Частное учереждение образования

«Колледж бизнеса и права»

Отчет по лабораторной работе №29

Т.091004.401

Преподаватель ( Рогалевич А.В. )

Обучающийся ( Зеневич А.О. )

2024

**Задание №1-5:**

**Задание 1.**

Создайте функцию, которая принимает на вход два аргумента - URL и таймаут в миллисекундах. Функция должна возвращать промис, который выполнится через указанный таймаут и вернет данные, полученные по указанному URL.

**Задание 2.**

Напишите функцию, которая принимает на вход массив промисов и возвращает новый промис, который выполнится, когда все промисы в массиве будут выполнены. Результатом должен быть массив значений, возвращенных каждым из промисов.

**Задание 3.**

Создайте промис, который выполнится через случайное время (от 1 до 5 секунд) и вернет случайное число от 1 до 100. Затем напишите функцию, которая будет вызывать этот промис и выводить результат в консоль.

**Задание 4.**

Напишите функцию, которая принимает на вход массив объектов, у каждого из которых есть поле "url". Функция должна создавать промис для каждого URL и возвращать массив промисов. Когда все промисы будут выполнены, функция должна вернуть массив объектов, в которых к каждому URL добавлено новое поле "data" с данными, полученными по этому URL.

**Задание 5.**

Напишите функцию, которая принимает на вход массив промисов и таймаут в миллисекундах. Функция должна возвращать новый промис, который выполнится через указанный таймаут и вернет массив результатов выполнения каждого из переданных промисов. Если какой-то из промисов не выполнится за указанный таймаут, то он должен быть прерван и вместо результата вернуться ошибку.

**Код задания:**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Лабораторная работа №29</title>

  </head>

  <body>

    <script>

      // Задание 1

      function fetchDataWithTimeout(url, timeout) {

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

          setTimeout(() => {

            resolve(`Данные, полученные от ${url}`);

          }, timeout);

        });

      }

      // Задание 2

      function allPromises(promises) {

        return Promise.all(promises);

      }

      // Задание 3

      function getRandomNumberPromise() {

        return new Promise((resolve) => {

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

          setTimeout(() => {

            const randomNumber = Math.floor(Math.random() \* 100) + 1;

            resolve(randomNumber);

          }, randomTimeout);

        });

      }

      // Задание 4

      function fetchDataForURLs(urls) {

        const promises = urls.map((url) => {

          return fetch(url)

            .then((response) => response.json())

            .then((data) => ({ url, data }))

            .catch((error) => ({ url, error }));

        });

        return Promise.all(promises);

      }

      // Задание 5

      function timeoutForPromises(promises, timeout) {

        return Promise.allSettled(

          promises.map((promise) =>

            Promise.race([

              promise,

              new Promise((\_, reject) =>

                setTimeout(() => reject(new Error("Тайм-аут")), timeout)

              ),

            ])

          )

        );

      }

      fetchDataWithTimeout("https://vk.com/api/data", 3000)

        .then((data) => {

          console.log(data);

          document.body.innerHTML += `<p>${data}</p>`;

        })

        .catch((error) => console.error(error));

      const promisesArray = [

        getRandomNumberPromise(),

        getRandomNumberPromise(),

        getRandomNumberPromise(),

      ];

      allPromises(promisesArray).then((results) => {

        console.log(results);

        document.body.innerHTML += `<p>${JSON.stringify(results)}</p>`;

      });

      fetchDataForURLs([

        "https://jsonplaceholder.typicode.com/comments?postId=1",

        "https://jsonplaceholder.typicode.com/posts/1/comments",

      ]).then((results) => {

        console.log(results);

        document.body.innerHTML += `<p>${JSON.stringify(results)}</p>`;

      });

      timeoutForPromises(promisesArray, 2000).then((results) => {

        console.log(results);

        document.body.innerHTML += `<p>${JSON.stringify(results)}</p>`;

      });

    </script>

  </body>

</html>

**Результат задания:**

