Intro to JavaScript Week 5 Coding Assignment

Points possible: 75

URL to Your GitHub Repository:

https://github.com/anagalacticRuby/Week-05-Object_Oriented_Programming

Link to Video submission:

https://youtu.be/3_-enTgSAjs

Instructions: In VS Code, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your JavaScript project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Coding Steps:

1. Create a menu app as seen in this week's video. What you create is up to you as long as it meets the following requirements.
    a. Use at least one array.
    b. Use at least two classes.
    c. Your menu should have the options to create, view, and delete elements.

Screenshots of Code:

PROMINEO TECH

```
class Exhibit {
  //This class holds information about Exhibits of animals to be featured at
  Zoos.
  constructor(animal, population) {
    this.animal = animal;
    this.population = population;
    //This constructor takes the name of the animal and the population of
    that animal as parameters
  }
}

class Zoo {
  //The Zoo class will hold Exhibits of animals, and each Zoo has its own
  name.
  constructor(zooName) {
    this.zooName = zooName;
    this.exhibits = [];
    //This constructor method sets the name of the Zoo to what was input by
    the user
    //The array for exhibits is declared but not assigned a value yet, as it
    will be filled later by the user
  }

  addExhibit(exhibit) {
    if (exhibit instanceof Exhibit) {
      this.exhibits.push(exhibit);
    } else {
      throw new Error(
        `You can only add an instance of Exhibit. Argument is not a exhibit: $
        {exhibit}`
      );
      //If the user tries to add something that isn't an Exhibit to a Zoo,
      then an error is thrown
    }
  }
```

PROMINEO TECH

```
class Menu {
  constructor() {
    this.Zoos = [];
    this.selectedZoo = null;
    //Menu objects will hold an array that contains all of the Zoo objects
    (which hold their own exhibits)
    //selectedZoo is initialized as null because this program requires the
    use of a property which will be updated with user input.
    //Later, selectedZoo will hold a Zoo object.
  }
```

PROMINEO TECH

```
start() {
  let selection = this.showMainMenuOptions();
  //First showMainMenuOptions is called, causing a prompt to appear when
  the user runs this program
  //The user's input in that prompt is assigned to the selection variable,
  which will be fed into a switch statement
  //Based on the value of the selection variable, the program will decide
  what method of the Menu class to execute next.
  while (selection != 0) {
    switch (selection) {
      case "1":
        this.createZoo();
        break;
      case "2":
        this.editZoo();
        break;
      case "3":
        this.deleteZoo();
        break;
      case "4":
        this.displayZoos();
        break;
      default:
        selection = 0;
    }
    selection = this.showMainMenuOptions();
  }

  alert("Goodbye!");
  //If the user inputs 0 at the main menu, the application will exit
}
```

PROMINEO TECH

```
showMainMenuOptions() {
  return prompt(`
  0) Exit
  1) Create new Zoo
  2) Edit Zoo
  3) Delete Zoo
  4) Display all Zoos
  `);
  //This method shows the user a prompt with a list of options available
  for this program
  //showMainMenuOptions is called by start(), and whatever is input in this
  prompt will be returned to start() to select one of the options
}
```

```
displayZoos() {
  let zooString = "";
  for (let i = 0; i < this.Zoos.length; i++) {
    zooString += i + ") " + this.Zoos[i].zooName + "\n";
    if (this.Zoos[i].exhibits >= 1) {
      zooString += "\t" + "Featuring these exhibits: " + "\n";
    }
    for (let j = 0; j < this.Zoos[i].exhibits.length; j++) {
      zooString +=
        "\t" +
        j +
        ") " +
        this.Zoos[i].exhibits[j].animal +
        " with a population of " +
        this.Zoos[i].exhibits[j].population +
        "\n";
    }
  }
  //These loops will compile a list of all Zoos stored in the menu
  application as well as the exhibits of those Zoos and then display them
  to the user in an alert
  alert(zooString);
}
```

