# Universitatea Tehnica Moldovei

**Facultatea Calculatoare, Informatică şi Microelectronică Departamentul Ingineria Software şi Automatică**

Отчёт

**лабораторной работы №1**

**по предмету Interactiune Om Calculator**

**Выполнил: Студент группы ТI-209 Olari Artiom Проверил: Romanenko Alexandr**

**Кишинев 2023**

**1.Общие сведения.**

Одним из основных показателей работы оператора является скорость, с которой он передает информацию от органов индикации на органы управления. Этот показатель, наряду с показателями быстродействия технических элементов, определяет быстродействие всей системы “человек-машина”.

Простая зрительно-моторная реакция (ПЗМР) — это элементарный вид произвольной реакции человека на зрительный стимул. Простая зрительно-моторная реакция состоит из двух последовательных компонентов: сенсорного (латентного) периода и моторного периода.

Методика «Простая зрительно-моторная реакция» предназначена для диагностики скорости данной реакции. Обследуемому последовательно предъявляются световые сигналы красного, зеленого или оранжевого цвета. При появлении сигнала обследуемый должен как можно быстрее нажать на соответствующую кнопку, стараясь при этом не допускать ошибок (ошибками считаются преждевременное нажатие кнопки и пропуск сигнала). Световой сигнал подается в достаточно случайные моменты времени, чтобы не вырабатывался рефлекс на время, при этом достаточно регулярно, чтобы каждый очередной сигнал был ожидаем. Интервал между сигналами составляет от 0,5 до 2,5 с. Первые 5—7 сигналов являются «пробными», предназначены для адаптации обследуемого и не регистрируются. Рекомендуемое число предъявляемых сигналов в одном обследовании — 70, минимальное — 30. Выбор цветового сигнала и количество хроматических проб зависит от целей и задач обследования. При этом необходимо учитывать, что красный цвет сигнала считается наиболее интенсивным раздражителем. Для первичной экспресс-диагностики с целью выявления «групп

риска» достаточно использовать красный световой сигнал. Для более тщательной диагностики необходимо применять не менее двух цветовых сигналов. Латентное время двигательной реакции (ЛВДР) (время или скорость сенсомоторной реакции, быстрота двигательной реакции) является, наряду со скоростью одиночного движения и частотой движений, одним из проявлений быстроты. Быстрота двигательной реакции оценивает лабильность нервно-мышечной системы. Измеряется ЛВДР в миллисекундах как время от начала воздействия зрительного, звукового или тактильного раздражителя до начала мышечного ответа. ЛВДР слагается из следующих составляющих:

- появления возбуждения в рецепторе;

- передачи возбуждения в центральную нервную систему;

- переход возбуждения по нервным путям и формирование эффекторного сигнала;

- проведение эффекторного сигнала от центральной нервной системы к мышце;

- возбуждение мышцы и проявление в ней механической активности.

ЛВДР у лиц, не занимающихся спортом, равно в среднем 250 мс (200 - 350 мс). У спортсменов это время меньше - 150 - 200 мс, а у спринтеров может достигать 100 -120 мс.

Важным условием получения достоверных результатов при измерении времени реакции является изоляция испытуемого от посторонних раздражителей.

**2. Ход выполнения работы.**

Работа выполняется в группе. Группа состоит из трех человек и создает программу для исследований.

Далее каждый член группы поочередно выступает в качестве испытуемого, исследователя и обработчика данных.

Требования к программно-аппаратному комплексу:

1. Если испытуемый готов к работе он нажимает и удерживает клавишу пробела;

2. После нажатия этой клавиши на экране через случайный интервал времени (равномерное распределение от 1 до 5 сек) появляется световой сигнал в виде круга диаметром 5см;

3. Круги могут быть различных цветов, например, красный, зеленый, желтый и цвет должен устанавливаться перед началом испытаний.

