в IDE VS Code.

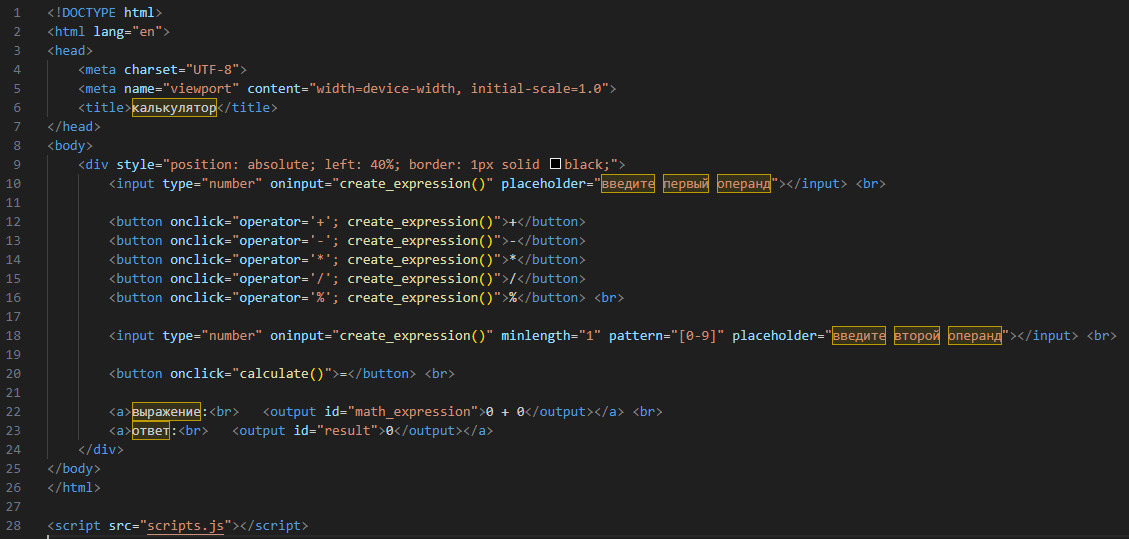
Рисунок 2 – HTML программы

Рисунок 3 – JS-код программы

Тестирование программы будет проводиться при помощи Python idle.

Таблица 1 – тестирование операции сложения.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| калькулятор | | Python idle | | соответствие |
| Входные данные | Выходные данные | Входные данные | Выходные данные |  |
| 0+1 | 1 | 0+1 | 1 | + |
| 1+14 | 15 | 1+14 | 15 | + |
| 0+0 | 0 | 0+0 | 0 | + |
| -1+2 | 1 | -1+2 | 1 | + |
| 100+100 | 200 | 100+100 | 200 | + |

Таблица 2 – тестирование операции вычитания.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| калькулятор | | Python idle | | соответствие |
| Входные данные | Выходные данные | Входные данные | Выходные данные |  |
| 1-1 | 0 | 1-1 | 0 | + |
| 0-1 | -1 | 0-1 | -1 | + |
| 12-5 | 7 | 12-5 | 7 | + |
| -1--1 | 0 | -1--1 | 0 | + |
| -1-23 | -24 | -1-23 | -24 | + |

Таблица 3 – тестирование операции умножения.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| калькулятор | | Python idle | | соответствие |
| Входные данные | Выходные данные | Входные данные | Выходные данные |  |
| -2 \* -2 | 4 | -2\*-2 | 4 | + |
| 14\*14 | 196 | 14\*14 | 196 | + |
| 0\*0 | 0 | 0\*0 | 0 | + |
| 0 \* 19 | 0 | 0 \* 19 | 0 | + |
| 5 \* -5 | -25 | 5 \* -5 | -25 | + |

Таблица 4 – тестирование операции деления.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| калькулятор | | Python idle | | соответствие |
| Входные данные | Выходные данные | Входные данные | Выходные данные |  |
| 5/-5 | -1 | 5/-5 | -1.0 | + |
| 1/0 | infinity | 1/0 | ZeroDivisionError | - |
| 0/0 | NaN | 0/0 | ZeroDivisionError | - |
| 0/1 | 0 | 0/1 | 0.0 | + |
| 10/100 | 0.1 | 10/100 | 0.1 | + |

Таблица 5 – тестирование операции нахождения остатка от деления.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| калькулятор | | Python idle | | соответствие |
| Входные данные | Выходные данные | Входные данные | Выходные данные |  |
| 10%100 | 10 | 10%100 | 10 | + |
| 5%3 | 2 | 5%3 | 2 | + |
| 1%0 | NaN | 1%0 | ZeroDivisionError | - |
| 0%1 | 0 | 0%1 | 0 | + |
| 248%-15 | 8 | 248%-15 | -7 | - |