PROMINEO TECH

```
createZoo() {
  let zooName = prompt("Enter name for new Zoo:");
  this.Zoos.push(new Zoo(zooName));
  //createZoo prompts the user to input a name for the new Zoo object they
  will be creating
}

deleteZoo() {
  let listZoos = "";
  for (let i = 0; i < this.Zoos.length; i++) {
    listZoos += i + ") " + this.Zoos[i].zooName + "\n";
  }
  //This loop gathers a numbered list of the Zoos created by the user

  let index = prompt(`
  Existing Zoos:
  ${listZoos}
  ----
  Please enter the index of the Zoo you wish to delete:
  `);
  if (index > -1 && index < this.Zoos.length) {
    this.Zoos.splice(index, 1);
  }
  //Then after the list of existing Zoos has been compiled, the list is
  shown to the user within a prompt
  //This prompt then asks the user to input the index of the Zoo they wish
  to delete
}
```

PROMINEO TECH

```
editZoo() {
  //--

  let listZoos = "";
  for (let i = 0; i < this.Zoos.length; i++) {
    listZoos += i + ") " + this.Zoos[i].zooName + "\n";
  }

  let index = prompt(`
  Existing Zoos:
  ${listZoos}
  ----
  Please enter the index of the Zoo you wish to edit details of:
  `);
  //This prompt shows the user all existing Zoos and asks them to input the
  index of the Zoo they want to edit details of

  //--

  if (index > -1 && index < this.Zoos.length) {
    this.selectedZoo = this.Zoos[index];
    //Set the selectedZoo property of the current Menu object to the
    particular Zoo object specified by the user's input
    let description =
      "Zoo Name: " +
      this.selectedZoo.zooName +
      "\n" +
      "Has these animals: " +
      "\n";
```

```
  "Has these animals: " +
  "\n";

for (let i = 0; i < this.selectedZoo.exhibits.length; i++) {
  description +=
    i +
    ") " +
    this.selectedZoo.exhibits[i].animal +
    " with a population of " +
    this.selectedZoo.exhibits[i].population +
    "\n";
}
//Gather the exhibits featured at the user-specified Zoo and also the
population of those exhibits
let selection = this.showZooMenuOptions(description);
//Pass all of the information regarding the Zoo selected by the user
into the showZooMenuOptions method
switch (selection) {
  case "1":
    this.createExhibit();
    break;
  case "2":
    this.deleteExhibit();
    break;
}
```

PROMINEO TECH

```
showZooMenuOptions(ZooInfo) {
  return prompt(`
  0) back
  1) create exhibit
  2) delete exhibit
  --------
  ${ZooInfo}
  `);
  //showZooMenuOptions displays a series of options the user can pick from
  followed by the information about the Zoo they selected earlier
}

createExhibit() {
  let animal = prompt("Enter animal for new exhibit:");
  let population = prompt(" Enter population for new exhibit:");
  this.selectedZoo.addExhibit(new Exhibit(animal, population));
  //createExhibit asks the user to input an animal name and the population
  of that animal to be featured as an exhibit for the Zoo they selected
  earlier
}
```

