МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

**ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

ОТЧЁТ

по лабораторной работе №6

**Имитационное моделирование.**

Выполнил:

студент группы ИИ-22  
Козырев Дмитрий

Проверила:

Слинко Е.В.

# 

# 

Брест 2022  
  
  
**Задача.**  
Вычислить сколько дней будет выполнятся задача в указанном для вашего варианта подразделении, вычислить среднедневное продвижение выполнения задания (в процентах). Выполнить 100 прогонов реализованной модели и вычислить среднее кол-во дней необходимых для выполнения задания, максимальное значение и минимальное значение на выполненном множестве экспериментов.   
  
Вариант **№9**Количество работников на предприятии: **15.**  
Распределение работников в отделе: **(0.1, 0.2, 0.0, 0.6, 0.1)**

***Код программы:***const amountOfWorkers = 15;

const types = [0.1, 0.2, 0.0, 0.6, 0.1]; // распределение работников

const levels = {

highLvl: 0,

middleLvl: 1,

lowLvl: 2,

recruit: 3,

lazyGenius: 4,

};

const days = [];

const workers = [];

let progress = 0;

let day = 1;

let experiment;  
  
for (experiment = 1; experiment < 100; experiment++) { //100 экспериментов

while (progress < 100) { // пока прогресс < 100, выполняем задачу

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

if (progress == 0) {

if (levels.highLvl == i) {

let amountOfHighLvl =

Math.floor(types[levels.highLvl] \* amountOfWorkers) + 1;

for (let i = 1; i <= amountOfHighLvl; i++) {

let highWorker = {

numberOfHighLevel: i,

efficiency: Math.random() \* (0.5 - 0.2) + 0.2,

activity: Math.random() \* (6.5 - 5.5) + 5.5,

};

workers.push(highWorker);

}

} else if (levels.lowLvl == i) {

let amountOfLowLvl = Math.floor(

types[levels.lowLvl] \* amountOfWorkers

);

for (let i = 1; i <= amountOfLowLvl; i++) {

let lowWorker = {

numberOfLowLevel: i,

efficiency: Math.random() \* (0.2 - 0.05) + 0.05,

activity: Math.random() \* (4.5 - 3.5) + 3.5,

};

workers.push(lowWorker);

}

} else if (levels.middleLvl == i) {

let amountOfMiddleLvl = Math.floor(

types[levels.middleLvl] \* amountOfWorkers

);

for (let i = 1; i <= amountOfMiddleLvl; i++) {

let middleWorker = {

numberOfMiddleLevel: i,

efficiency: Math.random() \* (0.3 - 0.1) + 0.1,

activity: Math.random() \* (5.5 - 4.5) + 4.5,

};

workers.push(middleWorker);

}

} else if (levels.recruit == i) {

let amountOfRecruits = Math.floor(

types[levels.recruit] \* amountOfWorkers

);

for (let i = 1; i <= amountOfRecruits; i++) {

let recruit = {

numberOfRecruit: i,

efficiency: Math.random() \* (0.02 - 0.01) + 0.01,

activity: Math.random() \* (2.5 - 1.5) + 1.5,

};

workers.push(recruit);

}

} else if (levels.lazyGenius == i) {

let amountOfLazyGeniuses = Math.floor(

types[levels.lazyGenius] \* amountOfWorkers

);

for (let i = 1; i <= amountOfLazyGeniuses; i++) {

let lazyGenius = {

numberOfLazyGenius: i,

efficiency: Math.random() \* (0.5 - 0.2) + 0.2,

activity: Math.random() \* (1.5 - 0.5) + 0.5,

};

workers.push(lazyGenius);

}

}

}

}

if (progress != 0) {

for (let j = 0; j < workers.length; j++) {

if (workers[j].hasOwnProperty("numberOfHighLevel")) {

workers[j].efficiency = 0;

workers[j].efficiency = Math.random() \* (0.5 - 0.2) + 0.2;

workers[j].activity = 0;

workers[j].activity = Math.random() \* (6.5 - 5.5) + 5.5;

} else if (workers[j].hasOwnProperty("numberOfMiddleLevel")) {

workers[j].efficiency = 0;

workers[j].efficiency = Math.random() \* (0.3 - 0.1) + 0.1;

workers[j].activity = 0;

workers[j].activity = Math.random() \* (5.5 - 4.5) + 4.5;

} else if (workers[j].hasOwnProperty("numberOfLowLevel")) {

workers[j].efficiency = 0;

workers[j].efficiency = Math.random() \* (0.2 - 0.05) + 0.05;

workers[j].activity = 0;

workers[j].activity = Math.random() \* (4.5 - 3.5) + 3.5;

} else if (workers[j].hasOwnProperty("numberOfRecruit")) {

workers[j].efficiency = 0;

workers[j].efficiency = Math.random() \* (0.02 - 0.01) + 0.01;

workers[j].activity = 0;

workers[j].activity = Math.random() \* (2.5 - 1.5) + 1.5;

} else if (workers[j].hasOwnProperty("numberOfLazyGenius")) {

workers[j].efficiency = 0;

workers[j].efficiency = Math.random() \* (0.5 - 0.2) + 0.2;

workers[j].activity = 0;

workers[j].activity = Math.random() \* (1.5 - 0.5) + 0.5;

}

}

}

workers.forEach((worker) => {

let procentOfProgress = worker.efficiency \* worker.activity;

progress += procentOfProgress;

});

console.log(workers);

console.log(`День: ${day}\n. Прогресс: ${progress}`);

day++;

}

days.push(day);

}

let daysNumber = 0;

days.forEach((day) => {

daysNumber = daysNumber + day;

});

let middleValue = daysNumber / days.length - 1;

console.log(

`Среднее значение дней, за которое задача была выполнена, при 100 экспериментах: \n ${middleValue}`

);  
 