Лабораторна робота №2

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

**Мета:** Навчитись працювати з класами JavaScript.

Хід роботи

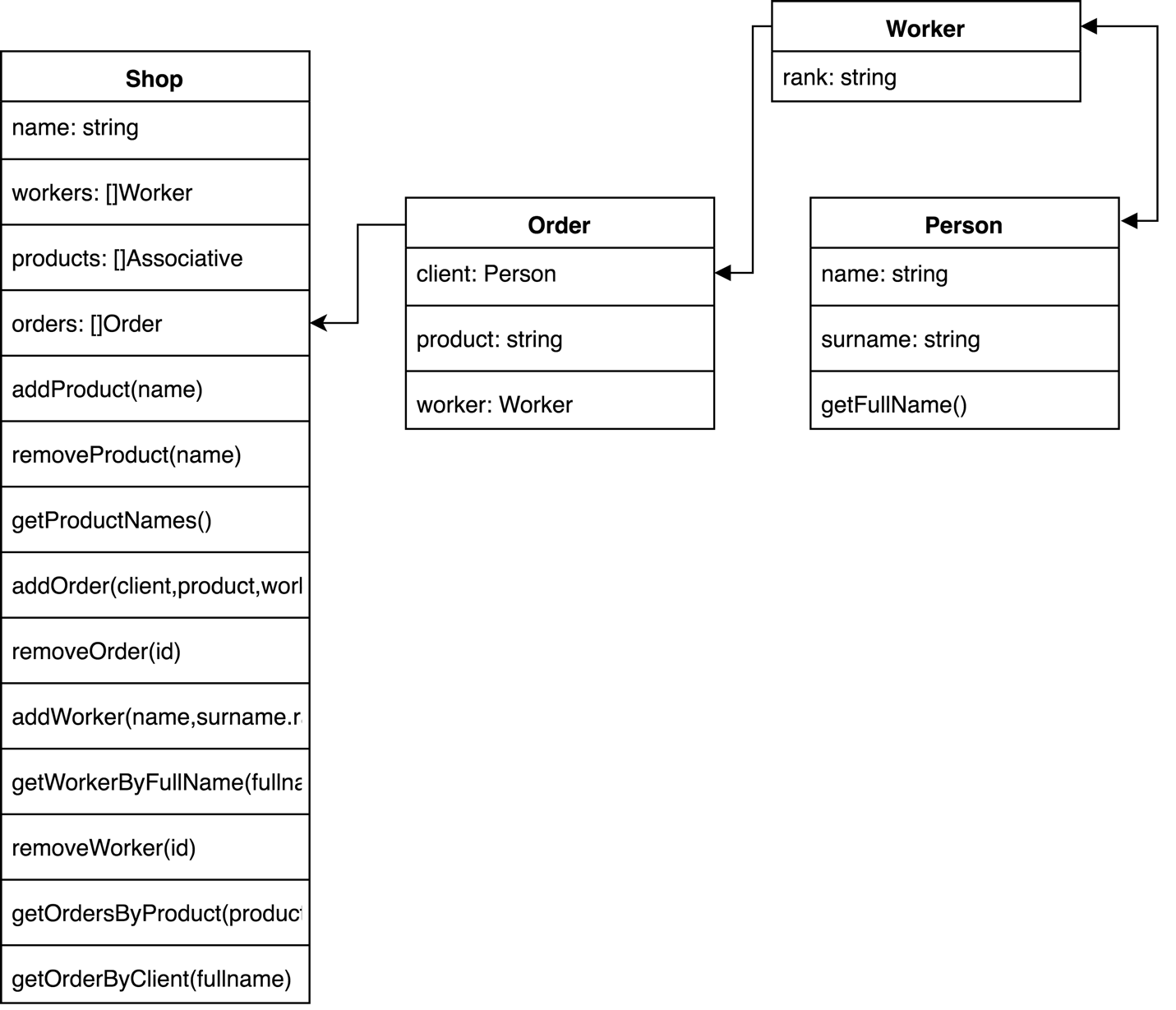
**Варіант 9**

1.Кожен запис за штатним розкладом підприємства містить наступні дані: номер цеху, прізвище і ініціали, професію, розряд, стаж роботи. Скласти список слюсарів і фрезерувальників, що працюють на даному підприємстві більше 10 років і що мають розряд вище четвертого.

2.Мережа магазинів (людина, покупка, продавець, магазин)

**Код програми**

**Діаграма класів**

****

**Завдання 1**

**lw2\_1.js**

**class** Worker {  
  
 // Кожен запис за штатним розкладом підприємства містить наступні дані: номер цеху, прізвище і ініціали, професію,  
 // розряд, стаж роботи. Скласти список слюсарів і фрезерувальників, що працюють на даному підприємстві більше 10 років  
 // і що мають розряд вище четвертого.  
  
  
 constructor(fac\_id, name, surname, midname, profession, level, exp) {  
 **this**.fac\_id = fac\_id;  
 **this**.name = name;  
 **this**.surname = surname;  
 **this**.midname = midname;  
 **this**.profession = profession;  
 **this**.level = level;  
 **this**.exp = exp;  
 }

getInitials() {  
 **return this**.name[0] + '. ' + **this**.midname[0] + '.';  
 }  
  
 **static** *getRandomProfession*() {  
 **let** professions = ['locksmith', 'freezer', 'cleaner', 'smith'];  
 **return** professions[getRandomInt(0, professions.length)];  
 }  
}  
  
**let** workers = [];  
**let** dataProvider = [];  
  
**function** getRandomInt(min = 0, max) {  
 **return** Math.floor(Math.random() \* (max - min)) + min;  
}  
  
**function** getRandomName() {  
 **let** names = [  
 'Adam', 'James', 'Sergey', 'John', 'Viktor',  
 'Wilson', 'Illya', 'Valeriy', 'Pope', 'Thompson',  
 'Rick', 'Vladislav', 'Vladimir', 'Oleg', 'Ulric',  
 'Petrovich', 'Stanislav', 'Yuriy', 'Miroslav', 'Sedrik',  
 'Colson', 'Cole', 'Johaness', 'Johannson', 'Joel',  
 ];  
 **return** names[getRandomInt(0, names.length)];  
}  
  