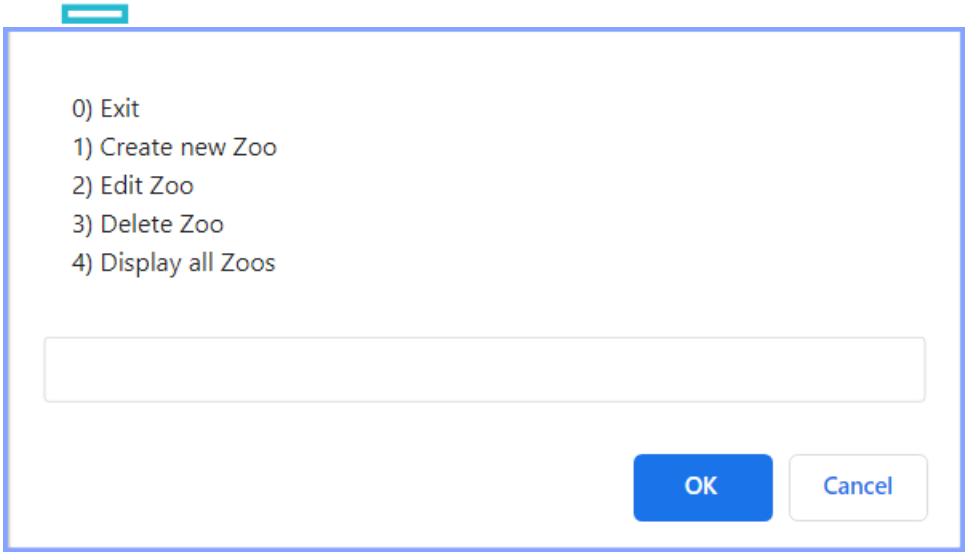
PROMINEO TECH

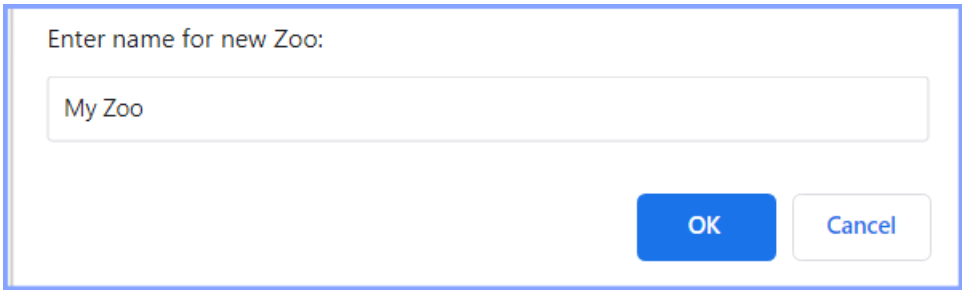
```
deleteExhibit() {
  let listExhibits = "";
  for (let i = 0; i < this.selectedZoo.exhibits.length; i++) {
    listExhibits += i + ") " + this.selectedZoo.exhibits[i].animal + " - "+
    this.selectedZoo.exhibits[i].population + "\n";
  }
  //Gather a list of exhibits that exist in the user's selected Zoo
  let index = prompt(`Existing exhibits:
  ${listExhibits}
  ----
  Please enter the index of the exhibit you wish to delete:"
  `);
  //Prompt the user with all existing exhibits of that Zoo along with
  instructions for the user to input an idex
  //That index input by the user will indicate which exhibit to delete

  //let index = prompt("Enter the index of the exhibit you wish to delete:
  ");
  if (index > -1 && index < this.selectedZoo.exhibits.length) {
    this.selectedZoo.exhibits.splice(index, 1);
  }
  //deleteExhibit asks the user to input the index of the exhibit they want
  to delete from the Zoo they selected earlier
}
```

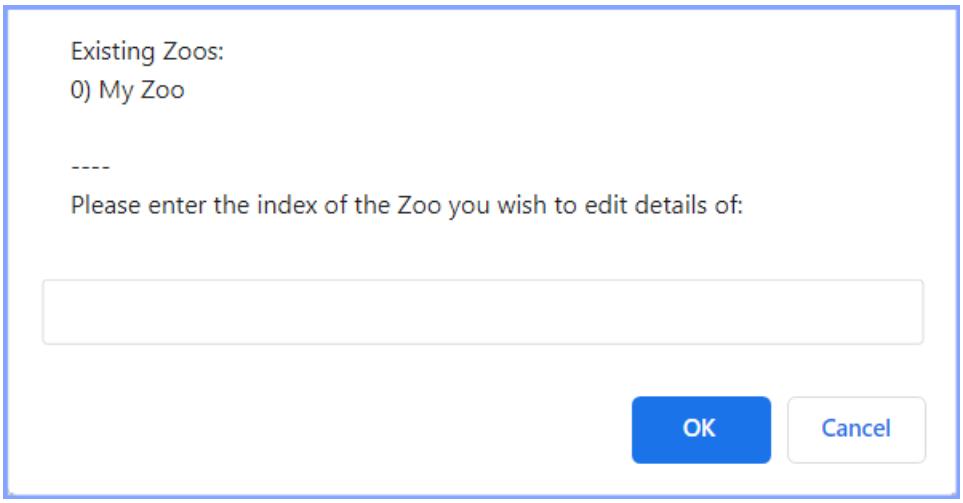
```
let menu = new Menu();
//In order for this program to actually do something, an instance of the Menu
object has to be created
menu.start();
//Then, with that new menu instance, start() is invoked, which will lead the
user through a series of prompts to use the menu application
```