4. Обработчик данных может провести исследование зависимости времени реакции от цвета зрительного раздражителя;

5. Программно-аппаратный комплекс должен фиксировать латентный период зрительно-моторной реакции, т.е. время от момента появления светового раздражителя на экране до момента отрыва руки испытуемого от клавиши пробела;

6. Испытуемый должен погасить световой раздражитель путем нажатия на клавишу F6;

7. Программно-аппаратный комплекс должен фиксировать общее время реакции от момента появления на экране светового раздражителя до нажатия клавиши F6 и фиксировать или вычислять время моторной реакции от момента отрыва руки испытуемого от клавиши пробела до момента нажатия клавиши F6;

8. Определяется также латентное время реакции = от момента появления раздражителя на экране до момента отрыва руки от клавиши пробела. Латентное время, это есть время от начала восприятия раздражителя до начала ответа на него.

9. Время реакции должно быть определено с точностью 1 мс и представлено в таблице данных в миллисекундах.

10. Каждый член группы должен проделать 50 испытаний и обработать полученные данные. Фиксируется латентное, моторное и общее время реакции.

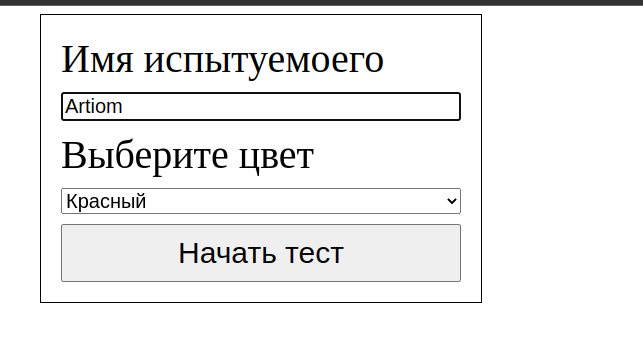
11. Вычислить среднее время латентной, моторной реакции и общего времени реакции;

12. Построить 3 гистограммы для латентной, моторной реакции и для общего времени реакции.

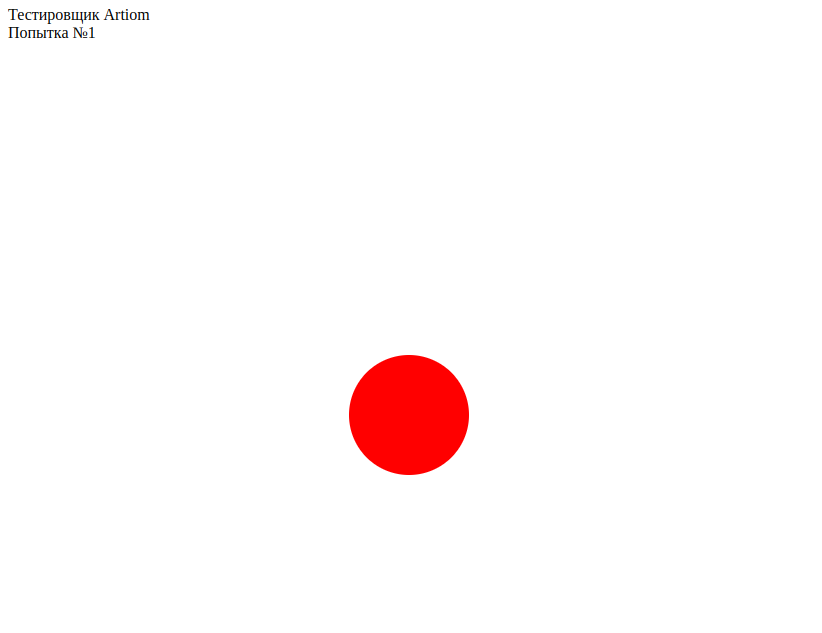
13. Каждый студент в бригаде может выбрать свой цвет раздражителя.

Программа:

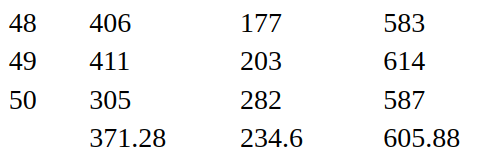
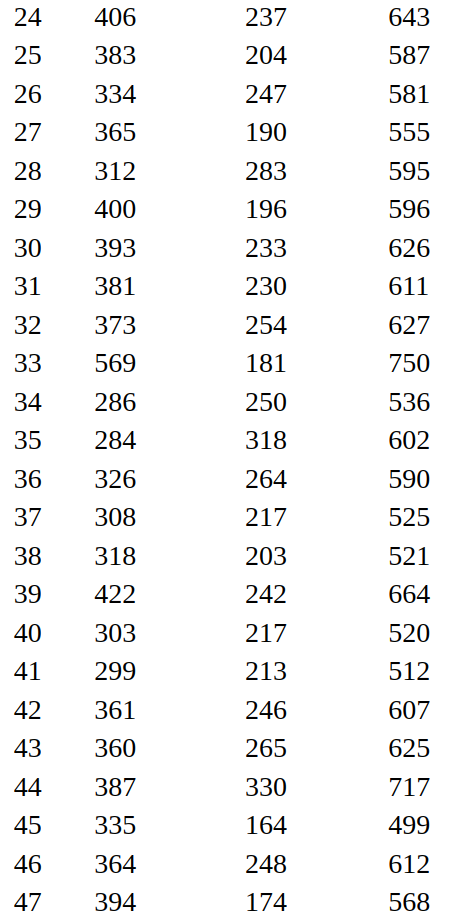
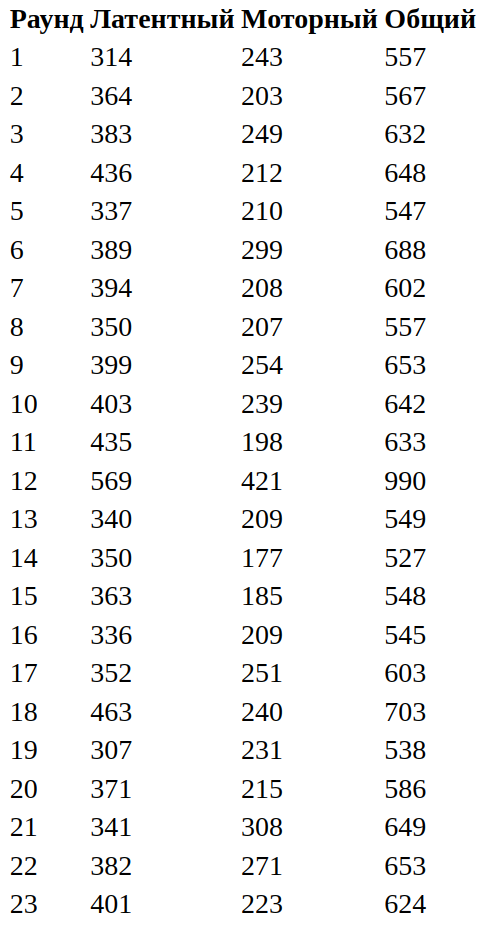
**Настройка:**