**function** addRecord() {  
 workers.push(**new** Worker(  
 parseInt(document.getElementById('fac\_id').value),  
 document.getElementById('name').value,  
 document.getElementById('surname').value,  
 document.getElementById('midname').value,  
 document.getElementById('profession').value,  
 parseInt(document.getElementById('level').value),  
 parseInt(document.getElementById('exp').value)  
 ));  
 renderContent();  
}  
  
**function** renderContent(filter = **false**) {  
 dataProvider = [];  
 **if** (filter) {  
 **let** exp = document.getElementById('exp-filter'),  
 fac = document.getElementById('fac-filter');  
 **let** prof = [];  
 [...document.getElementsByClassName('prof-filter')].forEach(element => {  
 **if** (element.checked) prof.push(element.getAttribute('placeholder'));  
 });  
 **if** (prof.length < 1) prof = ['locksmith', 'freezer', 'cleaner', 'smith'];  
 workers.forEach(value => {  
 **if** (fac.value === **undefined** || fac.value === **null** || fac.value === "") {  
 **if** (value.exp >= parseInt(exp.value) && prof.includes(value.profession))  
 dataProvider.push(value);  
 }  
 **else if** (value.exp >= parseInt(exp.value) && value.fac\_id === parseInt(fac.value) && prof.includes(value.profession))  
 dataProvider.push(value);  
 });  
 } **else** workers.forEach(value => dataProvider.push(value));  
  
  
 **let** container = document.getElementById('content');  
 container.innerHTML = document.getElementById('headers').innerHTML;  
  
 dataProvider.forEach(value => {  
 **let** row = document.createElement('tr');  
  
 **let** col = document.createElement('td');  
 col.innerHTML = value.fac\_id;  
 row.appendChild(col);  
  
 col = document.createElement('td');  
 col.innerHTML = value.surname + ' ' + value.getInitials();  
 row.appendChild(col);  
  
 col = document.createElement('td');  
 col.innerHTML = value.profession;  
 row.appendChild(col);  
  
 col = document.createElement('td');  
 col.innerHTML = value.level;  
 row.appendChild(col);  
  
 col = document.createElement('td');  
 col.innerHTML = value.exp;  
 row.appendChild(col);  
  
 container.appendChild(row);  
 });  
}  
  
**function** randomFill() {  
 **for** (**let** i = 0; i < 15; i++)  
 workers.push(  
 **new** Worker(  
 getRandomInt(1, 4),  
 getRandomName(),  
 getRandomName(),  
 getRandomName(),  
 Worker.*getRandomProfession*(),  
 getRandomInt(1, 6),  
 getRandomInt(1, 30)  
 )  
 );  
}  
// Auto exec  
randomFill();  
renderContent();

