**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

###### ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

###### КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

**ИНСТИТУТ ЦИФРЫ**

**ОТЧЁТ**

**О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

«Spread, деструктуризация, модули, динамический импорт»

Студентки 2 курса, ФИТ-211 группы

**Колесник Полины Олеговны**

Направление 02.03.02 – «Фундаментальная информатика и информационные технологии»

Руководитель:

Доцент Зимин А. И.

Работа защищена

« »

“ ” 2023 г.

Кемерово 2023 г.

**ОТЧЁТ О ПРОДЕЛАННОЙ РАБОТЕ**

**Spread оператор**

**1 задание**

'use strict'

let arr = [2, 5, 7]

console.log(Math.min(...arr));



**2 задание**

'use strict'

let arr1 = [1, 2, 3];

let arr2 = [...arr1, 4, 5, 6];

let arr3 = [...arr2, 7, 8, 9];

let arr = [...arr3, 10, 11];

console.log(arr);



**3 задание**

'use strict'

// let arr = [...12345]

// console.log(arr); //number 12345 is not iterable (cannot read property Symbol(Symbol.iterator))

let arr = [..."12345"]

console.log(arr);



**4 задание**

'use strict'

function mean(...values) {

    let sr\_znac =0

    for (let arg of values) {

        sr\_znac = (sr\_znac + arg)

    }

    console.log(sr\_znac/values.length)

}

mean(1,5,3,4)



**5 задание**

'use strict'

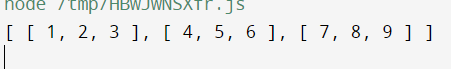
function unite(...res) {

    return res;

}

let result = unite([1, 2, 3], [4, 5, 6], [7, 8, 9]);

console.log(result); // выведет [ [1, 2, 3,] [4, 5, 6], [7, 8, 9] ]

****

**6 задание**

'use strict'

function merge(...res) {

    return [].concat(...res);

}

let result = merge([1, 2, 3], [4, 5, 6], [7, 8, 9]);

console.log(result); // выведет [1, 2, 3, 4, 5, 6, 7, 8, 9]

****

**Деструктуризация массивов**

**7 задание**

'use strict'

let [name, surname, department, position, salary] = "John Smit development programmer 2000".split(" ")

console.log(name, surname, department, position, salary)



**8 задание**

'use strict'

function func () {

    return ['John', 'Smit', 'development', 'programmer', 2000]

}

let [name,surname,department,position, salary ] = func()

console.log(name,surname,department,position, salary);



**9 задание**

'use strict'

let [ , , department, position]  = ['John', 'Smit', 'development', 'programmer', 2000]

console.log(department, position)



**10 задание**

'use strict'

let [name, surname, ...info] = ['John', 'Smit', 'development', 'programmer', 2000]

console.log(info)

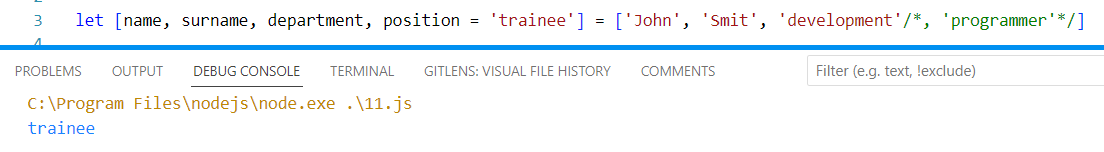


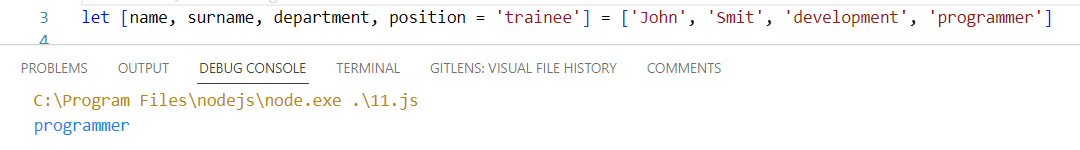
**11 задание**

'use strict'

let [name, surname, department, position = 'trainee'] = ['John', 'Smit', 'development'/\*, 'programmer'\*/]

console.log(position);





