

ЛАБОРАТОРНА РОБОТА № 1

Робота з файлами, JSON, об'єктами, модулями, npm, lodash, nodemon, yargs

Хід роботи:

Task 1. Hello, World

- В виконавчому файлі `sandbox/task01.js` виведіть на консоль "Hello, World!".

Лістинг програми:

```
console.log("Hello, World!");
```

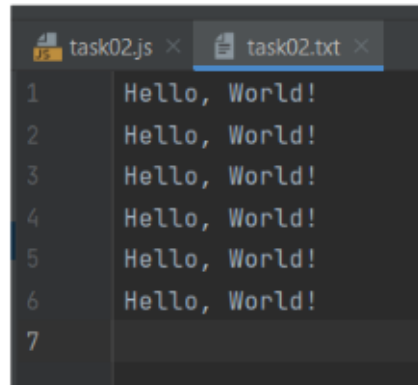
Результат запуску програми:

```
PS F:\documents\univers\NodeJS\Lab1> node task1.js
Hello, World!
PS F:\documents\univers\NodeJS\Lab1> █
```

					ДУ «Житомирська політехніка».21.121.4.000 – Лр1						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Волківський Я.С.			Звіт з лабораторної роботи			Літ.	Арк.	Аркушів	
Перевір.		Лисенко М.С.								1	13
Керівник								ФІКТ Гр. ІПЗ-21-5[1]			
Н. контр.											
Зав. каф.											

Task 2. Створити додаток, що приєднує до файлу рядок

- Щоразу при запуску скрипта в текстовий файл повинен додаватись новий рядок



Лістинг програми:

```
const fs = require('fs');
const readline = require('readline');
const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout,
  prompt: 'Створити відсутній файл? (Y/N) - '
});
const dir_f = './data/task2.txt';
const dir = 'data';

if (fs.existsSync(dir_f)){
  rl.close();
  fs.appendFile(dir_f, " Hello World!\n", () => {
    console.log(`Дані записані в файл!\n Дані, котрі щойно записані і були записані раніше:\n${fs.readFileSync(dir_f, 'utf-8')}`);
  });
} else {
  console.log(`\n      !WARNING ERROR DETECTED!\n\n      Error to Read/Write File\n Directive ${dir_f} not found\n`);
  rl.prompt();
  rl.on('line', (input) => {
    input = input.toUpperCase();
    if (input == "Y"){
      if (!fs.existsSync(dir)) fs.mkdirSync(dir);
    }
  });
}
```

```

        if (!fs.writeFileSync(dir_f, " Hello World!\n")) console.log(`File has
        been created\n\n Дані записані в файл!\n Дані, котрі щойно записані і були записані
        раніше:\n${
            fs.readFileSync(dir_f, 'utf-8')
        }`);
        else console.log("Error to Create File");
    }
    rl.close();
});
}

```

Результат виконання програми:

```

PS F:\documents\univers\NodeJS\Lab1> node task2.js
Дані записані в файл!
Дані, котрі щойно записані і були записані раніше:
Hello World!
Hello World!
Hello World!
Hello World!

PS F:\documents\univers\NodeJS\Lab1>

```

```

data > task2.txt
3 | Hello World!
4 | Hello World!
5 |

```

Task 3. Вивести інформацію про користувача операційної системи

- Вивести привітання у форматі "Hello,{userName}!", де userName – користувач Вашої операційної системи. Вивід здійснити в окремий файл або на консоль.

Лістинг програми:

```
const os = require('os');
const fs = require('fs');
const readline = require('readline');
const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout,
  prompt: '    Записати також в файл? (Y/N) - '
});
const path = './data/task3.txt';
const dir = 'data';

console.log(`\n    Hello, ${os.userInfo().username}\n`);

rl.prompt();
rl.on('line', (input) => {
  input = input.toUpperCase();
  if(input == "Y"){
    if(!fs.existsSync(dir)) fs.mkdir(dir);
    if(!fs.appendFileSync(path, `    Hello, ${os.userInfo().username}\n`))
console.log("    Успішно записано в файл\n\n");
    else console.log("    Помикла запису в файл\n\n");
  }
  rl.close();
});
```

Результат запусненої програми:

```
Hello, Yaroslav
```

```
Записати також в файл? (Y/N) -
```

```
PS F:\documents\univers\NodeJS\Lab1> node task3.js
```

```
Hello, Yaroslav
```

```
Записати також в файл? (Y/N) - Y
```

```
Успішно записано в файл
```

```
PS F:\documents\univers\NodeJS\Lab1>
```

```
data > task3.txt
```

```
1 Hello, Yaroslav
```

```
2 Hello, Yaroslav
```

```
3 Hello, Yaroslav
```

```
4 Hello, Yaroslav
```

```
5
```

Волківський Я.С.