**Зображення**

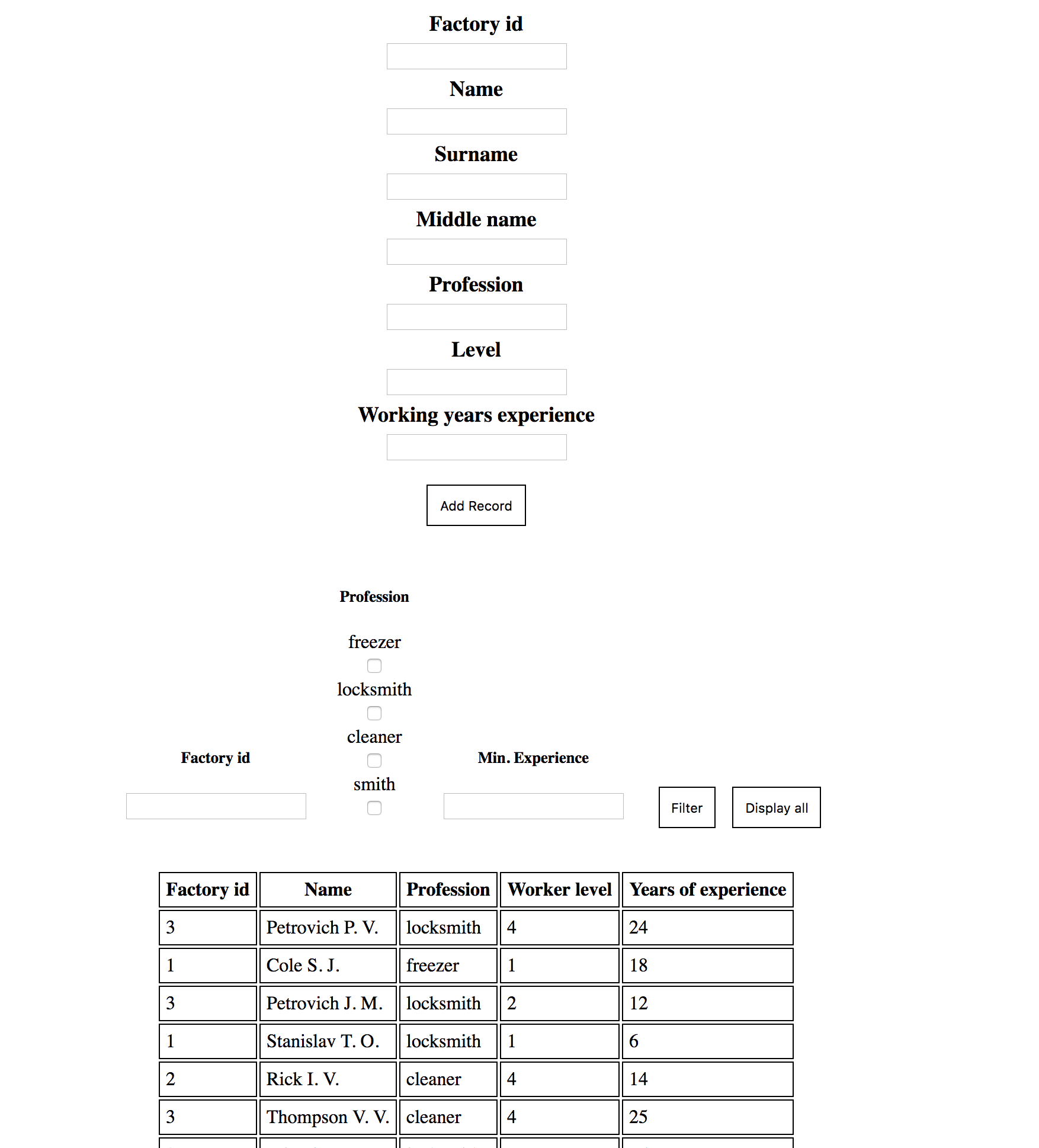


Рисунок 1. Завдання 1

**Завдання 2**

**lw2\_2.js**

// Utility  
**function** getRandomInt(min = 0, max) {  
 **return** Math.floor(Math.random() \* (max - min)) + min;  
}  
  
**function** getRandomName() {  
 **let** names = [  
 'Adam', 'James', 'Sergey', 'John', 'Viktor',  
 'Wilson', 'Illya', 'Valeriy', 'Pope', 'Thompson',  
 'Rick', 'Vladislav', 'Vladimir', 'Oleg', 'Ulric',  
 'Petrovich', 'Stanislav', 'Yuriy', 'Miroslav', 'Sedrik',  
 'Colson', 'Cole', 'Johaness', 'Johannson', 'Joel',  
 'Eugen', 'Jeremy', 'Sean', 'Alexander', 'Markus',  
 ];  
 **return** names[getRandomInt(0, names.length)];  
}  
  
**function** getRandomShop() {  
 **let** shops = [  
 'Pickleboard', 'McDuck', 'McCheese', 'Doughnuts Co.', 'IKEA Workshop',  
 'Intel Hard', 'AMD Prime', 'Aesthetics Inn.', 'Miramar', 'Aquamarine',  
 'PomedorMagaz', 'SomeCool', 'NiceShop', 'IRanOutOfNames', 'Randomness',  
 'JesusHelpsYou', 'Tovari\_Petrovicha', 'HelpMe', 'I\'mSoTired', 'FfsEndItPls',  
 'JSisAwfulForThisTask', 'Why\_Tho', 'MuchRealism', 'NiceNames', 'CulShewps',  
 ];  
 **return** shops[getRandomInt(0, shops.length)];  
}  
  
**function** getRandomRank() {  
 **let** ranks = [  
 'Assistant', 'Cashier', 'Manager', 'Cleaner',  
 ];  
 **return** ranks[getRandomInt(0, ranks.length)];  
}  
  
**function** getRandomProduct() {  
 **let** products = [  
 'Cup', 'Plate', 'Gun', 'Car', 'Something',  
 ];  
 **return** products[getRandomInt(0, products.length)];  
}  
  
// Classes  
**class** Shop {  
  
 constructor(name, workers = [], products = {}, orders = []) {  
 **this**.name = name;  
 **this**.workers = workers;  
 **this**.products = products;  
 **this**.orders = orders;  
 }  
  
 addProduct(name) {  
 **if** (Object.keys(**this**.products).includes(name))  
 **this**.products[name]++;  
 **else this**.products[name] = 1;  
 }  
  
 removeProduct(name) {  
 **if** (Object.keys(**this**.products).includes(name))  
 **if** (**this**.products[name] !== 0) **this**.products[name]--;  
 **let** zeroCount = [];  
 **this**.products.forEach((value, index) => {  
 **if** (value === 0) zeroCount.push(index);  
 });  
 zeroCount.forEach(value => **this**.products.splice(value, 1));  
 }  
  
 getProductNames() {  
 **let** names = [];  
 **this**.products.forEach(item => {  
 names.push(item.name);  
 });  
 **return** names;  
 }  
  
 addOrder(client, product, worker) {  
 **this**.orders.push(**new** Order(client, product, worker));  
 }  
  
 removeOrder(id) {  
 **this**.orders.splice(id, 1);  
 }  
  
 addWorker(name, surname, rank) {  
 **this**.workers.push(**new** Worker(name, surname, rank));  
 }  
  
 getWorkerByFullName(fullname) {  
 **let** names = fullname.split(' '),  
 name = names[0],  
 surname = names[1];  
 **for** (**let** i **in this**.workers) {  
 **if** (**this**.workers[i].name === name && **this**.workers[i].surname === surname) **return this**.workers[i];  
 }  
 **return null**;  
 }  
  
 removeWorker(id) {  
 **this**.workers.splice(id, 1);  
 }  
  
 getOrdersByProduct(product) {  
 **let** match = [];  
 **this**.orders.forEach(value => {  
 **if** (value.product === product) match.push(value);  
 });  
 **return** match;  
 }  
  
 getOrdersByClient(fullname) {  
 **let** match = [];  
 **this**.orders.forEach(value => {  
 **if** (value.client.getFullName() === fullname) match.push(value);  
 });  
 **return** match = [];  
 }  
  
}  
  
**class** Product {  
  
 constructor(name) {  
 **this**.name = name;  
 **this**.count = 1;  
 }  
  
 addCount() {  
 **this**.count++;  
 }  
  
 removeCount() {  
 **if** (**this**.count !== 0)  
 **this**.count--;  
 }  
  
}  
  
**class** Order {  
  
 constructor(client, product, worker) {  
 **this**.client = client;  
 **this**.product = product;  
 **this**.worker = worker;  
 }  
  
