

МIНIСТЕРСТВО ОСВIТИ 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”

|  |  |  |
| --- | --- | --- |
| Виконала  студентка Ⅱ курсу  групи КП-92  Шестакова Олександра Вячеславівна  (*прізвище, ім’я, по батькові*)  варіант №23 |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Гадиняк Руслан Анатолійович  (*прізвище, ім’я, по батькові*) |

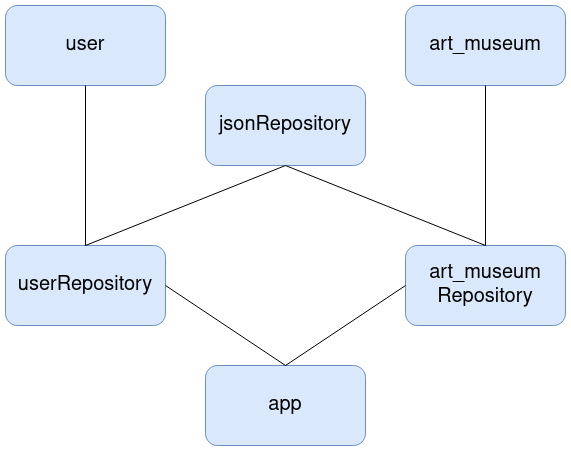
Київ 2020

**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 (музеї мистецтв).



**Лістинг програми**

|  |
| --- |
| app.js |
| **const** colors = require('colors');  **const** UserRepository = require('./repositories/userRepository');  **const** ArtMuseumRepository = require('./repositories/art\_museumRepository');  **const** JsonStorage = require('./jsonStorage');  **const** userStorage = new JsonStorage('./data/users.json');  **const** userRepository = new UserRepository('./data/users.json');  **const** artMuseumStorage = new JsonStorage('./data/art\_museums.json');  **const** artMuseumRepository = new ArtMuseumRepository('./data/art\_museums.json');  **const** readline = require('readline-sync');  **const** ArtMuseum = require('./models/art\_museum');  while (true) {  **const** text = readline.question("Enter your command:");  **const** parts = text.split("/");  **const** command = parts[0] + '/' + parts[1] + '/' + parts[2];  **const** input = parts[3];  *//exit*  *//command to exit programm*  if (text.toLowerCase() === "exit") {  console.log('exiting...'.zalgo);  return;  }  *//users/get/all*  *//command to get short info abt all usrs*  else if (command === "users/get/all") {  **let** users = userRepository.getUsers();  **let** out = 'All users:';  for (**const** user of users) {  **const** usrlog = user.login.length;  out = out + "\n\n" + user.login + " -> fullname: "  + user.fullname + "\n" + `role: ${user.role}`.padStart(usrlog + 11, "-");  }  console.log(out);  }  *//users/get/id*  *//command to get full info abt 1 user*  else if (command === "users/get/id") {  **const** userId = parseInt(input);  **const** user = userRepository.getUserById(userId);  if (!user) {  console.log(`Error: user with id ${userId} was not found.`.red);  continue;  }  console.log(`User (${user.id}):`.bold + `  login: ${user.login}  role: ${user.role}  fullname: ${user.fullname}  registered at: ${new Date(user.registeredAt)}  avatar: ${user.avaUrl}`);  }  *//artmuseums/get/all*  *//command to get short info abt all art museum*  else if (command === "artmuseums/get/all") {  **let** artMuseums = artMuseumRepository.getArtMuseums();  **let** out = 'All art museums:';  for (**const** artMuseum of artMuseums) {  out = out + "\n" + artMuseum.name + " -> " + artMuseum.country;  }  console.log(out);  }  *//artmuseums/get/id*  *//command to get full info abt 1 art museum*  else if (command === "artmuseums/get/id") {  **const** artMuseumId = parseInt(input);  **const** artMuseum = artMuseumRepository.getArtMuseumById(artMuseumId);  if (!artMuseum) {  console.log(`Error: art museum with id ${artMuseumId} was not found.