МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Магнитогорский государственный технический университет

им. Г. И. Носова»

(ФГБОУ ВО «МГТУ им. Г.И. Носова»)

Кафедра вычислительной техники и программирования

**Спецификация класса “LongInt” на JavaScript**

по дисциплине «Объектно-ориентированное программное обеспечение»

Исполнитель: Варламов М.Н. студент 2 курса, группа АВб-19-1

Руководитель: Ильина Е.А., доцент каф. ВТиП, к.п.н.

Магнитогорск, 2021

**Описание полей и методов**

static sum(a, b) – Операция суммы. Алгоритм представляет собой складывание чисел в столбик.

static diff(a, b) – Операция разности. Алгоритм представляет собой разность чисел в столбик.

static multiply(a, b) – Операция умножения. Алгоритм представляет собой произведение чисел в столбик.

static div(a, b) – Операция целочисленного деления. Алгоритм представляет собой деление чисел в столбик.

static mod(a, b) – Операция нахождения остатка от деления. Алгоритм находит остаток от деления по формуле r = n - mk, где n - делимое, m - делитель, k - частное, r – остаток. Вызывает все соответствующие функции.

static greater(a, b) – Оператор сравнения “больше”. Сравнивает посимвольно числа, при первом несовпадении возвращает соответствующий результат.

static greaterOrEquals(a, b) – Оператор сравнения “больше, либо равно”. Возвращает логическое “Или” между операторами “Больше” и “Равно”.

static less(a, b) – Оператор сравнения “меньше”. Работает аналогично оператору “Больше”

static lessOrEquals(a, b) – Оператор сравнения “меньше либо равно”. Работает аналогично оператору “Больше либо равно”

static equals(a, b) – Оператор сравнения. Сравнивает посимвольно числа, при первом несовпадении возвращает “false”.

static abs(a) – Операция нахождения модуля. Посимвольно убирает ‘-’ в начале.

**Листинг**

class LongInt{

static sum(a, b){

var negative = false

if (a[0] == '-' && b[0] == '-'){

a = a.substring(1, a.length);

b = b.substring(1, b.length);

negative = true;

}

var max = a, min = b;

if (max.toString().length < min.toString().length){

[max, min] = [min, max]; //swap

}

var maxInd = max.toString().length - 1, minInd = min.toString().length - 1, ost = 0;

var res = "";

for (let i = 0; i < max.toString().length; i++){

var n1 = parseInt(max[maxInd])

if (minInd !== -1){

var n2 = parseInt(min[minInd])

var c = n1 + n2 + ost

ost = 0;

if (c >= 10) {

c -= 10;

ost++

}

res += c

minInd--;

} else{

var c = n1 + ost

ost = 0

if (c >= 10){

c -= 10

ost++

}

res += c

}

maxInd--;

}

res += ost;

while(res[res.length - 1] == '0' && res.length !== 1){

console.log(res[res.length - 1]);

res = res.substring(0, res.length - 1);

}

res = res.split("").reverse().join("");

if(negative)

res = "-" + res;

return res;

}

static diff(a, b){

var max = a, min = b;

var res = "";

var negative = false;

if (LongInt.greater(min, max)){

[max, min] = [min, max];

negative = true;

}

var maxInd = max.toString().length - 1, minInd = min.toString().length - 1, ost = 0;

for (let i = 0; i < max.toString().length; i++){

var n1 = parseInt(max[maxInd]);

if (minInd !== -1){

var n2 = parseInt(min[minInd])

var c = n1 - n2 - ost;

console.log(c);

ost = 0;

if (c < 0) {

c += 10;

ost++;

}

res += c;

minInd--;

} else{

var c = n1 - ost;

ost = 0;

if (c < 0){

c += 10

ost++

}

res += c

}

maxInd--;

}

res += ost;

while(res[res.length - 1] == '0' && res.length !== 1){

console.log(res[res.length - 1]);

res = res.substring(0, res.length - 1);

}

res = res.split("").reverse().join("");

if(negative)

res = "-" + res;

return res;

}

static multiply(a, b){

}

static div(a, b){

if (LongInt.equals(a, "0") && !LongInt.equals(b, "0"))

return "0";

if (LongInt.equals(b, "0"))

return "Division by zero";

if (LongInt.less(a, b))

return "0";

var negative = false;

if (LongInt.less(a, 0)){

negative = !negative;

a = LongInt.abs(a);

}

if (LongInt.less(b, 0)){

negative = !negative;

b = LongInt.abs(b);

}

var res = "";

while (LongInt.greaterOrEquals(LongInt.multiply(res, b)))

LongInt.sum(1, res);

return res;

}

static mod(a, b){

if (LongInt.equals(a, "0") && !LongInt.equals(b, "0"))

return "0";

if (LongInt.equals(b, "0"))

return "Division by zero";

if (LongInt.less(LongInt.abs(a), LongInt.abs(b)))

return a.toString()

return (LongInt.diff(a, (LongInt.multiply(b, LongInt.div(a, b)))));

}

static greater(a, b){

a = a.toString();

b = b.toString();

if (a[0] == '-' && b[0] != '-')

return false;

if (a[0] != '-' && b[0] == '-')

return true;

if (a[0] == '-' && b[0] == '-'){

a = a.substring(1, a.length - 1);

b = b.substring(1, b.length - 1);

return LongInt.less(a, b);

}

if (a.length > b.length)

return true;

else

return false;

for (let i = 0; i < a.length; i++){

var n1 = parseInt(a[i]), n2 = parseInt(b[i]);

if (n1 > n2)

return true;

else if (n2 > n1)

return false;

}

return false;

}

static greaterOrEquals(a, b){

return LongInt.greater(a, b) || LongInt.equals(a, b);

}

static less(a, b){

a = a.toString();

b = b.toString();

if (a[0] == '-' && b[0] != '-')

return true;

if (a[0] != '-' && b[0] == '-')

return false;

if (a[0] == '-' && b[0] == '-'){

a = a.substring(1, a.length - 1);

b = b.substring(1, b.length - 1);

return LongInt.greater(a, b);

}

if (a.length > b.length)

return false;

else if (a.length < b.length)

return true;

for (let i = 0; i < a.length; i++){

var n1 = parseInt(a[i]), n2 = parseInt(b[i]);

if (n1 < n2)

return true;

else if (n2 < n1)

return false;

}

return false;

}

static lessOrEquals(a, b){

return LongInt.greater(a, b) || LongInt.equals(a, b);

}

static equals(a, b){

a = a.toString();

b = b.toString();

if (a.length > b.length)

return false;

else if (a.length < b.length)

return false;

if ((a[0] == '-' && b[0] != '-' ) || (a[0] != '-' && b[0] == '-'))

return false;

if (a[0] == '-' && b[0] == '-'){

a = a.substring(1, a.length - 1);

b = b.substring(1, b.length - 1);

}

for (let i = 0; i < a.length; i++){

var n1 = parseInt(a[i]), n2 = parseInt(b[i]);

if (n1 != n2)

return false;

}

return true;

}

static abs(a){

a = a.toString();

if (a[0] == '-') {

return a.substring(1);

}

return a;

}

}