ЛАБОРАТОРНА РОБОТА № 0

Рандомізовані алгоритми

Мета: 1) уяснити роль випадковості у фізиці;
2) отримати уявлення про рандомізовані
алгоритми та закони розподілу випадкових величин.
Хід роботи:

Завдання 1:

Лістинг сайту:

```
<!DOCTYPE html>
<html>
<head>
   <meta charset ="utf-8">
   <link rel ="stylesheet" href="data/css/style.css" type="text/css">
   <title>Практична робота №0</title>
<body>
   <header>
       <script type="text/javascript" src="data/js/whoit.js"></script>
   </header>
   <div class="container">
       <canvas id="RandomScript" width="100" height="300">
            <script type="text/javascript" src="data/js/random.js"></script>
       </canvas>
   </div>
</body>
```

					ДУ «Житомирська політехніка».21. <mark>121.4</mark> .000 — Лр0		00 — Лр0	
Змн.	Арк.	№ докум.	Підпис	Дата				•
Розроб.		Волківський Я.С.			Звіт з	Літ.	Арк.	Аркушів
Перевір.		Коломієць Р.О.					1	5
Керіє	вник							
Н. контр.					лабораторної роботи ФІКТ Гр. ІПЗ-23		3-21-5[1]	
Зав.	каф.							

```
header {
    border: 1px solid black;
}

.container {
    display: grid;
    border: 1px solid black;
}

#RandomScript {
    display: inline-grid;
    border: 1px solid black;
}
```

```
whoit(0,"Рандомізовані алгоритми", "Волківський Ярослав", "ІПЗ-21-5")

function whoit(lessonNumber, lessonTheme, studentName, studetnGroup)
{
    document.write("Практичне заняття " + lessonNumber);
    document.write("<h2>" + lessonTheme + "</h2>");
    document.write("" + studentName + "");
    document.write("" + studetnGroup + "");
}
```

```
let min = 0;
let max = 100;
let numPool=[];
let i,tmp;
let maxIterations = 10000;

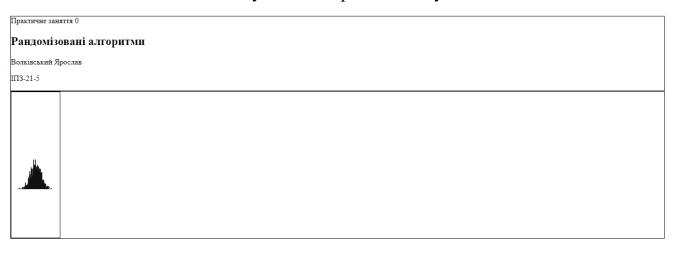
for(i = 0; i < 101; i++)
{
    numPool[i]=0;
}

for(i = 0; i < maxIterations; i++)
{
    tempNumber = (getIntRandomInRange(min, max) +
    getIntRandomInRange(min, max) +getIntRandomInRange(min, max) +
    getIntRandomInRange(min, max))/7;
    numPool[tempNumber]++;
}</pre>
```

		Волківський Я.С.		
		Коломієць Р.Н.		
Змн.	Арк.	№ докум.	Підпис	Дата

```
let canvas = document.getElementById('RandomScript');
if (canvas.getContext)
{
    let ctx = canvas.getContext('2d');
    let x0 = 0;
    let y0 = 200;
    for (i=0; i<max+1; i++)
    {
        ctx.beginPath();
        ctx.moveTo(x0+i, y0);
        ctx.lineTo(x0+i, y0-numPool[i]);
        ctx.closePath();
        ctx.stroke();
    }
}
function getIntRandomInRange(min, max)
{
    return Math.floor(Math.random()*(max-min+1)+min);
}</pre>
```

Результат відкритого сайту:



		Волківський Я.С.		
		Коломієць Р.Н.		
Змн.	Арк.	№ докум.	Підпис	Дата

Лістинг проміжного варіанту файлу randoms.js, який виводить на екран випадкові числа (масив numPool[]):

Лістинг сайту:

```
let min = 0;
let max = 100;
let numPool = [];
let i,tmp;
let maxIterations = 10000;

for(i = 0; i < 101; i++)
{
    numPool[i]=0;
}
for(i = 0; i < maxIterations; i++)
{
    tmp = getIntRandomInRange(0, 100);
    numPool[tmp]++;
}
for(i = 0; i < 101; i++)
{
    document.write("<p>" + i + " " + numPool[i] + "");
}
function getIntRandomInRange(min, max)
{
    return Math.floor(Math.random()*(max-min+1)+min);
}
```

		Волківський Я.С.		
		Коломієць Р.Н.		
Змн.	Арк.	№ докум.	Підпис	Дата

Висновок: уяснив роль випадковості у фізиці; отримав уявлення про ра домізовані алгоритми та закони розподілу випадкових величин.

Перший закон розподілу називається «рівномірним», якщо на інтервалі,

до якого належать всі можливі значення випадкової величини, густина розподілу

зберігає стале значення.

- -Щоб зробити квазі-рівномірний закон розподілу більш рівномірним, потрібно збільшити кількість подій
 - -Математичне сподівання $(середн\epsilon) = середн\epsilon$ арифметичне від тіп та тах
- якщо перед початком формування закону розподілу не обнулити примусово масив, розподіл не працюватиме.

		Волківський Я.С.		
		Коломієць Р.Н.		
Змн.	Арк.	№ докум.	Підпис	Дата