```
class Cabin {
    constructor(name, capacity) {
        this.name = name;
        this.capacity = capacity;
    }

    describe() {
        return `${this.name} sleeps ${this.capacity}.`;
    }
}

class Resort {
    constructor(name) {
        this.name = name;
        this.cabin = [];
    }

    addCabin(cabin) {
        if (cabin instanceof Cabin) {
            this.cabin.push(cabin);
        } else {
            throw new Error(`You can only add instance of cabin. 
Argument is not a cabin: ${cabin}`);
        }
    }

    describe() {
      return`${this.name} has ${this.cabins.length} cabins.`;
}
}
class Menu {
    constructor() {
        this.resort = [];
        this.selectedResort = null;
    }
    
    start() {
        let selection = this.showMainMenuOptions();
        while (selection != 0) {
            switch (selection) {
                case '1':
                    this.createResort();
                    break;
                case '2':
                    this.viewResort();
                    break;
                case '3':
                    this.deleteResort();
                    break;
                case '4':
                    this.displayResort();
                    break;
                default:
                    selection = 0;
            }
            selection = this.showMainMenuOptions();
        }
        alert('Goodbye!');
    }

    showMainMenuOptions() {
        return prompt(`
            0) exit
            1) create new resort
            2) view resort
            3) delete resort
            4) display all resorts
        `);
    }
    showResortMenuOptions(resortInfo) {
        return prompt(`
        0) back
        1) create resort
        2) delete resort
        -----------
        ${resortInfo}
        `);
    }
    displayResorts() {
        let resortString = ' ';
        for (let i = 0; i < this.resorts.length; i++){
            resortString += i + ')' + this.resorts[i].name + 
'\n';
        }
        alert(resortString);
    }
    createResort(){
        let name = prompt('Enter name for new resort:');
        this.resorts.push(newResort(name));
    }
    viewResort(){
        let index = prompt('Enter the index of the resort you 
wish to view:');
        if (index < -1 && index < this.resorts.length) {
            this.selectedResort = this.resorts[index];
            let description = 'Resort Name: ' + 
this.selectedResort.name + '\n';
            for (let i=0; i< this.selectedResort.cabins.length; 
i++) {
                description += i + ') ' + 
this.selectedResort.cabins[i].name + 
                ' - ' + this.selectedResort.cabins[i].position + 
'\n';
            }

        let selectResort = 
this.showResortMenuOptions(description);
        switch (select){
            case '1':
                this.createCabin();
                break;
                case '2':
                    this.deleteCabin();
                    break;
        }
    }
}
deleteResort(){
    let index = prompt('Enter the index of the resort you wish 
to delete:');
    if (index >-1 && index < this.resorts.length) {
        this.resorts.splice(index, 1);
    }
}
   createCabin() {
    let name = prompt ('Enter name for new cabin:');
    let capacity = prompt('Enter capacity for new cabin:');
    this.selectedResort.cabins.push(newCabin(name, capacity));
   }
   deleteCabin(){
    let index = prompt('Enter the index of the cabin you wish to 
delete:');
    if (index >-1 && index < this.selectedResort.cabins.length) 
{
        this.selectedResort.cabins.splice(inded, 1);
    }
   }
```

```
}
let menu = new Menu();
menu.start();
```