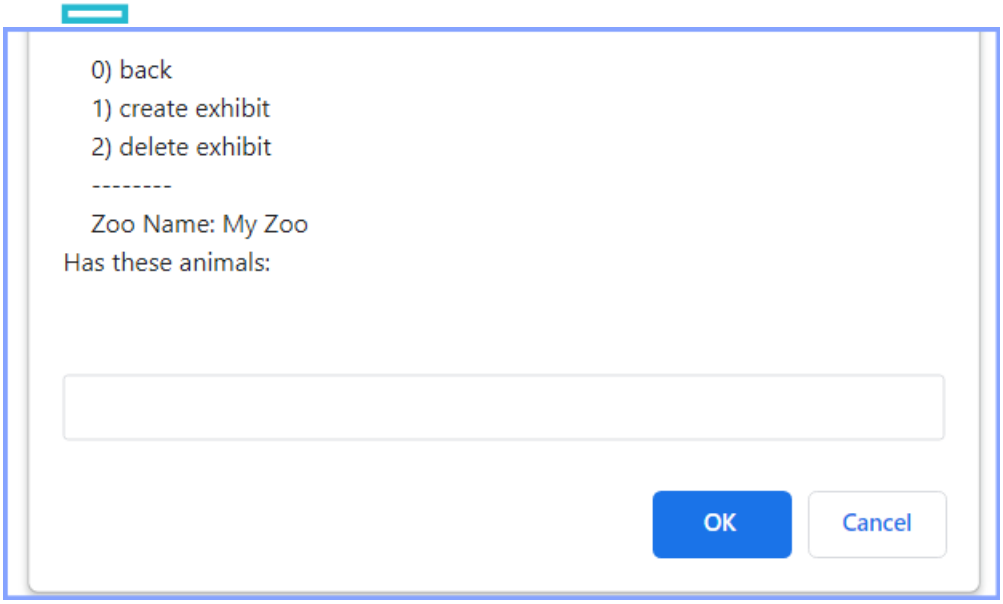
Screenshots of Running Application:

