

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота №** 1

з дисципліни “Основи web-програмування”

тема “**Основи JavaScript**”

|  |  |  |
| --- | --- | --- |
| Виконав  студент II курсу  групи КП-92  Гришаєв Дмитро Ігорович  (*прізвище, ім’я, по батькові*)  варіант №4 |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Гадиняк Руслан Анатолійович  (*прізвище, ім’я, по батькові*) |

Київ 2020

**Мета роботи**

Познайомитись з мовою програмування JavaScript та середовищем Node.js.

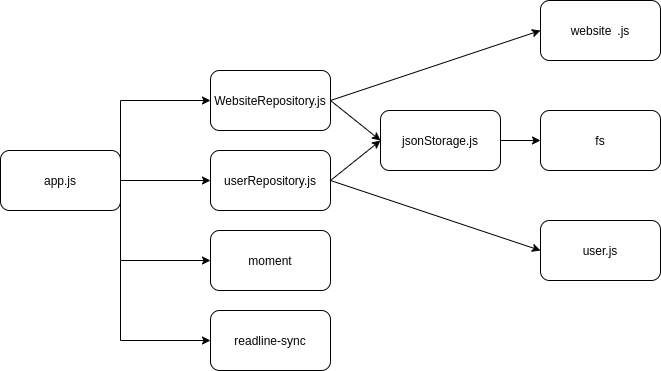
Навчитись створювати модулі, синхронно працювати із файловими потоками та обробляти JSON-текст.

**Постановка завдання**

Реалізувати на мові JavaScript (або TypeScript) консольну програму для керування двома типами сутностей із файлового сховища у форматі JSON: сутність "Користувач системи" та сутність за варіантом.

В якості сутностей за варіантом використати основну або додаткову сутність за темою вашої курсової роботи з "Основ програмування".

