**Лабораторна робота №9**

**Тема: ООП в JavaScript**

**Мета роботи:** вивчити особливості використання ООП в JavaScript

Хід роботи:

Завдання 1. Створіть клас "Модальне вікно" (Modal), який створює модальне вікно на сторінці. Клас повинен мати методи для відкриття, закриття та встановлення вмісту модального вікна. Зробіть можливість переміщення вікна по екрану за допомогою мишки (щоб можна було перетягувати за заголовок вікна).

Лістинг:

index.html

<body>

    <div class="container">

        <button>Відкрити модульне вікно</button>

    </div>

    <script src="script.js"></script>

</body>

style.css

body {

  margin: 0;

  font-size: 20px;

  overflow:hidden ;

}

.container {

  position: relative;

  height: 100vh;

}

.modal {

  position: absolute;

  width: 500px;

  height: 250px;

  border-radius: 3%;

  background-color: white;

  box-shadow: 1px 1px 5px black;

}

.windowMove {

  padding: 0 0 0 2px;

  display: flex;

  background-color: rgb(64, 64, 64);

  width: 100%;

  height: 30px;

  align-items: center;

  color: white;

  cursor: pointer;

  justify-content: space-between;

  font-size: 20px;

}

.windowText {

  max-height: 190px;

  padding: 10px;

  overflow-y: auto;

}

.overlay {

  display: none;

  position: absolute;

  top: 0;

  left: 0;

  height: 100vh;

  width: 100vw;

  background-color: rgba(141, 141, 141, 0.4);

  align-items: center;

  justify-content: center;

}

script.js

const container = document.querySelector(".container");

const btnOpen = document.querySelector("button");

class Modal {

  constructor(btnOpen) {

    this.overlay = document.createElement("div");

    this.overlay.classList.add("overlay");

    this.modal = document.createElement("div");

    this.windowMove = document.createElement("div");

    this.windowText = document.createElement("div");

    this.btnExit = document.createElement("div");

    this.title = document.createElement("div");

    this.btnExit.innerText = "❌";

    this.title.innerText = "Модульне вікно";

    this.windowText.innerText = "Це модульне вікно";

    this.btnExit.addEventListener("click", () => this.Close());

    this.modal.classList.add("modal");

    this.windowMove.classList.add("windowMove");

    this.windowText.classList.add("windowText");

    this.windowMove.appendChild(this.title);

    this.windowMove.appendChild(this.btnExit);

    this.modal.appendChild(this.windowMove);

    this.modal.appendChild(this.windowText);

    this.overlay.appendChild(this.modal);

    container.appendChild(this.overlay);

    btnOpen.addEventListener("click", () => this.Open());

  }

  Close() {

    this.overlay.style.display = "none";

  }

  Open() {

    this.overlay.style.display = "flex";

  }

  setContent(title, content) {

    this.title.innerText = title;

    this.windowText.innerText = content;

  }

}

let mouseCoord = { x: 0, y: 0 };

let isMove = false;

let element = null;

document.documentElement.addEventListener("mousedown", (event) => {

  event.preventDefault();

  if (!event.target.classList.contains("windowMove")) return;

  element = event.target.closest(".modal");

  mouseCoord.x = event.clientX;

  mouseCoord.y = event.clientY;

  isMove = true;

});

document.documentElement.addEventListener("mousemove", (event) => {

  event.preventDefault();

  if (!isMove || !element) return;

  let coordX = event.clientX;

  let coordY = event.clientY;

  let dx = coordX - mouseCoord.x;

  let dy = coordY - mouseCoord.y;

  mouseCoord.x = coordX;

  mouseCoord.y = coordY;

  moveElement(dx, dy);

});

document.documentElement.addEventListener("mouseup", () => {

  isMove = false;

  element = null;

});

function moveElement(dx, dy) {

  let leftCoord = parseInt(getComputedStyle(element).left);

  let topCoord = parseInt(getComputedStyle(element).top);

  let elementWidth = element.clientWidth;

  let elementHeight = element.clientHeight;

  let containerWidth = container.clientWidth;

  let containerHeight = container.clientHeight;

  leftCoord += dx;

  topCoord += dy;

  if (leftCoord < 0) leftCoord = 0;

  if (topCoord < 0) topCoord = 0;

  if (leftCoord + elementWidth > containerWidth)

    leftCoord = containerWidth - elementWidth;

  if (topCoord + elementHeight > containerHeight)

    topCoord = containerHeight - elementHeight;

  element.style.left = leftCoord + "px";

  element.style.top = topCoord + "px";