`.red);  continue;  }  **const** foundedDate = new Date(artMuseum.founded);  console.log(`ArtMuseum (${artMuseum.id}):`.bold + `  name: ${artMuseum.name}  country: ${artMuseum.country}  founded: ${foundedDate.getFullYear()}  artists: ${artMuseum.artistNum}  exhibits: ${artMuseum.exhibitNum}`);  }  *//artmuseums/add*  *//command to add new art museum to file*  else if (parts[0] + '/' + parts[1] === "artmuseums/add") {  console.log("Creating new art museum:");  **let** artMuseum = new ArtMuseum();  **let** array = Object.getOwnPropertyNames(artMuseum);  **let** answers = [];  for (**let** i = 1; i <= 5; i++) {  if (i === 3) {  **let** answer = [];  while (isNaN(answer[0]) || isNaN(answer[1]) || isNaN(answer[2])) {  console.log('When founded (numbers):'.bgCyan);  answer[0] = readline.question('year:');  answer[1] = readline.question('month:');  answer[2] = readline.question('day:');  }  answers[3] = new Date(Number(answer[0]),  Number(answer[1]) - 1, Number(answer[2]) + 1).toISOString();  continue;  }  else if (i === 4 || i === 5) {  while (isNaN(answers[i])) {  answers[i] = readline.question(array[i] + ' ' + '(numbers):'.bgCyan);  }  continue;  }  answers[i] = readline.question(array[i] + ':');  }  artMuseum = new ArtMuseum(-1, answers[1],  answers[2], answers[3],  Number(answers[4]), Number(answers[5]));  if (!artMuseumRepository.addArtMuseum(artMuseum)) {  console.error(`Given art museum was not created.`.red);  }  else{  console.log(`Art museum successfully created`.green);  }  }  *//artmuseums/update/id*  *//command to update an art museum*  else if (command === "artmuseums/update/id") {  **const** artMuseumId = parseInt(input);  console.log("Updating art museum:");  **let** artMuseum = new ArtMuseum();  **let** array = Object.getOwnPropertyNames(artMuseum);  **let** answers = [];  for (**let** i = 1; i <= 5; i++) {  if (i === 3) {  **let** answer = [];  while (isNaN(answer[0]) || isNaN(answer[1]) || isNaN(answer[2])) {  console.log('When founded (numbers):'.bgCyan);  answer[0] = readline.question('year:');  answer[1] = readline.question('month:');  answer[2] = readline.question('day:');  }  answers[3] = new Date(Number(answer[0]),  Number(answer[1]) - 1, Number(answer[2]) + 1).toISOString();  continue;  }  else if (i === 4 || i === 5) {  while (isNaN(answers[i])) {  answers[i] = readline.question(array[i] + ' ' + '(numbers):'.bgCyan);  }  continue;  }  answers[i] = readline.question(array[i] + ':');  }  artMuseum = new ArtMuseum(artMuseumId, answers[1],  answers[2], answers[3],  Number(answers[4]), Number(answers[5]));  if (!artMuseumRepository.updateArtMuseum(artMuseum)) {  console.error(`Given art museum with id ${artMuseumId} was not updated.`.red);  }  else{  console.log(`Art museum successfully created`.green);  }  }  *//artmuseums/delete/id*  *//command to delete an art museum*  else if (command === "artmuseums/delete/id") {  **const** artMuseumId = parseInt(input);  if (!artMuseumRepository.deleteArtMuseum(artMuseumId)) {  console.error(`Given art museum with id ${artMuseumId} was not deleted.`.red);  }  else {  console.log(`Deleted art museum with id ${artMuseumId}.`.blue)  }  }  *//if command is incorrect*  else {  console.error(`This command is not supported: '${text}'.`.yellow);  }  } |