**Діаграма залежності модулів**

****

**Код всіх розроблених модулів**

|  |
| --- |
| **app.js** |
| const UserRepository = require('./repositories/userRepository');  const userRepository = new UserRepository("data/users.json");  const WebsiteRepository = require('./repositories/WebsiteRepository');  const websiteRepository = new WebsiteRepository("./data/Websites.json");  const Websites = require('./models/websites');  const moment = require('moment');  const readlineSync = require('readline-sync');      while(1)  {  const data = readlineSync.question('Enter your command: ');  try{  const text = data.toString().trim();  const cmd = text.split('/');  const available\_start\_cmd = ['get', 'delete', 'update', 'post'];  const available\_get\_cmd = ['users', 'websites'];  if (cmd.length < 2 || !available\_start\_cmd.includes(cmd[0])) {  throw new Error('Invalid command');  }  if (cmd[0] === 'get') {  if (!available\_get\_cmd.includes(cmd[1])) {  throw new Error('Invalid command');  }  if (cmd[1] === 'users') {  if (cmd.length === 2) {  console.log('Success! We are finding all users!');  const users = userRepository.getUsers();  if(users.length === 0){  console.log('There are no users');  }  else{  console.log(users);  }  }  else {  if (!isNaN(cmd[2])) {  const uid = parseInt(cmd[2]);  console.log(`Success! We are finding id ${uid}!`);  const user = userRepository.getUserById(uid);  if(user === null){  console.log(`There is no user with id ${uid}`);  }  else{  console.log(user);  }  }  else {  throw new Error('Invalid id');  }  }  }  if (cmd[1] === 'websites') {  if (cmd.length === 2) {  console.log('Success! We are finding all websites!');  const websites = websiteRepository.getWebsites();  if(websites.length === 0){  console.log('There are no websites');  }  else{  console.log(websites);  }  }  else {  if (!isNaN(cmd[2])) {  const wid = parseInt(cmd[2]);  console.log(`Success! We are finding website with id ${wid}!`);  const website = websiteRepository.getWebsitesById(wid);  if(website === undefined){  console.log(`There is no website with id ${wid}`);  }  else{  console.log(website);  }  }  else {  throw new Error('Invalid id');  }  }  }  }  if (cmd[0] === 'delete') {  if (cmd.length !== 3 || cmd[1] !== 'websites') {  throw new Error('Invalid command');  }  if (isNaN(cmd[2])) {  throw new Error('Invalid id');  }  const dwid = parseInt(cmd[2]);  console.log(`Success! We are deleting website with id ${dwid}!`);  const del = websiteRepository.deleteWebsites(dwid);  if(del === undefined){  console.log(`There is no website with id ${dwid}`)  }  else{  console.log(`Website with id ${cmd[2]} has been deleted`);  }  }  if (cmd[0] === 'update') {  let date\_of\_creating = new Date;  if (cmd[1] !== 'websites') {  throw new Error('Invalid command');  }  const id = parseInt(cmd[2]);  if(websiteRepository.getWebsitesById(id)===undefined){  throw new Error('Invalid id');  }  const name = readlineSync.question('Input name of the website: ');  const address = readlineSync.question('Input address of the website: ');  if (!address.includes('.')) {  throw new Error('Invalid address');  }  const visits\_per\_day = readlineSync.question('Input visits\_per\_day of the website: ');  if (isNaN(visits\_per\_day)) {  throw new Error('Invalid visits\_per\_day');  }  const count\_of\_ad\_bunners = readlineSync.question('Input count\_of\_ad\_bunners of the website: ');  if (isNaN(count\_of\_ad\_bunners)) {  throw new Error('Invalid count\_of\_ad\_bunners');  }  date\_of\_creating = readlineSync.question('Input date\_of\_creating of the website: ');  if (!moment(date\_of\_creating, moment.ISO\_8601).isValid()) {  throw new Error('Invalid date');  }  console.log(`Success! We are updating website with id ${id}!`);  const site = new Websites(id, name, address, visits\_per\_day, count\_of\_ad\_bunners, date\_of\_creating);  const result = websiteRepository.updateWebsite(site);  if(result!== undefined){  console.log(`Website with id ${id} has been updated`);  }  }  if (cmd[0] === 'post') {  let date\_of\_creating = new Date;  if (cmd[1] !== 'websites' || cmd.length !== 2) {  throw new Error('Invalid command');  }  const name = readlineSync.question('Input name of the website: ');  const address = readlineSync.question('Input address of the website: ');  if (!address.includes('.')) {  throw new Error('Invalid address');  }  const visits\_per\_day = readlineSync.question('Input visits\_per\_day of the website: ');  if (isNaN(visits\_per\_day)) {  throw new Error('Invalid visits\_per\_day');  }  const count\_of\_ad\_bunners = readlineSync.question('Input count\_of\_ad\_bunners of the website: ');  if (isNaN(count\_of\_ad\_bunners)) {  throw new Error('Invalid count\_of\_ad\_bunners');  }  date\_of\_creating = readlineSync.question('Input date\_of\_creating of the website: ');  if (!moment(date\_of\_creating, moment.ISO\_8601).isValid()) {  throw new Error('Invalid date');  }  console.log(`Success! We are adding a new website!`);  const site = new Websites(0, name, address, visits\_per\_day, count\_of\_ad\_bunners, date\_of\_creating);  const addres = websiteRepository.addWebsite(site);  if(addres!== undefined){  console.log(`A new website has been added`);  }  }  }  catch(err){  console.error(err);  }  } |

|  |
| --- |
| **WebsiteRepository.js** |
| const JsonStorage = require('../jsonStorage');  const Website = require('../models/websites');  class UserRepository {    constructor(filePath) {  this.storage = new JsonStorage(filePath);  }  addWebsite(entityModel) {  const items = this.storage.readItems();  let jsonvalue = this.storage.readId();  entityModel.id = jsonvalue.id;  jsonvalue.id++;  this.storage.writeitems("./data/nextId.json", JSON.stringify(jsonvalue));  items.push(entityModel);  this.storage.writeitems("./data/Websites.json", JSON.stringify(items, null, 4));  return jsonvalue.id--;  }  getWebsites() {  const items = this.storage.readItems();  const arr = [];  for(let i of items){  arr.push(new Website(i.id, i.name, i.address, i.visits\_per\_day, i.count\_of\_ad\_bunners, i.date\_of\_creating))  }  return arr;  }  getWebsitesById(entityId) {  const items = this.storage.readItems();  for (let i of items) {  if (entityId === i.id) {  return new Website(i.id, i.name, i.address, i.visits\_per\_day, i.count\_of\_ad\_bunners, i.date\_of\_creating);  }  }  }  updateWebsite(entityModel) {  let items = this.storage.readItems();  for (let i = 0; i < items.length; i++) {  if (entityModel.id === items[i].id) {  items[i] = entityModel;  this.storage.writeitems("./data/Websites.json", JSON.stringify(items, null, 4));  return items[i]  }  }  }  deleteWebsites(entityId) {  let items = this.storage.readItems();  for (let i = 0; i < items.length; i++) {  if (entityId === items[i].id) {  items.splice(i, i + 1);  this.storage.writeitems("./data/Websites.json", JSON.stringify(items, null, 4));  let jsonvalue = this.storage.readId();  jsonvalue.id--;  this.storage.writeitems("./data/nextId.json", JSON.stringify(jsonvalue));  return entityId;  }  }  }  };    module.exports = UserRepository; |