PROMINEO TECH

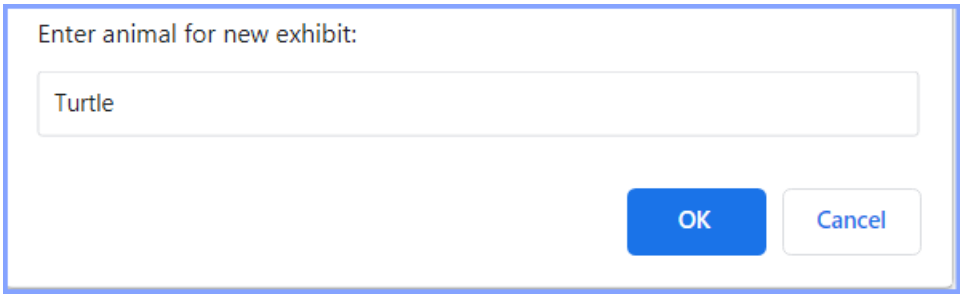


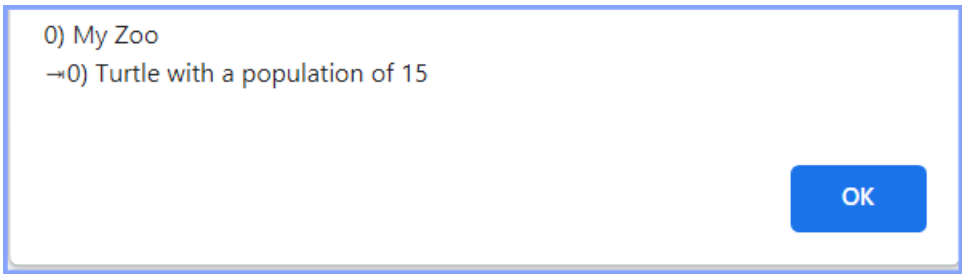


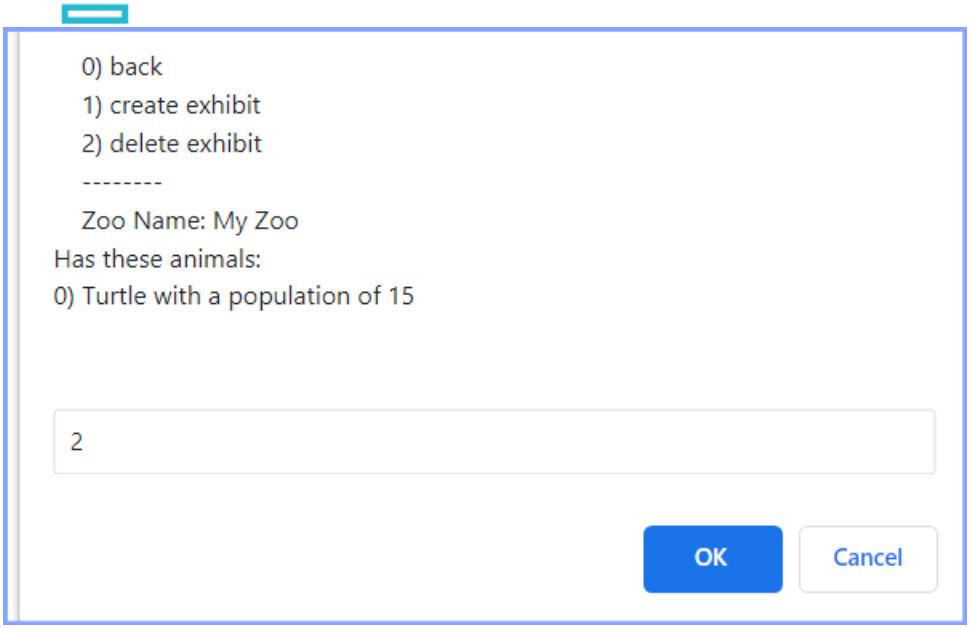



0) back
1) create exhibit
2) delete exhibit

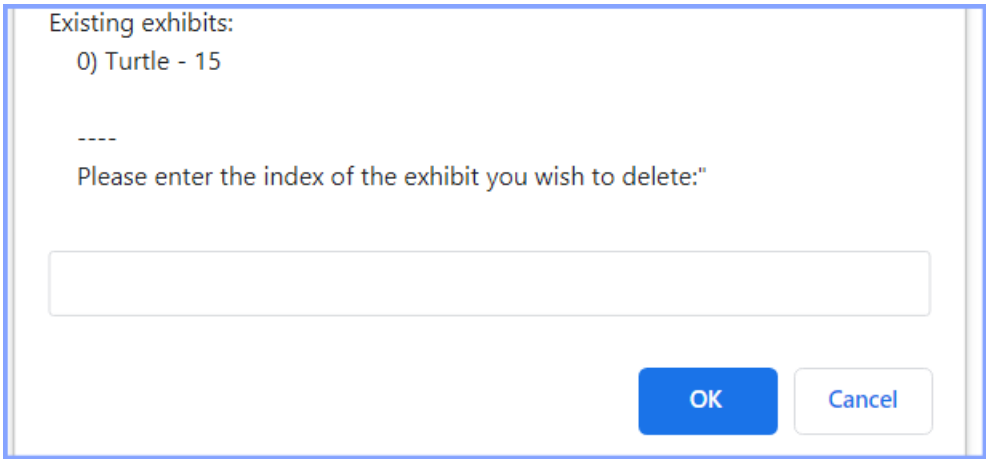
Zoo Name: My Zoo
Has these animals:
OK
Cancel


Enter animal for new exhibit:
Turtle
OK
Cancel


0) My Zoo
→0) Turtle with a population of 15
OK


0) back
1) create exhibit
2) delete exhibit

Zoo Name: My Zoo
Has these animals:
0) Turtle with a population of 15
2
OK
Cancel


Existing exhibits:
0) Turtle - 15

Please enter the index of the exhibit you wish to delete:"
OK
Cancel

PROMINEO TECH