

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 веб програмування”

тема “Основи JavaScript”

|  |  |  |
| --- | --- | --- |
| Виконав  студент IІ курсу  групи КП-93  Деркач Станіслав Дмитрович  (*прізвище, ім’я, по батькові*)  варіант №22 |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Гадиняк Руслан Анатолійович  (*прізвище, ім’я, по батькові*) |

Київ 2020

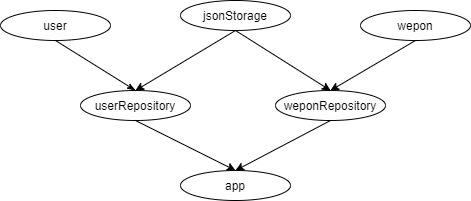
**Мета роботи**

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 (“Організації”).

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



**Тексти коду програм**

|  |
| --- |
| **app.js** |
| const readline = require('readline-sync');  const UserRepository = require('./repositories/userRepository');  const WeponRepository = require('./repositories/weponRepository');  const userStorage = new UserRepository('./data/users.json');  const weponStorage = new WeponRepository('./data/wepons.json');  function printUser(user){      const id = "  id: "+user.id;      const login = "  login: '"+user.login+"'";      const fullname = "  fullname: '"+user.fullname+"'";      const role = "  role: "+user.role;      const reg = "  geristeregAt: "+user.registeredAt;      const ava = "  avaUrl: '"+user.avaUrl+"'";      let isE;      if (user.isEnabled)          isE = "  isEnabled: "+"enabled";      else          isE = "  isEnabled: "+"disabled";      console.log(id+'\n'+login+'\n'+fullname+'\n'+role+'\n'+reg+'\n'+ava+'\n'+isE);  }  function printWepon(wepon){      const id = "  id: "+wepon.id;      const name = "  name: '"+wepon.name+"'";      const author = "  author: '"+wepon.author+"'";      const damage = "  damage: "+wepon.damage;      const speed = "  speed: "+wepon.speed;      const createdAt = "  Created at: "+wepon.createdAt;      console.log(id+'\n'+name+'\n'+author+'\n'+damage+'\n'+speed+'\n'+createdAt);  }  while (true) {      const input = readline.question("Enter command: ");      if (input === "stop") break;      if (input === "get/users") {          const items = userStorage.getUsers();          for (const item of items)              console.log("  "+item.id+". "+item.login);      } else      if (input.startsWith("get/users/")){          const str\_id=input.slice(10);          const id = Number(str\_id);          const user = userStorage.getUserById(id);          if (user===null) {              console.log("Incorrect ID");              continue;          }          printUser(user);      } else      if (input.startsWith("get/wepons/")){          const str\_id=input.slice(11);          if (str\_id.length<=0) continue;          const id = Number(str\_id);          const wepon = weponStorage.getWeponById(id);          printWepon(wepon);      } else      if (input.startsWith("delete/wepons/")){          const str\_id=input.slice(14);          const id = Number(str\_id);          if (isNaN(id)) {              console.log("Incorrect ID");              continue;          }          const f = weponStorage.deleteWepon(id);          if (!f) {              console.log("Incorrect ID");          }else              console.log("  Deleted");      } else      if (input.startsWith("update/wepons/")){          const str\_id=input.slice(14);          const id = Number(str\_id);          if (isNaN(id)) {              console.log("Incorrect ID");              continue;          }          let wepon = weponStorage.getWeponById(id);          if (wepon===null) {              console.log("Incorrect ID");              continue;          }          const name = readline.question("  name: ");          const author = readline.question("  author: ");          const str\_damage = readline.question("  damage: ");          const str\_speed = readline.question("  speed: ");          if (name!=="") wepon.name=name;          if (author!=="") wepon.author=author;          const damage = Number(str\_damage);          const speed = Number(str\_speed);          if (!isNaN(damage) && str\_damage!=="") wepon.damage=Number(damage);          if (!isNaN(speed) && str\_speed!=="") wepon.speed=Number(speed);          weponStorage.updateWepon(wepon);          console.log("  Updated");      } else      if (input === "get/wepons") {          const items = weponStorage.getWepons();          for (const item of items)              console.log("  "+item.id+". "+item.name);      } else      if (input === "post/wepons") {          const name = readline.question("  name: ");          const author = readline.question("  author: ");          const damage = Number(readline.question("  damage: "));          const speed = Number(readline.question("  speed: "));          const date = new Date();          let id;          if (name.length>0 && author.length>0 &&              !isNaN(damage) && damage>=0 && !isNaN(speed) && speed>0){              id = weponStorage.addWepon(name, author, damage, speed, date);              console.log("Added with id "+id);          }          else console.log("  Incorrect Data");      }    } |