**12 задание**

'use strict'

function func () {

    return new Date().getDate()

}

function func1() {

    return new Date().getMonth() + 1

}

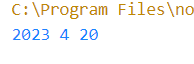
function func2() {

    return new Date().getFullYear()

}

let [year = func2(), month = func1(), day = func()] = []

console.log(year, month, day)

****

**13 задание**

'use strict'

function func ([name, surname,department, position, salary]) {

    console.log(name, surname,department, position, salary)

}

func(['John', 'Smit', 'development', 'programmer', 2000])



**14 задание**

'use strict'

function func ([name, surname, department, position = 'джуниор']) {

    console.log(name, surname, department, position)

}

func(['John', 'Smit', 'development'])



**15 задание**

'use strict'

function func (department, [name, surname], [year, month, day]) {

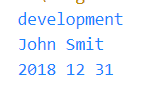
    console.log(department);

    console.log(name, surname);

    console.log(year, month, day);

}

func('development', ['John', 'Smit'], [2018, 12, 31])



**Деструктуризация объектов**

**16 задание**

'use strict'

let options = {

    color: 'red',

    width: 400,

    height: 500

}

let {color, width, height} = options

console.log(color, width, height)



**17 задание**

'use strict'

let options = {

    color: 'red',

    width: 400,

    height: 500

}

let{color: c,width: w, height: h  } = options

console.log(c, w, h);



**18 задание**

'use strict'

let options = {

    width: 400,

    height: 500

}

let {width, height, color = 'black'} = options

console.log(width, height, color);



**19 задание**

'use strict'

let options = {

    width: 400,

    height: 500

}

let {width: w, height: h, color:c = 'black'} = options

console.log(w, h, c);



**20 задание**

'use strict'

function func ({color, width, height}) {

    console.log(color, width, height);

}

func({ color: 'red', width: 400, height: 500 })



**21 задание**

'use strict'

function func ({color = 'black', width, height}) {

    console.log(color, width, height);

}

func({ color: 'red', width: 400, height: 500 })



**Модули, динамический импорт**

**22 задание**

**Файл 22.mjs**

async function load() {

    let readConfig = await import('./22readconfig/readConfig.mjs')

    let doQuery = await import('./22doQuery/doQuery.mjs')

    let httpGet = await import('./22httpGet/httpGet.mjs')

    let readFile = await import('./22readFile/readFile.mjs')

    await readConfig.readConfig('myConfig')

    await doQuery.doQuery('select \* from cities')

    await httpGet.httpGet('http://google.com')

    await readFile.readFile('README.md')

}

load()

**Файл readConfig.mjs**

export async function readConfig (name) {

    return new Promise((resolve, reject) => {

        setTimeout(() => {

            resolve(console.log('(1) config from ' + name + ' loaded'))

        }, Math.floor(Math.random() \* 1000))

    })

}

**Файл doQuery.mjs**

export async function doQuery (statement) {

    return new Promise((resolve, reject) => {

        setTimeout(() => {

            resolve(console.log('(2) SQL query executed: ' + statement))

        }, Math.floor(Math.random() \* 1000))

    })

}

**Файл httpGet.mjs**

export async function httpGet (url) {

    return new Promise((resolve, reject) => {

        setTimeout(() => {

            resolve(console.log('(3) Page retrieved: ' + url))

        }, Math.floor(Math.random() \* 1000))

    })

}

**Файл readFile.mjs**

export async function readFile (path) {

    return new Promise((resolve, reject) => {

        setTimeout(() => {

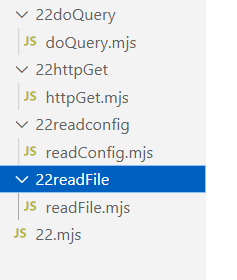
            resolve(console.log('(4) Readme file from ' + path + ' loaded'))

        }, Math.floor(Math.random() \* 1000))

    })

}

**Иерархия файлов**



**Запуск программы**