}  
  
**class** Person {  
  
 constructor(name, surname) {  
 **this**.name = name;  
 **this**.surname = surname;  
 }  
  
 getFullName() {  
 **return this**.name + ' ' + **this**.surname;  
 }  
  
}  
  
**class** Worker **extends** Person {  
  
 constructor(name, surname, rank) {  
 **super**(name, surname);  
 **this**.rank = rank;  
 }  
  
}  
  
// Core  
**let** shops = [];  
  
**let** shop\_table = document.getElementById('shop-table'),  
 worker\_table = document.getElementById('worker-table'),  
 order\_table = document.getElementById('order-table');  
  
// Renders  
**function** renderWorkers(table = document.getElementById('worker-table'), selector = document.getElementById('shop-selector')) {  
 shops.forEach(item => {  
 **if** (item.name === selector.options[selector.selectedIndex].value) {  
 renderData(item.workers, table);  
 **return null**;  
 }  
 });  
}  
  
**function** renderData(dataProvider, table, properties = **null**) {  
 table.innerHTML = '';  
 **if** (dataProvider !== **undefined** && dataProvider !== **null** && dataProvider.length !== 0)  
 **if** (properties == **null**) {  
 table.appendChild(renderHeaders(dataProvider[0]));  
 dataProvider.forEach(item => {  
 table.appendChild(renderValues(item));  
 });  
 } **else** {  
 table.appendChild(renderHeaders(dataProvider[0], properties));  
 dataProvider.forEach(item => {  
 table.appendChild(renderValues(item, properties));  
 });  
 }  
}  
  
**function** renderProductHeaders() {  
 **let** row = document.createElement('tr');  
 **let** header = document.createElement('th');  
 header.innerHTML = 'Product';  
 row.appendChild(header);  
 header = document.createElement('th');  
 header.innerHTML = 'Count';  
 row.appendChild(header);  
 **return** row;  
}  
  
  
**function** renderProducts(table = document.getElementById('product-table'), selector = document.getElementById('shop-selector')) {  
 table.innerHTML = '';  
 table.appendChild(renderProductHeaders());  
 **let** shop = findShopByName(selector.options[selector.selectedIndex].value);  
 **if** (shop.products !== **undefined** && shop.products !== **null**)  
 **for** (**let** name **in** shop.products) {  
 **let** row = document.createElement('tr');  
 **let** col = document.createElement('td');  
 col.innerHTML = name;  
 row.appendChild(col);  
 col = document.createElement('td');  
 col.innerHTML = shop.products[name];  
 row.appendChild(col);  
 table.appendChild(row);  
 }  
}  
  
**function** renderShops() {  
 renderData(shops, shop\_table, ['name']);  
}  
  
**function** renderOrders(table = document.getElementById('order-table'), selector = document.getElementById('order-shop-selector')) {  
 table.innerHTML = '';  
 table.appendChild(renderOrderHeaders());  
 **let** shop = findShopByName(selector.options[selector.selectedIndex].value);  
 **if** (shop.orders !== **undefined** && shop.orders !== **null** && shop.orders.length > 0)  
 shop.orders.forEach((order, index) => {  
 **let** row = document.createElement('tr');  
 **let** col = document.createElement('td');  
 col.innerHTML = index.toString();  
 row.appendChild(col);  
 col = document.createElement('td');  
 col.innerHTML = order.client.getFullName();  
 row.appendChild(col);  
 col = document.createElement('td');  
 col.innerHTML = order.product;  
 row.appendChild(col);  
 col = document.createElement('td');  
 col.innerHTML = order.worker.getFullName();  
 row.appendChild(col);  
 table.appendChild(row);  
 });  
}  
  
**function** renderOrderHeaders() {  
 **let** row = document.createElement('tr');  
 **let** header = document.createElement('th');  
 header.innerHTML = 'Id';  
 row.appendChild(header);  
 header = document.createElement('th');  
 header.innerHTML = 'Client';  
 row.appendChild(header);  
 header = document.createElement('th');  
 header.innerHTML = 'Product';  
 row.appendChild(header);  
 header = document.createElement('th');  
 header.innerHTML = 'Worker';  
 row.appendChild(header);  
 **return** row;  
}  
  
**function** renderHeaders(object, properties = **null**) {  
 **if** (properties == **null**) {  
 **let** row = document.createElement('tr');  
 Object.keys(object).forEach(key => {  
 **let** header = document.createElement('th');  
 header.innerHTML = key;  
 row.appendChild(header);  
 });  
 **return** row;  
 } **else** {  
 **let** row = document.createElement('tr');  
 Object.keys(object).forEach(key => {  
 **if** (properties.includes(key)) {  
 **let** header = document.createElement('th');  
 header.innerHTML = key;  
 row.appendChild(header);  
 }  
 });  
 **return** row;  
 }  
}  
  
**function** renderValues(object, properties = **null**) {  
 **if** (properties == **null**) {  
 **let** row = document.createElement('tr');  
 Object.keys(object).forEach(key => {  
 **let** col = document.createElement('td');  
 // if (Array.isArray(object[key])) {  
 // object[key].forEach(item=>{  
 // Object.keys(item).forEach(key=>{  
 // col.innerHTML+=item[key]+' ';  
 // });  
 // });  
 // } else  
 col.innerHTML = object[key];  
 row.appendChild(col);  
 });  
 **return** row;  
 } **else** {  
 **let** row = document.createElement('tr');  
 Object.keys(object).forEach(key => {  
 **if** (properties.includes(key)) {  
 **let** col = document.createElement('td');  
 // if (Array.isArray(object[key])) {  
 // object[key].forEach(item=>{  
 // Object.keys(item).forEach(key=>{  
 // col.innerHTML+=item[key]+' ';  
 // });  
 // });  
 // } else  
 col.innerHTML = object[key];  
 row.appendChild(col);  
 }  
 });  
 **return** row;  
 }  
}  
  