}

const modal = new Modal(btnOpen);

titleHistory = "Місто Снів";

textHistory =

  "У маленькому містечку, де люди ніколи не спали, з'явився незвичайний чоловік на ім'я Андрій, який пропонував 'Еліксир Снів'. " +

  "Цей еліксир обіцяв яскраві сни, що змінюють життя.\n Містяни, спочатку скептичні, швидко зацікавилися та почали випробовувати його." +

  "Сни були чудові, але після пробудження вони поверталися до сірої реальності, відчуваючи невдоволення." +

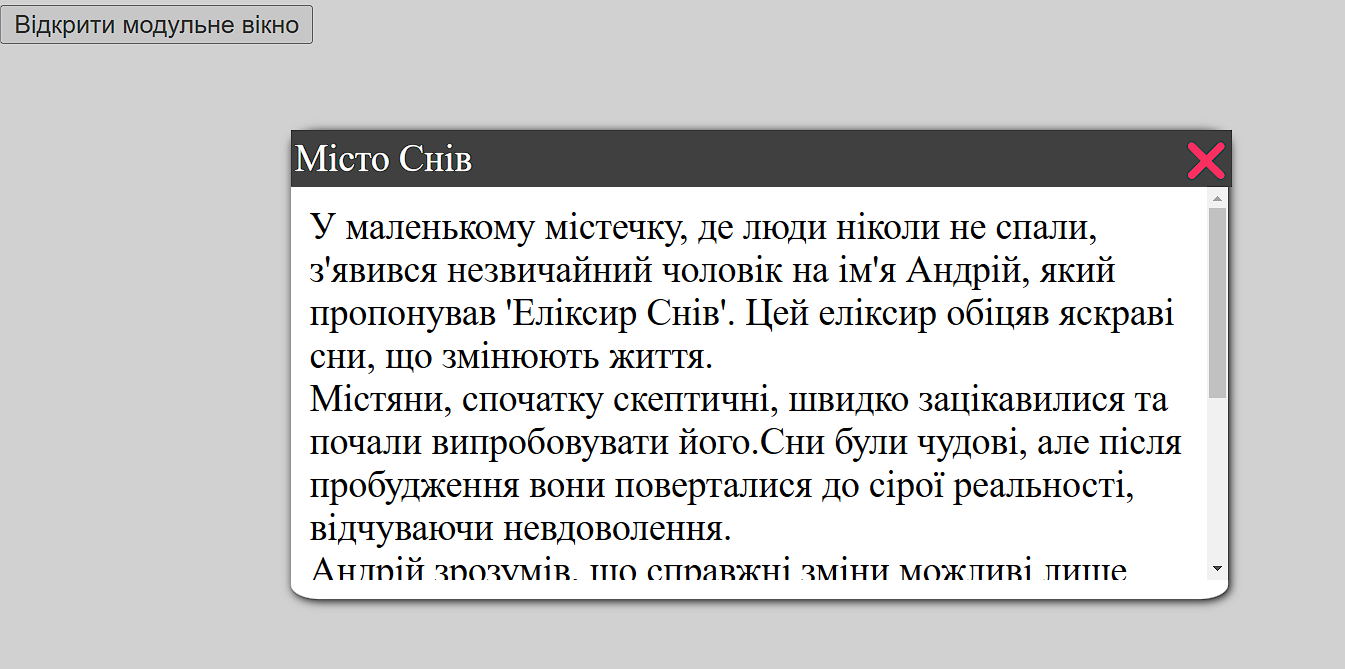
  "\n Андрій зрозумів, що справжні зміни можливі лише через зусилля, і вирішив навчити містян, як втілювати свої мрії в реальність." +

  "З часом вони почали працювати над своїми бажаннями, відкрили нові бізнеси та зайнялися творчістю. Місто, яке колись було бездушним," +

  "стало яскравим центром натхнення, де мрії здійснювалися, і кожен знав, що справжні сни можна створити самостійно.";

modal.setContent(titleHistory, textHistory);

Виконання програми:



Завдання 2. Створіть клас "Слайдер" (Slider), який дозволяє створювати горизонтальний або вертикальний слайдер з можливістю перегляду зображень або вмісту. Клас повинен мати методи для додавання слайдів та переміщення між ними.