Лисенко М.С.

Змн.

Арк.

№ докум.

Підпис

Дата

ДУ «Житомирська політехніка».21.121.4.000 – Лр1

Арк.

5

Task 4. Використання модуля lodash

- Підключіть модуль lodash в скрипт index.js
- Протестуйте 5 методів модуля lodash з відповідними коментарями

Лістинг програми:

```
const lodash = require('lodash')

const shuf = lodash.shuffle([1, 2, 3, 4, 5, 6, 7, 8, 9, 10]);
console.log(`First operation:`);
console.log(shuf)
console.log('\n\n');

const items = [
  {name: 'First Item', price: 260},
  {name: 'Second Item', price: 120},
  {name: 'Third Item', price: 600},
  {name: 'Fourth Item', price: 60},
  {name: 'Fiveth Item', price: 1000}
]

const item_res = lodash.sortBy(items, ['price', 'name']);
console.log(`Second operation:`);
console.log(item_res)
console.log('\n\n');

const array = [1, 2, 3, 4, 5, 6];
const array_operation = lodash.chunk(array, 2);
console.log(`Third operation:`);
console.log(array_operation);
console.log('\n\n');

const array_take_operation = lodash.take([1, 2, 3, 4, 5, 6], 3);
console.log(`Fourth operation:`);
console.log(array_take_operation);
console.log('\n\n');

const array_intersection = lodash.intersection([1, 2, 3, 4],[2, 3, 4, 5, 6, 7]);
console.log(`Fiveth operation:`);
console.log(array_intersection);
console.log('\n`');
```

Результат виконання програми:

```
PS F:\documents\univers\NodeJS\Lab1> node task4.js
First operation:
[
  1, 9, 4, 3, 7,
  10, 8, 6, 2, 5
]

Second operation:
[
  { name: 'Fourth Item', price: 60 },
  { name: 'Second Item', price: 120 },
  { name: 'First Item', price: 260 },
  { name: 'Third Item', price: 600 },
  { name: 'Fiveth Item', price: 1000 }
]

Third operation:
[ [ 1, 2 ], [ 3, 4 ], [ 5, 6 ] ]

Fourth operation:
[ 1, 2, 3 ]

Fiveth operation:
[ 2, 3, 4 ]
```

Волківський Я.С.

Лисенко М.С.

Змн.

Арк.

№ докум.

Підпис

Дата

ДУ «Житомирська політехніка».21.121.4.000 – Лр1

Арк.

7

ProjectTask. Створіть додаток для додавання, перегляду, видалення мов користувача

Запуск скрипта *app.js* повинен супроводжуватись командами для виконання операцій з мовами:

```
node app.js add --title="JavaScript" --level="Junior"
node app.js remove --title="JavaScript"
node app.js list
node app.js read --title="JavaScript"
```

Методи для виконання операцій з мовами винести в окремий модуль *user.js*. Методи повинні містити необхідну валідацію

Команди можна обробити з використанням модуля *yargs*. Для всіх команд потрібно здійснити валідацію.