// Events  
**function** shopSelected(selector = document.getElementById('shop-selector')) {  
 **let** selectedValue = selector.options[selector.selectedIndex].value;  
 renderWorkers();  
 renderProducts();  
 loadWorkerSelector(selectedValue);  
 loadProductSelector(selectedValue);  
 workerSelected();  
 productSelected();  
 loadShopEdit(selectedValue);  
}  
  
**function** shopApplied(selector = document.getElementById('shop-selector')) {  
 **let** selectedValue = selector.options[selector.selectedIndex].value,  
 shop = findShopByName(selectedValue),  
 nameInput = document.getElementById('edit-shop-name');  
 **if** (findShopByName(nameInput.value) === **null**)  
 shop.name = nameInput.value;  
 **else** alert('Such shop name already exists.')  
 renderShops();  
 loadShopSelector();  
 shopSelected();  
}  
  
**function** shopDeleted(selector = document.getElementById('shop-selector')) {  
 **let** selectedValue = selector.options[selector.selectedIndex].value,  
 shop = findShopByName(selectedValue);  
 shops.splice(shops.indexOf(shop), 1);  
 renderShops();  
 loadShopSelector();  
 shopSelected();  
}  
  
**function** workerSelected(selector = document.getElementById('worker-selector')) {  
 **let** selectedOption = selector.options[selector.selectedIndex];  
 **let** selectedValue = (selectedOption === **undefined** || selectedOption === **null**) ? '' : selectedOption.value;  
 loadWorkerEdit(selectedValue);  
}  
  
**function** reloadWorkerSelector() {  
 **let** shopSelector = document.getElementById('shop-selector'),  
 shop = findShopByName(shopSelector.options[shopSelector.selectedIndex].value),  
 workerSelector = document.getElementById('worker-selector'),  
 index = workerSelector.selectedIndex;  
 loadWorkerSelector(shop.name);  
 workerSelector.selectedIndex = index;  
}  
  
**function** workerApplied(selector = document.getElementById('worker-selector')) {  
 **let** selectedValue = selector.options[selector.selectedIndex].value,  
 shopSelector = document.getElementById('shop-selector'),  
 shop = findShopByName(shopSelector.options[shopSelector.selectedIndex].value);  
  
 **let** worker = shop.getWorkerByFullName(selectedValue),  
 nameInput = document.getElementById('edit-name-worker'),  
 surnameInput = document.getElementById('edit-surname-worker'),  
 rankInput = document.getElementById('edit-rank-worker');  
 worker.name = nameInput.value;  
 worker.surname = surnameInput.value;  
 worker.rank = rankInput.value;  
 renderWorkers();  
 reloadWorkerSelector();  
}  
  
**function** workerDeleted(selector = document.getElementById('worker-selector')) {  
 **let** selectedValue = selector.options[selector.selectedIndex].value,  
 shopSelector = document.getElementById('shop-selector'),  
 shop = findShopByName(shopSelector.options[shopSelector.selectedIndex].value);  
 **let** worker = shop.getWorkerByFullName(selectedValue);  
 shop.removeWorker(shop.workers.indexOf(worker));  
 renderWorkers();  
}  
  
**function** productSelected(selector = document.getElementById('product-selector')) {  
 **let** selectedOption = selector.options[selector.selectedIndex];  
 **let** selectedValue = (selectedOption === **undefined** || selectedOption === **null**) ? '' : selectedOption.value;  
 loadProductEdit(selectedValue);  
}  
  
**function** productApplied(selector = document.getElementById('product-selector')) {  
 **let** selectedValue = selector.options[selector.selectedIndex].value,  
 shopSelector = document.getElementById('shop-selector'),  
 shop = findShopByName(shopSelector.options[shopSelector.selectedIndex].value),  
 countInput = document.getElementById('edit-count-product');  
 // shop.products[selectedValue] = parseInt(countInput.value);  
 shop.products[selectedValue] = countInput.value;  
 renderProducts();  
}  
  
**function** productDeleted(selector = document.getElementById('product-selector')) {  
 **let** selectedValue = selector.options[selector.selectedIndex].value,  
 shopSelector = document.getElementById('shop-selector'),  
 shop = findShopByName(shopSelector.options[shopSelector.selectedIndex].value);  
 **delete** shop.products[selectedValue];  
 renderProducts();  
 reloadProductSelector();  
 productSelected();  
}  
  
**function** productAdd(selector = document.getElementById('shop-selector')) {  
 **let** shop = findShopByName(selector.options[selector.selectedIndex].value),  
 nameInput = document.getElementById('add-name-product');  
 shop.addProduct(nameInput.value);  
 renderProducts();  
 reloadProductSelector();  
 // productSelected();  
}  
  
**function** workerAdd(selector = document.getElementById('shop-selector')) {  
 **let** shop = findShopByName(selector.options[selector.selectedIndex].value),  
 nameInput = document.getElementById('add-name-worker'),  
 surnameInput = document.getElementById('add-surname-worker'),  
 rankInput = document.getElementById('add-rank-worker');  
 shop.addWorker(nameInput.value, surnameInput.value, rankInput.value);  
 renderWorkers();  
 reloadWorkerSelector();  
 // workerSelected();  
}  
  
**function** shopAdd() {  
 **let** nameInput = document.getElementById('add-name-shop');  
 **if** (findShopByName(nameInput.value) === **null**)  
 shops.push(**new** Shop(nameInput.value));  
 **else** alert('Such shop already exists.');  
 renderShops();  
 loadShopSelector();  
}  
  
**function** orderShopSelected(table = document.getElementById('order-table'), selector = document.getElementById('order-shop-selector')) {  
 **let** selectedValue = selector.options[selector.selectedIndex].value;  
 renderOrders();  
 loadOrderSelector(selectedValue);  
 loadProductSelector(selectedValue, document.getElementById('add-order-product-selector'));  
 loadWorkerSelector(selectedValue, document.getElementById('add-order-worker-selector'));  
}  
  