Лістинг:

index.html

  <body>

    <div class="container"></div>

    <div class="container"></div>

    <script src="script.js"></script>

  </body>

style.css

body {

  display: flex;

}

.container {

  position: relative;

  width: 300px;

  height: 300px;

  overflow: hidden;

  margin-right: 10px;

}

.img-slider {

  display: flex;

  transition: transform 0.5s;

}

img {

  width: 300px;

  height: 300px;

  object-fit: cover;

}

button {

  position: absolute;

  background-color: rgba(0, 0, 0, 0.5);

  color: white;

  border: none;

  cursor: pointer;

}

.horizontal button {

  padding: 20px 10px 20px 10px;

  top: 50%;

  transform: translateY(-50%);

}

.vertical button {

  padding: 10px 20px 10px 20px;

  left: 50%;

  transform: translateX(-50%);

}

script.js

const container1 = document.querySelectorAll(".container")[0];

const container2 = document.querySelectorAll(".container")[1];

class Slider {

  constructor(container, horizontal = true) {

    this.currentIndex = 0;

    this.horizontal = horizontal;

    this.imgSlider = document.createElement("div");

    this.imgSlider.classList.add("img-slider");

    this.createImg(this.imgSlider);

    this.nextBtn = document.createElement("button");

    this.prevBtn = document.createElement("button");

    if (this.horizontal) {

      this.imgSlider.style.flexDirection = "row";

      this.nextBtn.innerText = "⮞";

      this.prevBtn.innerText = "⮜";

      container.classList.add("horizontal");

      this.nextBtn.style.right = "10px";

      this.prevBtn.style.left = "10px";

    } else {

      this.imgSlider.style.flexDirection = "column";

      this.nextBtn.innerText = "⮟";

      this.prevBtn.innerText = "⮝";

      container.classList.add("vertical");

      this.nextBtn.style.bottom = "10px";

      this.prevBtn.style.top = "10px";

    }

    this.nextBtn.addEventListener("click", () => this.next());

    this.prevBtn.addEventListener("click", () => this.prev());

    container.appendChild(this.imgSlider);

    container.appendChild(this.nextBtn);

    container.appendChild(this.prevBtn);

    this.images = this.imgSlider.querySelectorAll("img");

  }

  next() {

    this.currentIndex = (this.currentIndex + 1) % this.images.length;

    this.updateTransform();

  }

  prev() {

    this.currentIndex =

      (this.currentIndex - 1 + this.images.length) % this.images.length;

    this.updateTransform();

  }

  updateTransform() {

    if (this.horizontal) {

      this.imgSlider.style.transform = `translateX(-${

        this.currentIndex \* 300

      }px)`;

    } else {

      this.imgSlider.style.transform = `translateY(-${

        this.currentIndex \* 300

      }px)`;

    }

  }

  createImg(imgContainer) {

    for (let i = 0; i < 9; i++) {

      let img = document.createElement("img");

      img.setAttribute("src", `img/${i + 1}.jpg`);

      img.setAttribute("alt", `img${i + 1}`);

      img.style.width = "300px";

      imgContainer.appendChild(img);

