// Определяем интерфейс для элемента списка дел

interface TodoItem {

id: number;

name: string;

done: boolean;

createdDate: Date;

}

// Класс для хранения и управления списком дел

class TodoList {

private items: TodoItem[] = [];

// Добавление нового элемента

addItem(item: TodoItem) {

this.items.push(item);

}

// Удаление элемента по id

removeItemById(id: number) {

this.items = this.items.filter(item => item.id !== id);

}

// Удаление всех элементов

clearAllItems() {

this.items = [];

}

// Фильтрация списка по выполненным/невыполненным делам

filterItems(done: boolean | null) {

if (done === null) {

return this.items;

} else {

return this.items.filter(item => item.done === done);

}

}

// Сортировка списка по наименованию

sortItemsByName() {

this.items.sort((a, b) => a.name.localeCompare(b.name));

}

// Сортировка по дате создания

sortItemsByCreatedDate() {

this.items.sort((a, b) => a.createdDate.getTime() - b.createdDate.getTime());

}

// Сохранение списка в localstorage

saveItemsToLocalStorage() {

localStorage.setItem('todoList', JSON.stringify(this.items));

}

// Загрузка списка из localstorage

loadItemsFromLocalStorage() {

const data = localStorage.getItem('todoList');

if (data) {

this.items = JSON.parse(data);

}

}

}

// Пример использования

const todoList = new TodoList();

// Добавление элементов

const item1: TodoItem = { id: 1, name: 'Купить продукты', done: false, createdDate: new Date() };

const item2: TodoItem = { id: 2, name: 'Прочитать книгу', done: true, createdDate: new Date() };

todoList.addItem(item1);

todoList.addItem(item2);

// Удаление элемента по id

todoList.removeItemById(1);

// Очистка списка

todoList.clearAllItems();

// Сортировка элементов по наименованию

todoList.sortItemsByName();

// Фильтрация элементов

const filteredItems = todoList.filterItems(true); // Получить все выполненные дела

// Сохранение в localstorage

todoList.saveItemsToLocalStorage();

// Загрузка из localstorage

todoList.loadItemsFromLocalStorage();