|  |
| --- |
| **weponRepository.js** |
| const Wepon = require('../models/wepon');  const JsonStorage = require('../jsonStorage');    class UserRepository {        constructor(filePath) {          this.storage = new JsonStorage(filePath);      }        getWepons() {          const wepons = this.storage.readItems();          return wepons;      }      addWepon(name, author, damage, speed, createdAt){          const id=this.storage.nextId;          this.storage.incrementNextId();          const wepon= new Wepon(id ,name, author, damage, speed, createdAt);          let items = this.storage.readItems();          items.push(wepon);          this.storage.writeItems(items);          return id;      }        getWeponById(id) {          const items = this.storage.readItems();          for (const item of items) {              if (item.id === id) {                  return new Wepon(item.id, item.name,   item.author, item.damage, item.speed, item.createdAt);              }          }          return null;      }      updateWepon(wepon) {          let items = this.storage.readItems();          for (const i in items) {              if (items[i].id === wepon.id) {                  items[i]=wepon;              }          }          this.storage.writeItems(items);      }      deleteWepon(id){          const items = this.storage.readItems();          const newItems = items.filter(item=> item.id!==id);          if (items.length===newItems.length) return false;            this.storage.writeItems(newItems);          return true;      }    };    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 users = this.storage.readItems();          return users;      }        getUserById(id) {          const items = this.storage.readItems();          for (const 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; |

|  |
| --- |
| **wepon.js** |
| class Wepon{      constructor(id, name, author, damage, speed, createdAt){          [this.id, this.name, this.author, this.damage, this.speed, this.createdAt]          = [id, name, author, damage, speed, createdAt];      }  }  module.exports=Wepon; |

|  |
| --- |
| **user.js** |
| class User{      constructor (id, login, fullname, role, registeredAt, avaUrl, isEnabled){          this.id=id;          this.login=login;          this.fullname=fullname;          this.role=role;          this.registeredAt=registeredAt;          this.avaUrl=avaUrl;          this.isEnabled=isEnabled;      }  }  module.exports=User; |

|  |
| --- |
| **jsonStorage.js** |
| const fs=require('fs');  class JsonStorage {      // filePath - path to JSON file      constructor(filePath) {          this.filePath=filePath;      }      get nextId() {          const jsonText = fs.readFileSync(this.filePath);          const jsonObj = JSON.parse(jsonText);          return jsonObj.nextId;      }      incrementNextId() {          const jsonText = fs.readFileSync(this.filePath);          let jsonObj = JSON.parse(jsonText);          jsonObj.nextId++;          const newJsonText = JSON.stringify(jsonObj, null, 2);          fs.writeFileSync(this.filePath, newJsonText);      }      readItems() {          const jsonText = fs.readFileSync(this.filePath);          const jsonObj = JSON.parse(jsonText);          return jsonObj.items;      }      writeItems(items) {          const jsonText = fs.readFileSync(this.filePath);          let jsonObj = JSON.parse(jsonText);          jsonObj.items=items;          const newJsonText = JSON.stringify(jsonObj, null, 2);          fs.writeFileSync(this.filePath, newJsonText);      }  };  module.exports = JsonStorage; |

**Вміст .json файлів**

*wepons.json*

|  |
| --- |
| {    "nextId": 5,    "items": [      {        "id": 1,        "name": "knifes",        "author": "Me",        "damage": 20,        "speed": 90,        "createdAt": "2020-10-11T11:54:40.608Z"      },      {        "id": 3,        "name": "Gun",        "author": "No name",        "damage": 121,        "speed": 85,        "createdAt": "2020-10-11T12:18:32.071Z"      },      {        "id": 4,        "name": "sword",        "author": "HEllBo7",        "damage": 130,        "speed": 50,        "createdAt": "2020-10-11T12:24:19.322Z"      }    ]  } |

*users.json*

|  |
| --- |
| {    "nextId": 3,    "items": [      {        "id": 1,        "login": "admin",        "fullname": "Admin Admin",        "role": 1,        "registeredAt":"2020-08-31T13:28:36.599Z",        "avaUrl":"some url adress",        "isEnabled": true      },      {        "id": 2,        "login": "testUser",        "fullname": "Test User",        "registeredAt":"2020-08-31T13:28:36.599Z",        "role": 0,        "avaUrl":"https://www.ava.jpeg",        "isEnabled": false      }    ]  } |

**Приклади результатів**

1) get/users

|  |
| --- |
| Enter command: get/users  1. admin  2. testUser |

2) get/users/1

|  |
| --- |
| Enter command: get/users/1  id: 1  login: 'admin'  fullname: 'Admin Admin'  role: 1  geristeregAt: 2020-08-31T13:28:36.599Z  avaUrl: 'some url adress'  isEnabled: enabled |

3) post/wepons

|  |
| --- |
| Enter command: post/wepons  name: sword  author: HEllBo7  damage: 130  speed: 50  Added with id 4 |

4) get/wepons

|  |
| --- |
| Enter command: get/wepons  1. knife  2. bazuka  3. Gun  4. sword |

5) get/wepons/4

|  |
| --- |
| Enter command: get/wepons/4  id: 4  name: 'sword'  author: 'undefined'  damage: 130  speed: 50  Created at: 2020-10-11T12:24:19.322Z |

7) update/wepons/3

|  |
| --- |
| Enter command: update/wepons/3  name:  author: No name  damage:  speed: 85  Updated |

8) delete/wepons/2

|  |
| --- |
| Enter command: delete/wepons/2  Deleted |

**Висновки**

Виконавши лабораторну роботу, я познайомився із мовою програмування JavaScript та середовищем Node.js, навчився створювати модулі, синхронно працювати із файловими потоками та обробляти JSON-текст.