**function** orderDeleted(selector = document.getElementById('order-selector')) {  
 **let** selectedValue = selector.options[selector.selectedIndex].value,  
 shopSelector = document.getElementById('order-shop-selector'),  
 shop = findShopByName(shopSelector.options[shopSelector.selectedIndex].value);  
 shop.removeOrder(selectedValue);  
 renderOrders();  
 reloadOrderSelector();  
}  
  
**function** orderAdd() {  
 **let** nameInput = document.getElementById('add-name-client-order'),  
 surnameInput = document.getElementById('add-surname-client-order'),  
 productSelector = document.getElementById('add-order-product-selector'),  
 workerSelector = document.getElementById('add-order-worker-selector'),  
 shopSelector = document.getElementById('order-shop-selector');  
 **let** shop = findShopByName(shopSelector.options[shopSelector.selectedIndex].value),  
 product = productSelector.options[productSelector.selectedIndex].value,  
 worker = shop.getWorkerByFullName(workerSelector.options[workerSelector.selectedIndex].value);  
 shop.addOrder(  
 **new** Person(nameInput.value, surnameInput.value),  
 product,  
 worker,  
 );  
 renderOrders();  
 reloadOrderSelector();  
}  
  
// Support  
**function** reloadOrderSelector() {  
 **let** shopSelector = document.getElementById('order-shop-selector'),  
 shop = findShopByName(shopSelector.options[shopSelector.selectedIndex].value),  
 orderSelector = document.getElementById('order-selector'),  
 index = orderSelector.selectedIndex;  
 loadOrderSelector(shop.name);  
 orderSelector.selectedIndex = index;  
}  
  
**function** loadOrderSelector(shopName, selector = document.getElementById('order-selector')) {  
 selector.innerHTML = '';  
 **let** shop = findShopByName(shopName);  
 shop.orders.forEach((order, index) => {  
 **let** option = document.createElement('option');  
 option.setAttribute('value', index.toString());  
 option.innerHTML = index.toString();  
 selector.appendChild(option);  
 });  
}  
  
**function** loadProductEdit(productName) {  
 // let nameInput = document.getElementById('edit-name-product'),  
 **let** countInput = document.getElementById('edit-count-product');  
 **let** shopSelector = document.getElementById('shop-selector'),  
 shop = findShopByName(shopSelector.options[shopSelector.selectedIndex].value);  
  
 **let** parsedVal = parseInt(shop.products[productName]);  
 countInput.value = (parsedVal === **undefined** || parsedVal === **null**) ? '' : parsedVal;  
}  
  
**function** reloadProductSelector() {  
 **let** shopSelector = document.getElementById('shop-selector'),  
 shop = findShopByName(shopSelector.options[shopSelector.selectedIndex].value),  
 productSelector = document.getElementById('product-selector'),  
 index = productSelector.selectedIndex;  
 loadProductSelector(shop.name);  
 productSelector.selectedIndex = index;  
}  
  
**function** loadProductSelector(shopName, selector = document.getElementById('product-selector')) {  
 selector.innerHTML = '';  
 **let** shop = findShopByName(shopName);  
 **for** (**let** name **in** shop.products) {  
 **let** option = document.createElement('option');  
 option.setAttribute('value', name);  
 option.innerHTML = name;  
 selector.appendChild(option);  
 }  
}  
  
**function** loadWorkerEdit(workerName) {  
 **let** shopSelector = document.getElementById('shop-selector'),  
 shop = findShopByName(shopSelector.options[shopSelector.selectedIndex].value);  
  
 **let** worker = shop.getWorkerByFullName(workerName),  
 nameInput = document.getElementById('edit-name-worker'),  
 surnameInput = document.getElementById('edit-surname-worker'),  
 rankInput = document.getElementById('edit-rank-worker');  
 **if** (worker === **undefined** || worker === **null**) {  
 nameInput.value = '';  
 surnameInput.value = '';  
 rankInput.value = '';  
 } **else** {  
 nameInput.value = worker.name;  
 surnameInput.value = worker.surname;  
 rankInput.value = worker.rank;  
 }  
}  
  
**function** loadShopEdit(shopName) {  
 **let** nameInput = document.getElementById('edit-shop-name'),  
 shop = findShopByName(shopName);  
 **if** (shop === **undefined** || shop === **null**)  
 nameInput.value = '';  
 **else** nameInput.value = shop.name;  
}  
  
**function** loadShopSelector(selector = document.getElementById('shop-selector')) {  
 **let** index = selector.selectedIndex;  
 selector.innerHTML = '';  
 shops.forEach(shop => {  
 **let** option = document.createElement('option');  
 option.setAttribute('value', shop.name);  
 option.innerHTML = shop.name;  
 selector.appendChild(option);  
 });  
 **if** (index > 0)  
 selector.selectedIndex = index;  
}  
  
**function** loadWorkerSelector(shopName, selector = document.getElementById('worker-selector')) {  
 selector.innerHTML = '';  
 **let** shop = findShopByName(shopName);  
 shop.workers.forEach(worker => {  
 **let** option = document.createElement('option');  
 option.setAttribute('value', worker.getFullName());  
 option.innerHTML = worker.getFullName();  
 selector.appendChild(option);  
 });  
 // I don't really remember what it does  
 // shops.forEach(item => {  
 // if (item.name === selector.options[selector.selectedIndex].value) {  
 // renderData(item.workers, table);  
 // return null;  
 // }  
 // });  
}  
  
**function** findShopByName(name) {  
 **for** (**let** i **in** shops) {  
 **if** (shops[i].name === name) **return** shops[i];  
 }  
 **return null**;  
}  
  