    }

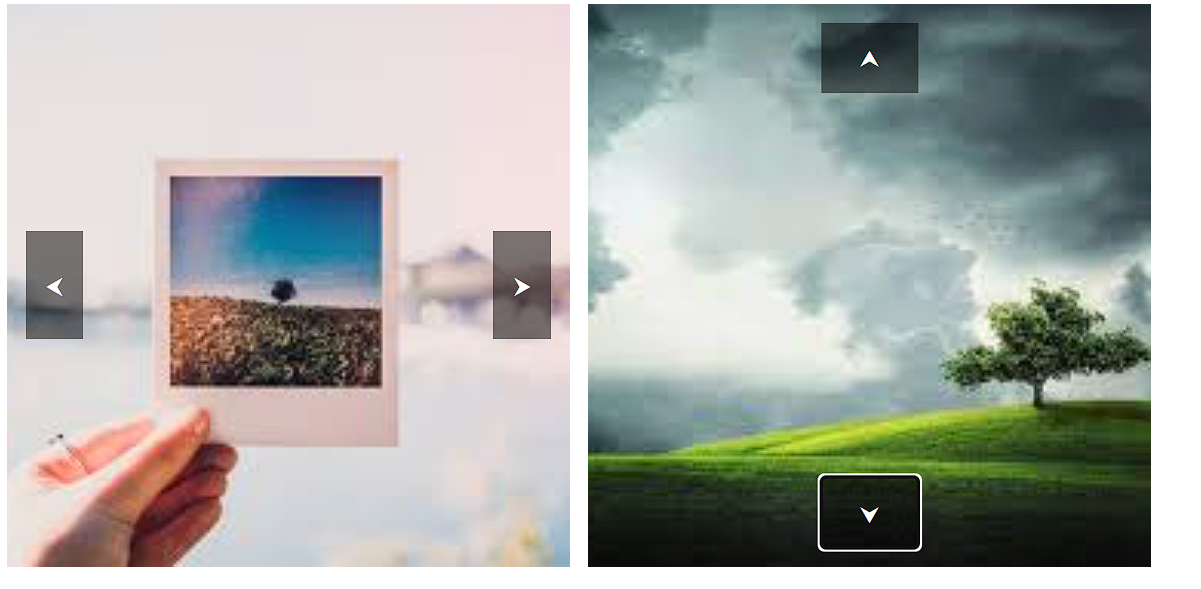
  }

}

const slider1 = new Slider(container1, true);

const slider2 = new Slider(container2, false);

Виконання програми:



Завдання 3. Створіть клас "Таблиця" (Table), який генерує HTML-таблицю на основі заданих даних. Клас повинен мати методи для додавання рядків та стовпців до таблиці.

Лістинг:

index.html

  <body>

    <div class="container"></div>

    <script src="script.js"></script>

  </body>

style.css

table {

  border-collapse: collapse;

  border: 1px solid black;

  margin-bottom: 20px;

  td {

    border: 1px solid black;

    padding: 5px;

  }

  tr:first-child {

    background-color: rgb(134, 134, 134);

    color: white;

  }

}

script.js

class Table {

  constructor(container, headers = []) {

    this.table = document.createElement("table");

    this.headers = headers;

    this.addRow(headers);

    container.appendChild(this.table);

  }

  addRow(info = []) {

    const tr = document.createElement("tr");

    info.forEach((cellInfo) => {

      const td = document.createElement("td");

      td.innerText = cellInfo;

      tr.appendChild(td);

    });

    this.table.appendChild(tr);

  }

  addColumn(title) {

    const tr = this.table.querySelector("tr");

    const tdTitle = document.createElement("td");

    tdTitle.innerText = title;

    tr.appendChild(tdTitle);

  }

}

const container = document.querySelector(".container");

const table = new Table(container, ["Ім'я", "Факультет", "Група"]);

table.addColumn("Місто");

table.addRow(["Віталій", "ФІКТ", "ВТк-24-1", "Житомир"]);

table.addRow(["Василь", "ФІКТ", "ІПЗк-24-1", "Любар"]);

table.addRow(["Богдан", "ФІКТ", "ІПЗк-24-1", "Житомир"]);

table.addRow(["Назар", "ФБСО", "ГРС-7", "Київ"]);

Виконання програми:



Завдання 4. Створіть клас "Форма" (Form), який дозволяє створювати HTML-форми з заданими полями вводу, кнопками та обробниками подій для них. Клас повинен мати методи для обробки відправки форми та валідації введених даних.