**Результаты работы программы и данные для проверки:**

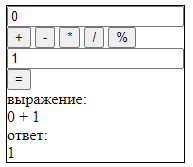


Рисунок 4



Рисунок 5

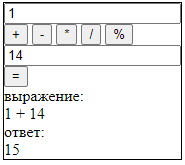


Рисунок 6



Рисунок 7

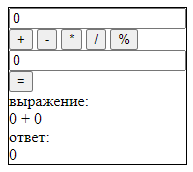


Рисунок 8



Рисунок 9

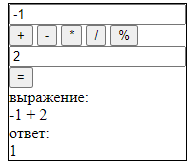


Рисунок 10



Рисунок 11

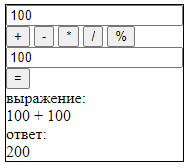


Рисунок 12



Рисунок 13

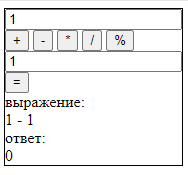


Рисунок 14



Рисунок 15

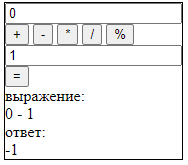


Рисунок 16



Рисунок 17

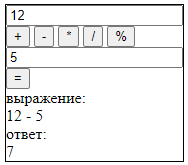


Рисунок 18



Рисунок 19

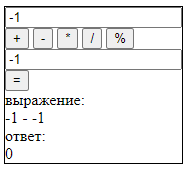


Рисунок 20



Рисунок 21

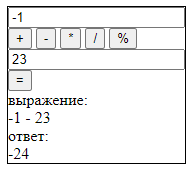


Рисунок 22



Рисунок 23

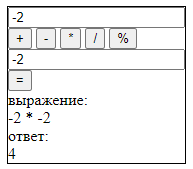


Рисунок 24



Рисунок 25

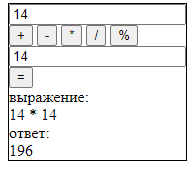


Рисунок 26



Рисунок 27

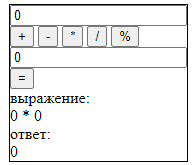


Рисунок 28



Рисунок 29

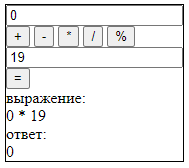


Рисунок 30



Рисунок 31

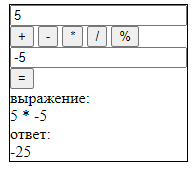


Рисунок 32



Рисунок 33

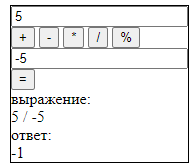


Рисунок 34



Рисунок 35

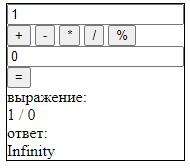


Рисунок 36

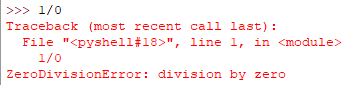


Рисунок 37

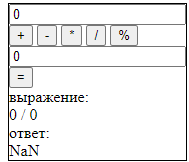


Рисунок 38

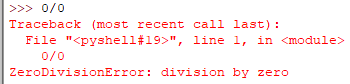


Рисунок 39

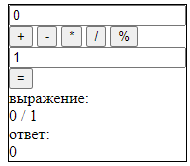


Рисунок 40



Рисунок 41

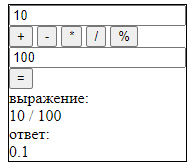


Рисунок 42



Рисунок 43

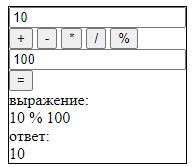


Рисунок 44



Рисунок 45

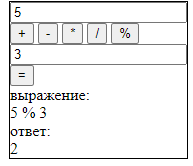


Рисунок 46



Рисунок 47

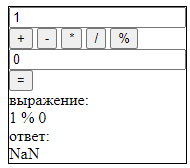


Рисунок 48

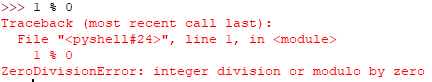


Рисунок 49

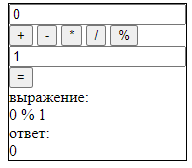


Рисунок 50



Рисунок 51

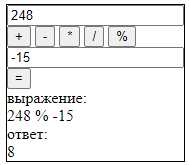


Рисунок 52



Рисунок 53

**Вывод:**

В ходе выполнения работы были изучены основы синтаксиса JS, стало известно, что остаток от целочисленного деления в JS считается по модулю. Значимой проблемой стало то, что JS при ошибке просто “молча перестаёт работать”, т.е. ломается, но не выдаёт ошибку из-за чего приходится постепенно откатывать изменения, отслеживая работоспособность кода, чтобы понять, в какой момент ошибка появилась в коде, только после этого, зная, в какой момент всё пошло не так, можно попытаться найти и исправить ошибку, скорее всего, эту проблему можно решить, используя отладчик, но на ранних этапах изучения такая практика приносит лишь вред, ведь позволяет не держать в голове всю логику кода. Ещё одной проблемой, хоть и не столь значимой, стало использование браузера для ввода/вывода данных, ведь многие браузеры позволяют запретить уведомления с сайта, через которые выводится информация.