|  |
| --- |
| **userRepository.js** |
| const User = require('../models/user');  const JsonStorage = require('../jsonStorage');  class UserRepository {  constructor(filePath) {  this.storage = new JsonStorage(filePath);  }  getUsers() {  const items = this.storage.readItems();  const arr = [];  for(let i of items){  arr.push(new User(i.id, i.login, i.fullname, i.role, i.registeredAt, i.avaUrl, i.isEnabled))  }  return arr;  }  getUserById(id) {  const items = this.storage.readItems();  for (let item of items) {  if (item.id === id) {  return new User(item.id, item.login, item.fullname, item.role, item.registeredAt, item.avaUrl, item.isEnabled);  }  }  return null;  }  };      module.exports = UserRepository; |

|  |
| --- |
| **jsonStorage.js** |
| const fs = require('fs');  class JsonStorage {  // filePath - path to JSON file  constructor(filePath) {  this.filePath = filePath;  }  readItems() {  const jsonText = fs.readFileSync(this.filePath);  const jsonArray = JSON.parse(jsonText);  return jsonArray;  }  readId(){  let jsonText = fs.readFileSync('./data/nextId.json');  return JSON.parse(jsonText);  }  writeItems(items){  fs.writeFileSync(this.fath, items);  }  writeId(items){  fs.writeFileSync(./data/nextId.json, items);  }  };      module.exports = JsonStorage; |

|  |
| --- |
| **websites.js** |
| class Websites {      constructor(id, name, address, visits\_per\_day, count\_of\_ad\_bunners, date\_of\_creating) {  this.id = id; // id  this.name = name; // string  this.address = address; // string  this.visits\_per\_day = visits\_per\_day; //number  this.count\_of\_ad\_bunners = count\_of\_ad\_bunners;//number  this.date\_of\_creating = date\_of\_creating;// Date ISO 8601  }  };      module.exports = Websites; |

|  |
| --- |
| **user.js** |
| class User {      constructor(id, login, fullname, role, registeredAt, avaUrl, isEnabled) {  this.id = id; // number  this.login = login; // string  this.fullname = fullname; // string  this.role = role; //number 0 || 1  this.registeredAt = registeredAt;// Date ISO 8601  this.avaUrl = avaUrl;//strign (URL)  this.isEnabled = isEnabled; //bool  }  };  module.exports = User; |

**Вміст .json файлів**

|  |
| --- |
| **Websites.json** |
| [  {  "id": 1,  "name": "HLTV",  "address": "hltv.org",  "visits\_per\_day": 2228,  "count\_of\_ad\_bunners": 15,  "date\_of\_creating": "2004-05"  },  {  "id": 2,  "name": "Youtube",  "address": "youtube.com",  "visits\_per\_day": 9999556,  "count\_of\_ad\_bunners": 900,  "date\_of\_creating": "1245-11"  },  {  "id": "3",  "name": "rutube",  "address": "rutube.ru",  "visits\_per\_day": "324",  "count\_of\_ad\_bunners": "324",  "date\_of\_creating": "2009-11-11"  },  {  "id": 4,  "name": "Telegram",  "address": "telegram.ru",  "visits\_per\_day": 7823,  "count\_of\_ad\_bunners": 0,  "date\_of\_creating": "2015-12"  },  {  "id": 5,  "name": "Coursera",  "address": "coursera.com",  "visits\_per\_day": 500,  "count\_of\_ad\_bunners": 0,  "date\_of\_creating": "2010-09"  },  {  "id": 6,  "name": "Twitch",  "address": "twitch.com",  "visits\_per\_day": 897443,  "count\_of\_ad\_bunners": 50,  "date\_of\_creating": "2015-03"  },  {  "id": 7,  "name": "lol",  "address": "kek.ru",  "visits\_per\_day": "123",  "count\_of\_ad\_bunners": "324",  "date\_of\_creating": "2009-11-11"  },  {  "id": 8,  "name": "lol",  "address": "lol.kek",  "visits\_per\_day": "213",  "count\_of\_ad\_bunners": "324",  "date\_of\_creating": "2009-11-11"  }  ] |