Лістинг:

index.html

  <body>

    <div class="container"></div>

    <script src="script.js"></script>

  </body>

style.css

button {

  margin-top: 5px;

}

form {

  width: 600px;

}

div{

  display: flex;

  width: 550px;

}

label {

  display: block;

  width: 60px;

}

script.js

const container = document.querySelector(".container");

class Form {

  constructor(container, object = []) {

    this.container = container;

    this.object = object;

    this.form = document.createElement("form");

    this.createElements();

    this.btnSubmit = document.createElement("button");

    this.btnSubmit.innerText = "Відправити";

    this.btnSubmit.type = "button";

    this.btnSubmit.addEventListener("click", () => {

      if (this.validate()) {

        this.submit();

      } else {

        alert("Будь ласка, заповніть всі поля");

      }

    });

    this.form.appendChild(this.btnSubmit);

    this.container.appendChild(this.form);

  }

  createElements() {

    this.object.forEach((element) => {

      const div = document.createElement("div");

      const label = document.createElement("label");

      label.setAttribute("for", element.name);

      label.innerText = element.label;

      label.classList.add("width");

      const input = document.createElement("input");

      input.type = element.type;

      input.name = element.name;

      input.id = element.id;

      input.placeholder = element.placeholder || "";

      div.appendChild(label);

      div.appendChild(input);

      this.form.appendChild(div);

    });

  }

  submit() {

    const data = this.form.querySelectorAll("input");

    let text = "";

    data.forEach((input) => {

      const label = this.form.querySelector(`label[for="${input.name}"]`);

      text += label.innerText + ": " + input.value + "\n";

    });

    alert("Форму надіслано\n" + text);

  }

  validate() {

    const data = this.form.querySelectorAll("input");

    let valid = true;

    data.forEach((input) => {

      if (input.value === "") {

        valid = false;

      }

    });

    return valid;

  }

}

const form = new Form(container, [

  {

    name: "name",

    label: "Ім'я:",

    id: "name",

    type: "text",

    placeholder: "Введіть ім'я",

  },

  {

    name: "password",

    label: "Пароль:",

    id: "password",

    type: "password",

    placeholder: "Введіть пароль",

  },

  {

    name: "email",

    label: "Пошта:",

    id: "email",

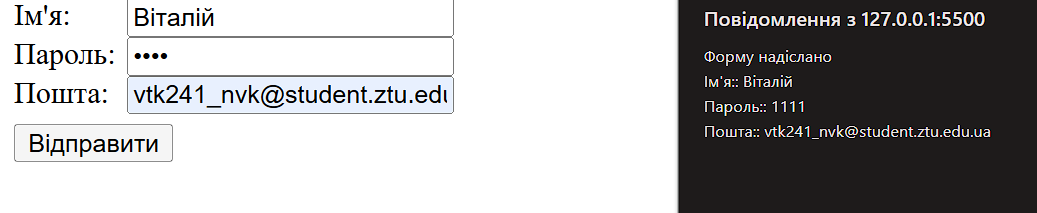
    type: "email",

    placeholder: "Введіть пошту",

  },

]);

Виконання програми:



Завдання 5. Створіть клас "Вкладка" (Tab), який створює вкладки для переключення між різними блоками вмісту. Клас повинен мати методи для додавання вкладок та вмісту для них, а також для переключення між вкладками.

Лістинг:

index.html

 <body>

    <div class="container"></div>

    <script src="script.js"></script>

  </body>

style.css

.container {

  position: relative;

}

.tab {

  position: absolute;

  padding: 15px;

  display: flex;

  background-color: rgb(196, 196, 196);

  border: 1px solid black;

  box-shadow: 5px 5px 5px black;

  z-index: 1;

  transition: left 0.5s ease;

}

button {

  padding: 6.5px;

  background-color: rgba(0, 0, 0, 0.5);

  color: white;

  border: none;

  cursor: pointer;

}

script.js

class Tab {

  constructor(container) {

    this.container = container;

    this.tabs = [];

    this.contents = [];

    this.activeIndex = 0;

  }

  addTab(title, content) {

    const ind = this.contents.length;

    this.tabs.push(title);

    const tabButton = document.createElement("button");

    tabButton.innerText = title;

    tabButton.addEventListener("click", () => this.switch(ind));

    this.container.appendChild(tabButton);

    const contentDiv = document.createElement("div");

    contentDiv.innerText = content;

    contentDiv.classList.add("tab");

    contentDiv.style.left = 50 \* ind + "px";

    contentDiv.style.zIndex = 5 - ind;

    this.container.appendChild(contentDiv);

    this.contents.push(contentDiv);