Лістинг програми:

```
const fs = require('fs');
const yargs = require('yargs');

const loadUser = () => {
  try{return JSON.parse(fs.readFileSync('user.json').toString());}
  catch (e){return {firstName: '', lastName: '', languages: []}}
}

const loadDes = () => {
  try{return JSON.parse(fs.readFileSync('package.json').toString());}
  catch(e){return {name: '', version: '', description: '', main: '', scripts: [],
keywords: [], author: '', license: '', dependencies: []}};
}

const saveUser = (user) => {fs.writeFileSync('user.json', JSON.stringify(user));}

const addLanguage = (title, level) => {
  const user = loadUser();
  const isDuplicate = user.languages.find((language) => language.title === title);

  if (!title || !level){
    console.log('\n Помилка! Потрібно ввести Мову та Рівень\n');
    return;
  }

  if (isDuplicate){
    console.log('\n Ця мова вже наявна!\n');
    return;
  }
}
```



```

    user.languages.push({title, level});
    saveUser(user);

    console.log(`\n Мова ${title} з рівнем ${level} успішно додані!\n`);
}

const removeLanguage = (title) => {
    const user = loadUser();
    const languageIndex = user.languages.findIndex((language) => language.title === title);

    if (languageIndex === -1){
        console.log('\n Помилка! Мова не знайдена.\n');
        return;
    }

    user.languages.splice(languageIndex, 1);
    saveUser(user);
    console.log(`\n Мова ${title} була успішно видалена!\n`);
}

const listLanguages = () => {
    const user = loadUser();
    console.log(`\n Користувач: ${user.firstName} ${user.lastName}`);

    if (user.languages.length === 0){
        console.log('\n Мова не додана\n');
        return;
    }

    console.log('\n Мови:');
    user.languages.forEach((language) => {console.log(`      - ${language.title} (${language.level})`)});
    console.log('\n');
}

const aboutProg = () => {
    const info = loadDes();

    console.log(`\n Назва: ${info.name}\n Версія: ${info.version}\n Опис: ${info.description}\n Головний файл (main): ${info.main}\n Скрипти:`);
    for(let key in info.scripts) console.log(`      - ${info.scripts[key]}`);
    console.log(`\n Автор проекту: ${info.author}\n Ліцензія: ${info.license}\n Залежності:`);
    for(let key in info.dependencies) console.log(`      - ${key}: ${info.dependencies[key]}`);
    console.log('\n');
}

```

```

const readLanguage = (title) => {
  const user = loadUser();
  const language = user.languages.find((language) => language.title === title);

  if (!language){
    console.log('\n Мова не знайдена\n');
    return;
  }

  console.log(`\n Мова: ${language.title}\n Рівень: ${language.level}\n`);
}

module.exports = {
  addLanguage,
  removeLanguage,
  listLanguages,
  readLanguage,
  aboutProg
}

yargs.command({
  command: 'add',
  describe: 'Add a new language',
  builder: {
    title: {
      describe: 'Title of the language',
      demandOption: true,
      type: 'string'
    },
    level: {
      describe: 'Level of proficiency',
      demandOption: true,
      type: 'string'
    }
  },
  handler(argv){addLanguage(argv.title, argv.level)}
});

yargs.command({
  command: 'info',
  describe: 'describe about program',
  handler(){aboutProg()}
});

yargs.command({
  command: 'remove',
  describe: 'Remove a language',
  builder: {
    title: {
      describe: 'Title of the language',

```

```

        demandOption: true,
        type: 'string'
    }
  },
  handler(argv){removeLanguage(argv.title, argv.level);}
});

yargs.command({
  command: 'list',
  describe: 'List all languages',
  handler(){listLanguages()}
})

yargs.command({
  command: 'read',
  describe: 'Read a language',
  builder:{
    title:{
      describe: 'Title of the language',
      demandOption: true,
      type: 'string'
    }
  },
  handler(argv){readLanguage(argv.title)}
})

yargs.parse();

```

Лістинг user.json:

```

{"firstName":"Yaroslav","lastName":"Volkivskyi","languages":[{"title":"JS","level":"Pro"}, {"title":"NodeJS","level":"Beginner"}]}

```

Результат виконаної програми:

```

F:\documents\univers\NodeJS\Lab1>node app.js add --title="JavaScript" --level="Junior"

Мова JavaScript з рівнем Junior успішно додані!

F:\documents\univers\NodeJS\Lab1>
F:\documents\univers\NodeJS\Lab1>node app.js remove --title="JavaScript"

Мова JavaScript була успішно видалена!

F:\documents\univers\NodeJS\Lab1>

```

```
F:\documents\univers\NodeJS\Lab1>node app.js list
```

```
Користувач: Yaroslav Volkivskyi
```

```
Мови:
```

- JS (Pro)
- NodeJS (Beginner)

```
F:\documents\univers\NodeJS\Lab1>node app.js read --title="JavaScript"
```

```
Мова не знайдена
```

```
F:\documents\univers\NodeJS\Lab1>
```

```
F:\documents\univers\NodeJS\Lab1>node app.js read --title="JavaScript"
```

```
Мова: JavaScript
```

```
Рівень: Junior
```

```
F:\documents\univers\NodeJS\Lab1>
```

		Волківський Я.С.			ДУ «Житомирська політехніка».21.121.4.000 – Лр1	Арк.
		Лисенко М.С.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновок: навчився працювати з файлами, JSON, об'єктами, модулями, прт, lodash, nodemon, yargs

		Волківський Я.С.			ДУ «Житомирська політехніка».21.121.4.000 – Лр1	Арк.
		Лисенко М.С.				13
Змн.	Арк.	№ докум.	Підпис	Дата		