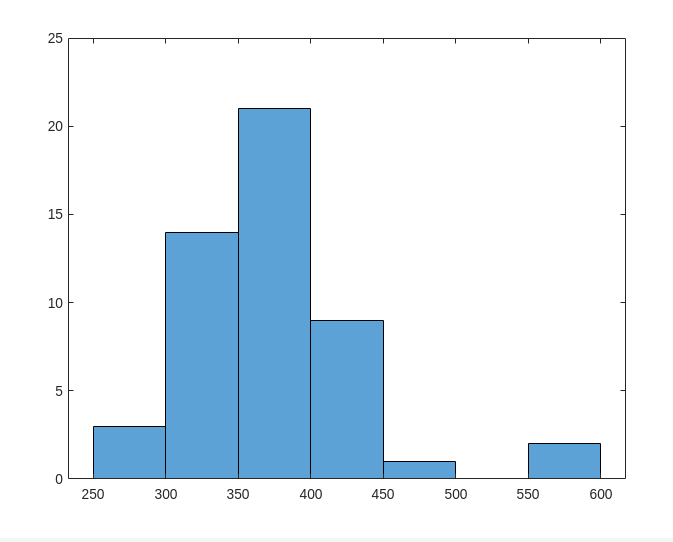
**Тест на скорость:**



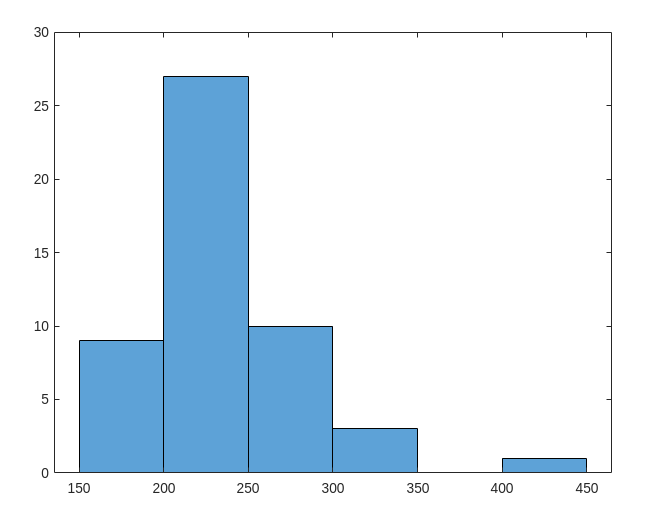
**Результаты:**



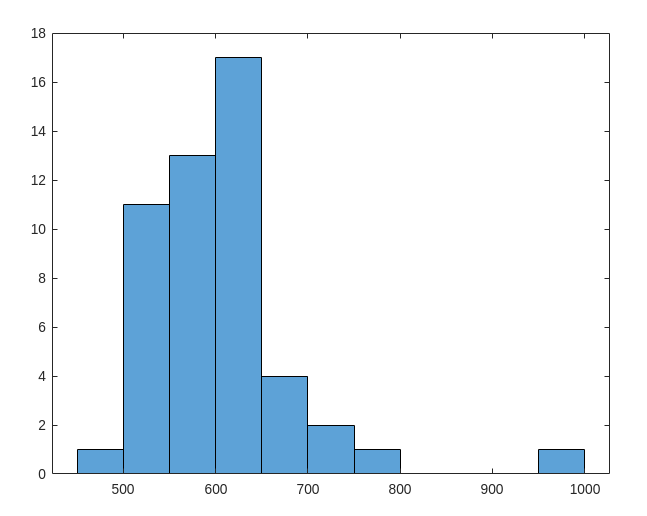
Гистограмма Латентной Реакции



Гистограмма моторной реакции



Гистограмма Общей реакции



Код сервиса отвечающего за вывод сигнала на экран:

import { Inject, Injectable } from '@angular/core';

import { Router } from '@angular/router';

import { BehaviorSubject, timer } from 'rxjs';

import { ROUNDS\_AMOUNT } from '../app.module';

import { SettingsService } from './settings.service';

@Injectable({

providedIn: 'root'

})

export class PlaygroundService {

started = false;

circleData$ = new BehaviorSubject({

showCircle: false,

top: '',

left: '',

size: '',

});

currentRound$ = new BehaviorSubject(0);

totalRounds: number;

score: { [key: number]: { signalShowed: number; spaceUnclicked: number; sevenClicked: number } } = {};

private \_roundInProgress = false;

private \_firstSpaceClicked = true;

constructor(

@Inject(ROUNDS\_AMOUNT) totalRounds: number,

private readonly \_router: Router,

) {

this.totalRounds = totalRounds;

}

spaceClick(): void {

if (this.\_firstSpaceClicked) {

this.started = true;

this.\_firstSpaceClicked = false;

}

if (this.\_roundInProgress) return;

this.\_startTimer();

}

spaceUnClicked(): void {

this.score[this.currentRound$.value].spaceUnclicked = new Date().getTime();

}

sevenClick(): void {

this.\_roundInProgress = false;

this.circleData$.next({

showCircle: false,

left: '',

top: '',

size: ''

});

this.score[this.currentRound$.value].sevenClicked = new Date().getTime();

if (this.currentRound$.value === this.totalRounds) {

this.started = false;

this.\_router.navigate(['results']);

return;

}

}

private \_startTimer(): void {

this.\_roundInProgress = true;

const randomTime = Math.floor(Math.random() \* 5000) + 1000;

timer(randomTime).subscribe(() => {

const top = Math.floor(Math.random() \* 90) + '%';

const left = Math.floor(Math.random() \* 90) + '%';

const size = Math.floor(Math.random() \* 100 + 100) + 'px';

this.circleData$.next({

showCircle: true,

top,

left,

size

});

this.currentRound$.next(this.currentRound$.value + 1);

this.score[this.currentRound$.value] = {

sevenClicked: 0,

signalShowed: 0,

spaceUnclicked: 0,

}

this.score[this.currentRound$.value].signalShowed = new Date().getTime();

})

}

}