**Лабораторна робота № 2**

**Тема:** Введення в JavaScript розробку. Об'єктно-орієнтоване програмування. Класи

**Мета:** Навчитись працювати з прототипами та класами

**Завдання**

№7

1.Відомості про студентів, що проживають в гуртожитку, містять наступні дані: прізвище, ім'я, по батькові, пів, факультет, номер кімнати, місто з якого приїхало студент. Скласти списки по факультетах і розташувати номери кімнат в порядку зростання і по містах.

2.Облік успішності (студент, оцінка, дисципліна, вид оцінки)

**Хід роботи**

Index.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>ЛМВ Лр2</title>  
</head>  
<body>  
<script src="script.js"></script>  
</body>  
</html>

Script.js

function task1() {  
 function Student(surname, name, middle\_name, faculty, room\_number, city) {  
 const student = {};  
 student.surname = surname;  
 student.name = name;  
 student.middle\_name = middle\_name;  
 student.faculty = faculty;  
 student.room\_number = room\_number;  
 student.city = city;  
 return student;  
 }  
  
 const Liza = new Student('Цапик', 'Єлизавета', 'Романівна', 'ПІ', 27, 'Хмельницький');  
 const Ania = new Student('Хуршудян', 'Аня', 'Юріївна', 'ПМ', 26, 'Хмельницький');  
 const Roman = new Student('Зейлик', 'Роман', 'Вікторович', 'ПІ', 32, 'Хмельницький');  
 const Sasha = new Student('Муштин', 'Олександр', 'Романович', 'ПІ', 37, 'Шепетівка');  
 const Olga = new Student('Кулачук', 'Ольга', 'Романівна', 'ПІ', 38, 'Шепетівка');  
 const Masha = new Student('Кулачук', 'Марія', 'Романівна', 'КІ', 28, 'Шепетівка');  
 const Vania = new Student('Городовий', 'Іван', 'Петрович', 'ПМ', 255, 'Шепетівка');  
  
 const students = [Liza, Ania, Roman, Sasha, Olga, Masha, Vania];  
 ***Array***.prototype.getListsBy = function (by) {  
 const lists = {};  
 this.forEach((item) => {  
 if (lists[item[by]]) {  
 lists[item[by]].push(item);  
 } else {  
 lists[item[by]] = [item];  
 }  
 });  
 return lists;  
 }  
  
 ***Array***.prototype.sortBy = function (by) {  
 function compare(a, b) {  
 if (a[by] < b[by]) {  
 return -1;  
 }  
 if (a[by] > b[by]) {  
 return 1;  
 }  
 return 0;  
 }  
  
 return this.sort(compare);  
 }  
 ***console***.log('Lists from faculties', students.getListsBy('faculty'));  
 ***console***.log('Sort by room number', students.sortBy('room\_number'));  
}  
  
function task2() {//студент, оцінка, дисципліна, вид оцінки  
 class Student {  
 static *newId* = 1;  
  
 constructor(surname, name, middle\_name, faculty, group) {  
 this.id = Student.*newId*++;  
 this.surname = surname;  
 this.name = name;  
 this.middle\_name = middle\_name;  
 this.faculty = faculty;  
 this.group = group;  
 }  
 }  
  
 class Discipline {  
 constructor(id, name) {  
 this.id = id;  
 this.name = name;  
 }  
  
 static *Math* = new Discipline(1, 'Math');  
 static *Ukrainian* = new Discipline(2, 'Ukrainian');  
 static *English* = new Discipline(3, 'English');  
 }  
  
 class Mark {  
 constructor(mark, markType, discipline, student) {  
 this.mark = mark;  
 this.markType = markType;  
 this.discipline = discipline;  
 this.student = student;  
 }  
  
 static *marks* = [];  
  
 static *addMark*(mark) {  
 this.*marks*.push(mark);  
 }  
  
 static *marksBy*(by) {  
 const lists = {};  
 this.*marks*.forEach((item) => {/\*  
 console.log('item[by]', item[by]);//student  
 console.log('item', item);//mark\*/  
 if (lists[item[by].id]) {  
 lists[item[by].id].marks.push(item);  
 } else {  
 const o = {marks:[item]};  
 o[by] = item[by];  
 lists[item[by].id] = o;  
 }  
 });  
 return lists;  
 }  
 }  
  
 class MarkType {  
 constructor(minScore, maxScore, increase) {  
 this.minScore = minScore;  
 this.maxScore = maxScore;  
 this.increase = increase;  
 }  
  
 static *MT1\_5* = new MarkType(1, 5, 1);  
 static *MT1\_12* = new MarkType(1, 12, 1);  
 }  
  
 const Liza = new Student('Цапик', 'Єлизавета', 'Романівна', 'ПІ', 181);  
 const Ania = new Student('Хуршудян', 'Аня', 'Юріївна', 'ПМ', 192);  
 const Roman = new Student('Зейлик', 'Роман', 'Вікторович', 'ПІ', 182);  
 const Sasha = new Student('Муштин', 'Олександр', 'Романович', 'ПІ', 211);  
 const Olga = new Student('Кулачук', 'Ольга', 'Романівна', 'ПІ', 181);  
 const Masha = new Student('Кулачук', 'Марія', 'Романівна', 'КІ', 202);  
 const Vania = new Student('Городовий', 'Іван', 'Петрович', 'ПМ', 201);  
  
 const students = [Liza, Ania, Roman, Sasha, Olga, Masha, Vania];  
 Mark.*addMark*(new Mark(5, MarkType.*MT1\_5*, Discipline.*Ukrainian*, students[0]));  
 Mark.*addMark*(new Mark(5, MarkType.*MT1\_5*, Discipline.*Math*, students[0]));  
 Mark.*addMark*(new Mark(4, MarkType.*MT1\_5*, Discipline.*Math*, students[1]));  
 Mark.*addMark*(new Mark(4, MarkType.*MT1\_12*, Discipline.*English*, students[6]));  
 Mark.*addMark*(new Mark(7, MarkType.*MT1\_12*, Discipline.*Ukrainian*, students[6]));  
 ***console***.log('Marks by students', Mark.*marksBy*('student'));  
 ***console***.log('Marks by discipline', Mark.*marksBy*('discipline'));  
}  
  
task1();  
task2();

**Висновок:** Навчилась працювати з прототипами та класами