**function** randomFill() {  
 **for** (**let** i = 0; i < 15; i++) {  
 **let** shopName = getRandomShop();  
 **while** (findShopByName(shopName) !== **null**)  
 shopName = getRandomShop();  
 shops.push(**new** Shop(shopName));  
 }  
 shops.forEach(shop => {  
 **for** (**let** i = 0; i < 10; i++) {  
 shop.addWorker(getRandomName(), getRandomName(), getRandomRank());  
 shop.addProduct(getRandomProduct());  
 }  
 **for** (**let** i = 0; i < 10; i++) {  
 **let** availableProducts = Object.keys(shop.products);  
 shop.addOrder(  
 **new** Person(getRandomName(), getRandomName()),  
 availableProducts[getRandomInt(0, availableProducts.length)],  
 shop.workers[getRandomInt(0, shop.workers.length)]  
 );  
 }  
 });  
}  
  
  
// Auto exec  
  
randomFill();  
renderShops();

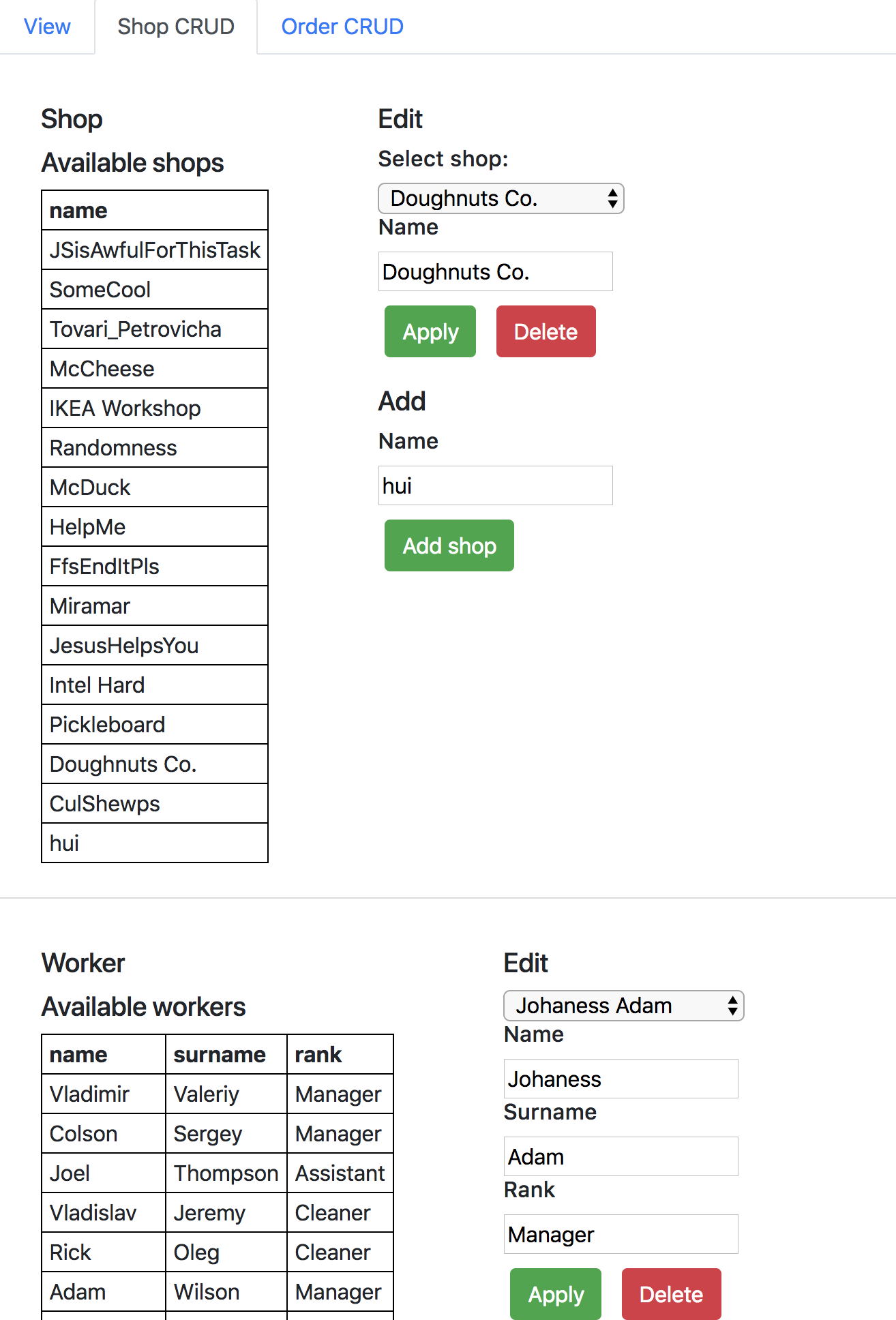


Рисунок 2. Завдання 2 (Магазин і Робітник)