|  |
| --- |
| **users.json** |
| [  {  "id": 1,  "login": "flomaster",  "fullname": "Roma Tsay",  "role": 0,  "registeredAt": "2004-05",  "avaUrl": "https://svirtus.cdnvideo.ru/8sKJ5rxIAFxnFTA8s2qLjniN1zU=/0x0:400x400/200x200/filters:quality(100)/https://hb.bizmrg.com/esports-core-media/7e/7ef607a7eeb9785d35330c1ab8d9edfa.jpg?m=0ea7b51e20c86ee12ea6e2b7fa81cb7f",  "isEnabled": false  },  {  "id": 2,  "login": "kurant",  "fullname": "Sonya Kurant",  "role": 0,  "registeredAt": "2003-09",  "avaUrl": "https://sun9-10.userapi.com/c840024/v840024776/3dd5b/zX4qlmtP0DE.jpg?ava=1",  "isEnabled": true  },  {  "id": 3,  "login": "balizzard",  "fullname": "Roma Marchenko",  "role": 1,  "registeredAt": "2002-04",  "avaUrl": "https://icdn.lenta.ru/images/2018/01/28/18/20180128185743511/detail\_d1e245a32c7d3cf2674b6e1435f0104e.jpg",  "isEnabled": true  },  {  "id": 4,  "login": "Johs",  "fullname": "Masha Johs",  "role": 0,  "registeredAt": "2003-07",  "avaUrl": "https://pp.userapi.com/c841035/v841035758/727f5/UO7RtzR3Mzk.jpg?ava=1",  "isEnabled": false  },  {  "id": 5,  "login": "Nazarchik",  "fullname": "Egor Nazarenko",  "role": 0,  "registeredAt": "2002-07",  "avaUrl": "https://svirtus.cdnvideo.ru/ukeAGDAxBX0UZDPpKRsAvaCPMak=/0x0:277x277/200x200/filters:quality(100)/https://hb.bizmrg.com/esports-core-media/2a/2a83adf3016966390a4937d03042397d.png?m=602a7431c9222b2994f1928dcc38a87b",  "isEnabled": true  },  {  "id": 6,  "login": "Strannaya",  "fullname": "Hellen Makarenko",  "role": 0,  "registeredAt": "2004-05",  "avaUrl": "https://www.meme-arsenal.com/memes/706b6bb9472bf84db5def515da4513bf.jpg",  "isEnabled": false  }  ] |

|  |
| --- |
| **nextId.json** |
| {"id":9} |

|  |
| --- |
| **package.json** |
| {  "name": "lab1",  "version": "1.0.0",  "description": "",  "main": "app.js",  "scripts": {  "test": "echo \"Error: no test specified\" && exit 1"  },  "keywords": [],  "author": "",  "license": "ISC",  "dependencies": {  "readline-sync": "^1.4.10"  }  } |

**Приклади результатів**

|  |
| --- |
| **get/users** |
|  |

|  |
| --- |
| **get/users/4** |
|  |

|  |
| --- |
| **get/websites** |
|  |

|  |
| --- |
| **get/websites/4** |
|  |

|  |
| --- |
| **post/websites** |
|  |

|  |
| --- |
| **update/websites/9** |
|  |

|  |
| --- |
| **delete/website/9** |
|  |

**Висновки**

Виконавши дану лабораторну роботу було ознайомлено з мовою програмування JavaScript та середовищем Node.js. Було навчено створювати модулі, синхронно працювати із файловими потоками та обробляти JSON-текст.  
 Лабораторна робота була виконана на мові JavaScript за допомогою середовища виконання Node.js у редакторі VS code.