  }

  switch(index) {

    this.contents[this.activeIndex].style.zIndex = 5 - this.activeIndex;

    const tmp = this.contents[this.activeIndex].style.left;

    this.contents[this.activeIndex].style.left =

      this.contents[index].style.left;

    this.contents[index].style.left = tmp;

    this.contents.forEach((content) => {

      if (content.style.left == "50px") {

        content.style.zIndex = 9;

      }

    });

    this.contents[index].style.zIndex = 10;

    this.activeIndex = index;

  }

}

const Container = document.querySelector(".container");

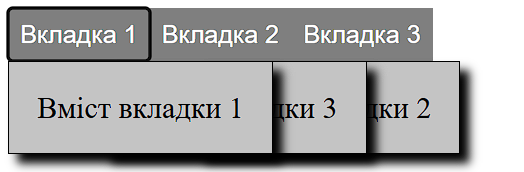
const tabs = new Tab(Container);

tabs.addTab("Вкладка 1", "Вміст вкладки 1");

tabs.addTab("Вкладка 2", "Вміст вкладки 2");

tabs.addTab("Вкладка 3", "Вміст вкладки 3");

Виконання програми:



Завдання 6. Створіть клас "Сповіщення" (Notification), який дозволяє створювати спливаючі повідомлення на сторінці. Клас повинен мати методи для створення сповіщень з різними стилями та для їх закриття.

Лістинг:

index.html

 <body>

    <script src="script.js"></script>

  </body>

style.css

.success{

    background-color: rgb(87, 189, 87);

}

.error{

    background-color: red;

}

.warning{

    background-color: orange;

}

.container{

    position: absolute;

    left: 76%;

}

.notification{

    display: flex;

    justify-content: space-between;

    width: 250px;

    padding: 10px 20px 10px 20px;

    margin-bottom: 10px;

    border-radius: 3px;

    color: white;

    cursor: pointer;

    font-size: 17px;

}

script.js

class Notification {

  constructor() {

    this.container = document.createElement("div");

    this.container.classList.add("container");

    document.body.appendChild(this.container);

  }

  create(message, type) {

    const btnShow = document.createElement("button");

    btnShow.innerText = "Вивести повідомлення: " + type;

    document.body.appendChild(btnShow);

    const notification = document.createElement("div");

    notification.classList.add("notification", type);

    const divText = document.createElement("div");

    const divClose = document.createElement("div");

    divText.innerText = message;

    divClose.innerText = "×";

    btnShow.addEventListener("click", () => this.open(notification));

    divClose.addEventListener("click", () => this.close(notification));

    notification.appendChild(divText);

    notification.appendChild(divClose);

  }

  open(notification) {

    if (!this.container.contains(notification)) {

      this.container.appendChild(notification);

    }

  }

  close(notification) {

    if (this.container.contains(notification)) {

      this.container.removeChild(notification);

    }

  }

}

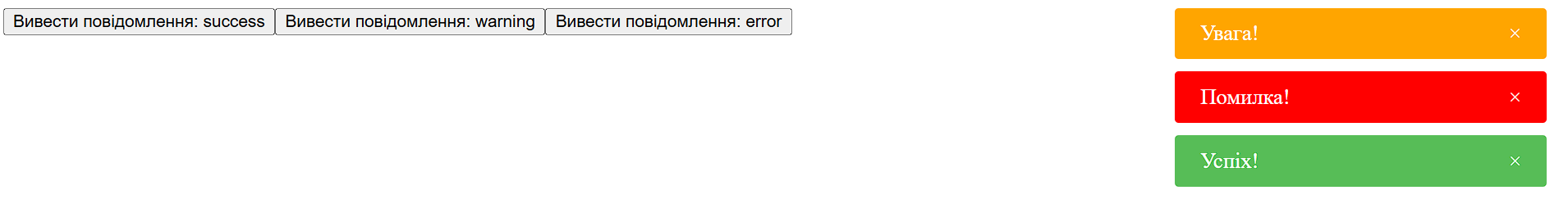
const notification = new Notification();

notification.create("Успіх!", "success");

notification.create("Увага!", "warning");

notification.create("Помилка!", "error");

Виконання програми:



Висновки: В ході виконання лабораторної роботи було вивчено особливості використання ООП в JavaScript.