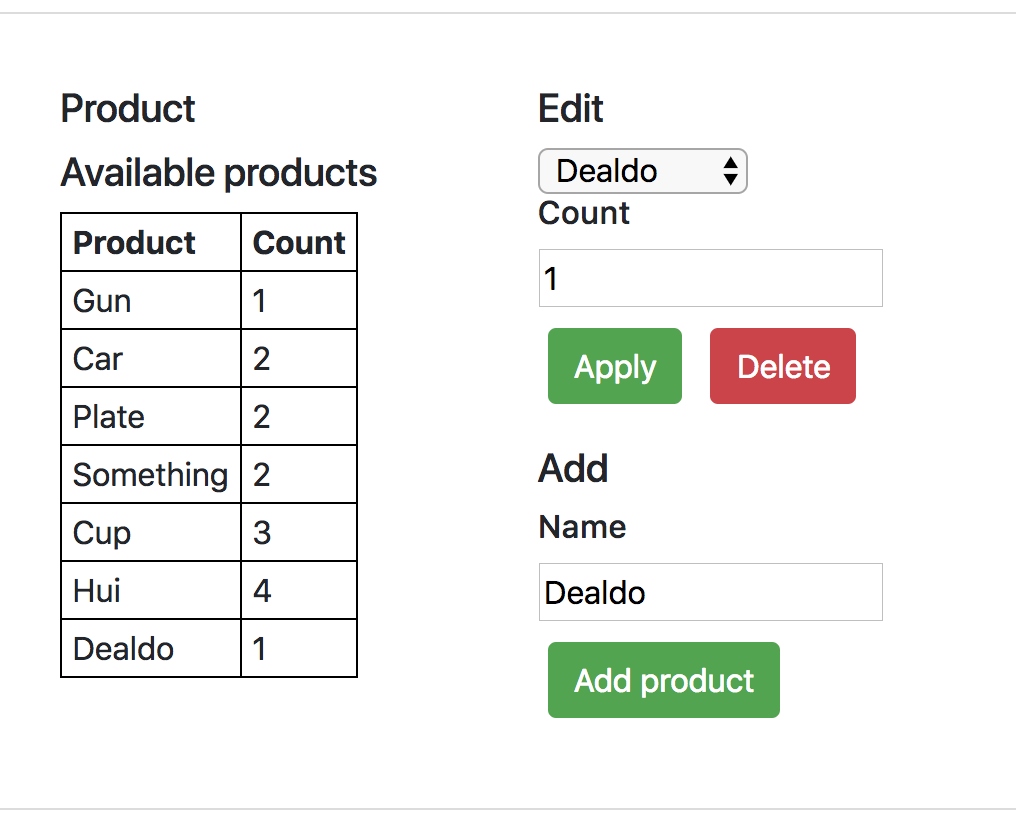


Рисунок 3. Завдання 2 (Продукт)

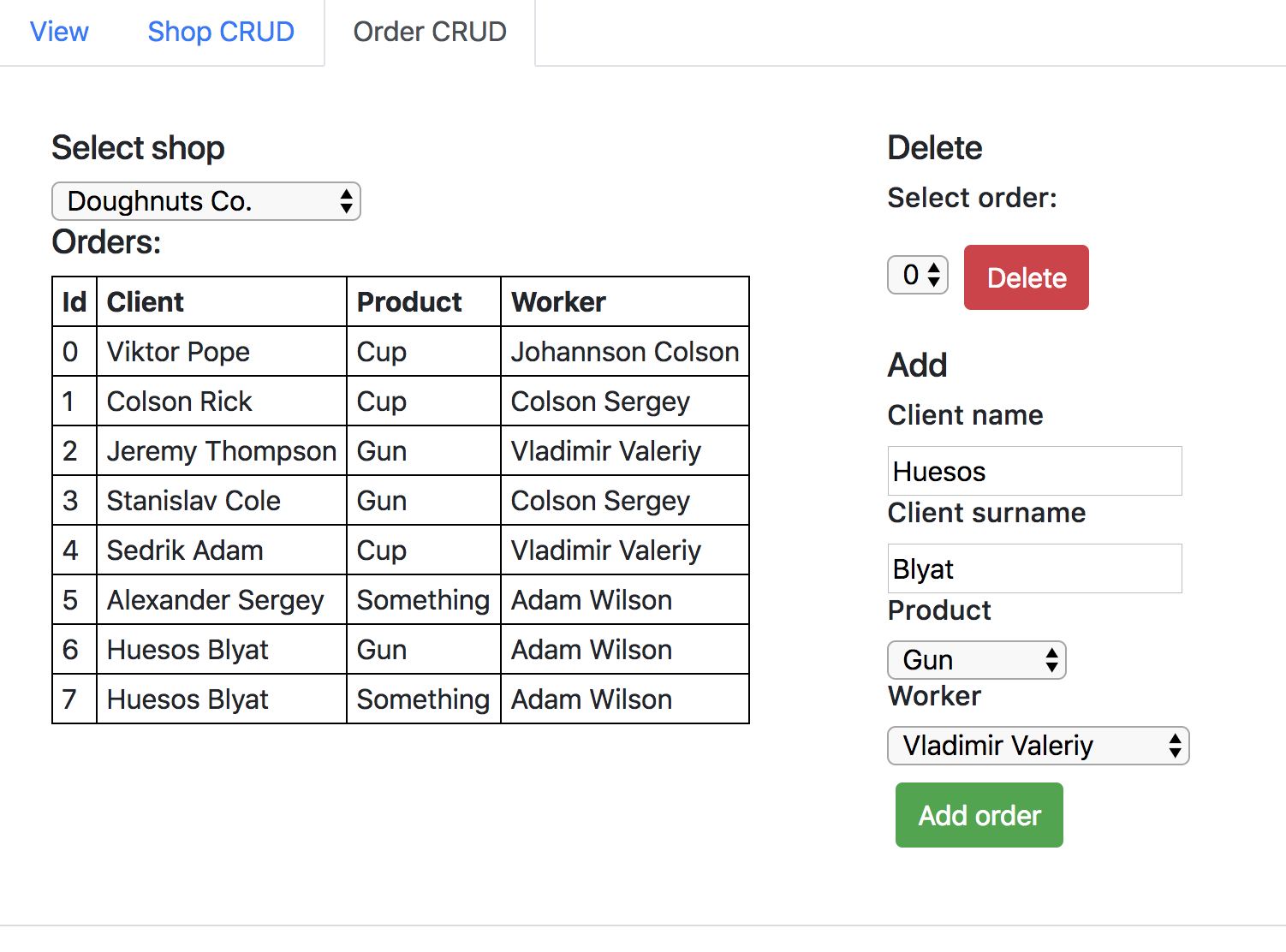


Рисунок 4. Завдання 2 (Замовлення)

**Висновок:** на даній лабораторній роботі я навчився працювати з класами в JavaScript ES6.