**0МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №5

по дисциплине: **«Современные технологии *Front-end* разработки»**

на тему: **«**Использование объектов типа *Promise* для организации асинхронного кода**»**

Выполнил: студент гр. ИТП-31

Сетко А.И.

Принял: преподаватель-стажер

Васюкова В.О.

Гомель 2020

**Цель**: использование обьектов типа *Promise* для организации асинхронного кода.

**Ход работы**

# **Задание:**

Подготовить json-файл, содержащий следующую информацию:

1. ФИО студента

2. Факультет

3. Специальность

4. Список дисциплин, по которым есть лабораторные работы (каждая дисциплина ­­– это объект, содержащий список лабораторных работ)

Объект "лабораторная работа" должен содержать следующие поля: наименование, дата сдачи, полученная отметка.

Выстроить цепь обработки подготовленного файла, включающую следующие шаги:

1. чтение данных

2. преобразование содержимого файла в объект

3. формирование списка оставшихся лабораторных работ на текущую дату

4. вывод полученного списка

Реализовать обработку ошибок в блоке catch. Для формирования цепи вызовов использовать метод then.

**Верификация программы:**

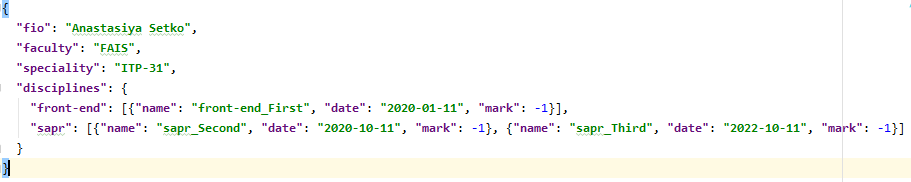


Рисунок 1.1 – Исходный набор данных

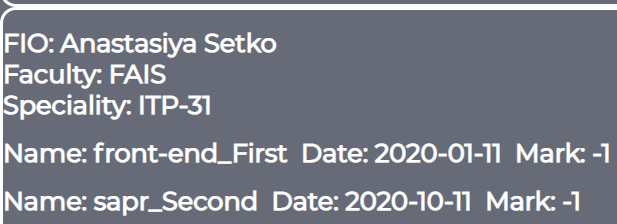


Рисунок 1.2 – Результат чтения данных

**Вывод:** в ходе выполнения лабораторной работы было изучено программирование с помощью объектов типа *Promise*. Данный объект позволяет построить цепочку вызовов избегая такой конструкции как *Callback*. Данный объект имеет два основных метода: *resolve*, *reject*. Первый используется в случае успеха, а второй в случае ошибки. Для цепочных вызовов используется ключевое слово *then*, а для обработки ошибок имеется ключевое слово *catch*.

Для получения данных в формате *json* использовался оператор *fetch*, который позволяет делать сетевые запросы и получать информацию с сервера.

**ПРИЛОЖЕНИЕ А**

**(обязательное)**

**Листинг программы**

**Lab5.js**

**"use strict"**;  
**const *promise*** = **new *Promise***((resolve, reject) => {  
 **let** data = *getJson*();  
  
 **if** (data !== **null** && data !== **undefined**) {  
 resolve(data); *//если в файле что то есть, то все ок* } **else** {  
 reject(**new *Error***(**'Что-то пошло не так'**));  
 }  
});  
  
**function** *readData*() {  
 ***promise*** .then((data) => {  
 **console**.log(data); *// вывести 'Все прошло отлично!'* **return** data;  
 })  
 .then((obj) => {  
 **let** today = ***Date***.parse(**"2020-12-13"**);  
 **let** labs = [];  
 **for**(**let** discipline **in** obj.disciplines) {  
 obj.disciplines[discipline].map(f => {  
 **if** (***Date***.parse(f.date) < today && f.**mark** === -1) labs.push(f);  
 })  
 }  
 **let** res = ***document***.getElementById(**"result-block"**);  
 **let** h1 = ***document***.createElement(**"h1"**);  
 h1.**innerHTML** = **"FIO: "** + obj.fio +  
 **"<br/>Faculty: "** + obj.faculty +  
 **"<br/>Speciality: "** + obj.**speciality**;  
 res.appendChild(h1);  
 **return** labs;  
 })  
 .then((labs) => {  
 **let** res = ***document***.getElementById(**"result-block"**);  
 labs.map(lab => {  
 **let** h1 = ***document***.createElement(**"h1"**);  
 h1.**innerHTML** = **"Name: "** + lab.**name** + **"&nbsp Date: "** + lab.date + **"&nbsp Mark: "** + lab.**mark**;  
 res.appendChild(h1);  
 });  
 })  
 .catch((error) => {  
 **console**.log(error);  
 });  
}  
  
**function** *getJson*() {  
 **return** *fetch*(**"./data.json"**) *//получаем информацию из файла* .then(**function** (resp) {  
 **return** resp.json(); *//возвращаем ее в виде объекта* });  
}