|  |
| --- |
| 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.items;  }  **get** nextId() {  **const** jsonText = fs.readFileSync(this.filePath.slice(0, -6)+'\_id.json');  **const** Id = JSON.parse(jsonText);  return Id.nextId;  }  incrementNextId() {  **let** newId = this.nextId;  newId +=1;  fs.writeFileSync(this.filePath.slice(0, -6)+'\_id.json', JSON.stringify({nextId : newId}, null, 4));  }  writeItems(items) {  fs.writeFileSync(this.filePath, JSON.stringify(items, null, 4));  }  };  module.exports = JsonStorage; |

|  |
| --- |
| 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; *//integer*  this.registeredAt = registeredAt; *//date*  this.avaUrl = avaUrl; *//string*  this.isEnabled = isEnabled; *//boolean*  }  };  module.exports = User; |

|  |
| --- |
| art\_museum.js |
| **class** ArtMuseum {  **constructor**(id, name, country, founded, artistNum, exhibitNum) {  this.id = id; *// number*  this.name = name; *// string*  this.country = country; *// string*  this.founded = founded; *// date*  this.artistNum = artistNum; *// integer*  this.exhibitNum = exhibitNum; *// integer*  }  };  module.exports = ArtMuseum; |

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

|  |
| --- |
| art\_museumRepository.js |
| **const** ArtMuseum = require('../models/art\_museum');  **const** JsonStorage = require('../jsonStorage');  **class** ArtMuseumRepository {  **constructor**(filePath) {  this.storage = new JsonStorage(filePath);  }  getArtMuseums() {  **const** items = this.storage.readItems();  **let** artMuseums = [];  for (**const** item of Object.values(items)) {  artMuseums.push(new ArtMuseum(  item.id, item.name, item.country,  item.founded, item.artistNum,  item.exhibitNum));  }  return artMuseums;  }  getArtMuseumById(artMuseumid) {  **const** artMuseums = this.getArtMuseums();  for (**const** artMuseum of artMuseums) {  if (artMuseum.id === artMuseumid) {  return new ArtMuseum(artMuseum.id,  artMuseum.name, artMuseum.country,  artMuseum.founded, artMuseum.artistNum,  artMuseum.exhibitNum);  }  }  return null;  }  addArtMuseum(artMuseumModel) {  **const** artMuseums = this.getArtMuseums();  artMuseumModel.id = this.storage.nextId;  this.storage.incrementNextId();  artMuseums.push(artMuseumModel);  this.storage.writeItems({ items: artMuseums });  return this.storage.nextId - 1;  }  updateArtMuseum(artMuseumModel) {  **const** artMuseums = this.getArtMuseums();  for (**let** i in artMuseums) {  if (artMuseums[i].id === artMuseumModel.id) {  artMuseums[i] = artMuseumModel;  this.storage.writeItems({ items: artMuseums });  return artMuseumModel.id;  }  }  return null;  }  deleteArtMuseum(artMuseumId) {  **const** artMuseums = this.getArtMuseums();  for (**let** i in artMuseums) {  if (artMuseums[i].id === artMuseumId) {  artMuseums.splice(i, 1);  this.storage.writeItems({ items: artMuseums });  return artMuseumId;  }  }  return null;  }  };  module.exports = ArtMuseumRepository; |

|  |
| --- |
| art\_museums.json |
| {  "items": [  {  "id": 0,  "name": "Grand Palais",  "country": "France",  "founded": "1900-05-01T21:57:56.000Z",  "artistNum": 112,  "exhibitNum": 12  },  {  "id": 1,  "name": "Somerset House",  "country": "United Kingdom",  "founded": "1776-03-24T21:57:56.000Z",  "artistNum": 132,  "exhibitNum": 28  },  {  "id": 4,  "name": "some",  "country": "some",  "founded": "1234-12-20T21:57:56.000Z",  "artistNum": 234,  "exhibitNum": 123  },  {  "id": 8,  "name": "n",  "country": "c",  "founded": "1898-04-03T21:57:56.000Z",  "artistNum": 12,  "exhibitNum": 23  },  {  "id": 9,  "name": "nam",  "country": "co",  "founded": "1857-06-07T21:57:56.000Z",  "artistNum": 12,  "exhibitNum": 34  },  {  "id": 10,  "name": "nam",  "country": "co",  "founded": "1857-05-08T21:57:56.000Z",  "artistNum": 12,  "exhibitNum": 34  }  ]  } |

|  |
| --- |
| art\_museum\_id.json |
| {  "nextId": 13  } |

|  |
| --- |
| users.json |
| {  "items": [  {  "id": 0,  "login": "someUser11",  "role": 1,  "fullname": "I Am User",  "registeredAt": "2018-09-19T10:05:19Z",  "avaUrl": "http://progbase.herokuapp.com/images/progbase.jpg",  "isEnabled": false  }  ,  {  "id": 1,  "login": "someOTHERUser11",  "role": 0,  "fullname": "I Am A User",  "registeredAt": "2018-09-19T10:05:19Z",  "avaUrl": "http://progbase.herokuapp.com/images/progbase.jpg",  "isEnabled": false  }  ]  } |

|  |
| --- |
| user\_id.json |
| {  "nextId": 2  } |

**Приклади результатів**

alexa@back-twig:~/2nd course/all progbases/webprogbase/labs/lab1$ node app.js

Enter your command:users/get/all

All users:

someUser11 -> fullname: I Am User

--------------role: 1

someOTHERUser11 -> fullname: I Am A User

-------------------role: 0

Enter your command:users/get/id/1

User (1):

login: someOTHERUser11

role: 0

fullname: I Am A User

registered at: Wed Sep 19 2018 13:05:19 GMT+0300 (Eastern European Summer Time)

avatar: http://progbase.herokuapp.com/images/progbase.jpg

Enter your command:artmuseums/get/all

All art museums:

Grand Palais -> France

Somerset House -> United Kingdom

some -> some

n -> c

nam -> co

nam -> co

Enter your command:artmuseums/get/id/10

ArtMuseum (10):

name: nam

country: co

founded: 1857

artists: 12

exhibits: 34

Enter your command:artmuseums/add

Creating new art museum:

name:museum

country:country

When founded (numbers):

year:1234

month:12

day:12

artistNum (numbers):1234

exhibitNum (numbers):123

Art museum successfully created

Enter your command:artmuseums/update/id/10

Updating art museum:

name:name

country:c

When founded (numbers):

year:123

month:12

day:12

artistNum (numbers):44

exhibitNum (numbers):44

Art museum successfully created

Enter your command:artmuseums/delete/id/10

Deleted art museum with id 10.

Enter your command:exit

exiting...

**Висновки**

Виконавши дану лабораторну роботу, ми познайомились із мовою програмування JavaScript та створили консольний додаток у середовищі Node.js, який дозволяє користувачу взаємодіяти з двома сутностями: користувачами системи та музеями мистецтв. При створенні програми ми навчилися синхронно працювати із файловими потоками, а також читати та обробляти JSON-текст.