

ЛАБОРАТОРНА РОБОТА № 1

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

Мета: *освоїти нові об'єкти, модулі і інші файли для роботи з програмним кодом*

Хід роботи:

Завдання 1:

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

Лістинг програми:

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

Результат виконання програми:

```
D:\Install\Node_js\node.exe D:\nodejs\lab01\sandbox\task01.js  
Hello, World!
```

Рис. 1. Результат виконання програми

Завдання 2:

Task 2. Створити додаток, що приєднує до файлу рядок

Лістинг програми:

```
text = "Hello, World!\n";  
const fs = require('node:fs');  
fs.appendFile("task02.txt", text, (err) => {  
  if (err) throw err;  
  console.log('Text appended to file');  
});
```

					ДУ «Житомирська політехніка».22.121.04.000 - Лр1		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Пасічник Б.Р			Звіт з лабораторної роботи	Літ.	Арк.
Перевір.		Лисенко М.С					1
Керівник						ФІКТ Гр. ВТ-22-1[2]	
Н. контр.							
Зав. каф.							

Результат виконання програми:

```
D:\Install\Node_js\node.exe D:\nodejs\lab01\task02\task02.js

Text appended to file

Process finished with exit code 0
JS task02.js task02.txt x
1 Hello, World!
2 Hello, World!
3 Hello, World!
```

Рис. 2-3. Результат виконання програми

Завдання 3:

Лістинг програми:

```
const os = require('os');
const text = os.userInfo().username;
console.log(`Hello, ${text}`);
```

Результат виконання програми:

```
D:\Install\Node_js\node.exe D:\nodejs\lab01\task03\task03.js
Hello, Bohdan P

Process finished with exit code 0
```

Рис. 4 – Результат виконання програми

Завдання 4:

Ініціалізація проекту, package.json

- В проекті даної лабораторної роботи здійсніть ініціалізацію пакетного менеджера: `npm init`
- Виконайте кроки по заповненню інформації про додаток
- Переконайтесь в наявності файлу `package.json`

Результат виконання програми:

```
D:\nodejs\lab01>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (lab01)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository: github.com/jolyjell/NODE_JS.git
keywords:
author: Bohdan Pasichnyk
license: (ISC)
About to write to D:\nodejs\lab01\package.json:
{
  "name": "lab01",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo `Error: no test specified` && exit 1"
  },
  "repository": {
    "type": "git",
    "url": "github.com/jolyjell/NODE_JS.git"
  },
  "author": "Bohdan Pasichnyk",
  "license": "ISC",
  "description": ""
}
```

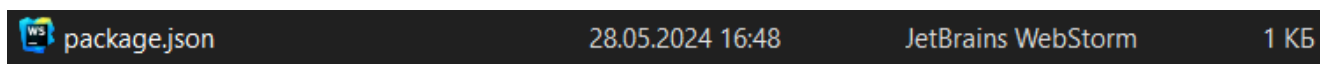


Рис. 5-6 – Результат виконання програми

Завдання 4:

Task 4. Використання модуля lodash

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

Лістинг програми:

```
// Підключення модуля lodash
const _ = require('lodash');

// 1. _.chunk - розбиває масив на менші масиви заданого розміру
const array = ['a', 'b', 'c', 'd'];
const chunkedArray = _.chunk(array, 2);
console.log('_.chunk:', chunkedArray);

// 2. _.random - генерує випадкове число в заданому діапазоні
const randomNum = _.random(1, 100);
console.log('_.random:', randomNum);

// 3. _.capitalize - перетворює першу букву рядка в велику
const string = 'hello world';
const capitalizedString = _.capitalize(string);
console.log('_.capitalize:', capitalizedString);
```

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.04.000 - Лр1	Арк.
		Лисенко М.С.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```
// 4. _.merge - глибоке злиття двох об'єктів
const object1 = { 'a': 1, 'b': { 'c': 2 } };
const object2 = { 'b': { 'd': 3 }, 'e': 4 };
const mergedObject = _.merge(object1, object2);
console.log('_.merge:', mergedObject);

// 5. _.uniq - створює новий масив з унікальними значеннями
const arrayWithDuplicates = [1, 2, 1, 4, 1, 3];
const uniqueArray = _.uniq(arrayWithDuplicates);
console.log('_.unique:', uniqueArray);
```

Результат виконання програми:

```
D:\Install\Node_js\node.exe D:\nodejs\lab01\task04\index.js
_.chunk: [ [ 'a', 'b' ], [ 'c', 'd' ] ]
_.random: 99
_.capitalize: Hello world
_.merge: { a: 1, b: { c: 2, d: 3 }, e: 4 }
_.uniq: [ 1, 2, 4, 3 ]

Process finished with exit code 0
```

Рис. 7 – Результат виконання програми

		Пасічник Б.Р			ДУ «Житомирська політехніка».22.121.04.000 - Лр1	Арк.
		Лисенко М.С.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 5:

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:

user.js

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

const userPath = path.join(__dirname, 'user.json');

function readUserFile() {
  const data = fs.readFileSync(userPath);
  return JSON.parse(data);
}

function writeUserFile(data) {
  fs.writeFileSync(userPath, JSON.stringify(data, null, 2));
}

function addLanguage(title, level) {
  const user = readUserFile();
  const duplicateLanguage = user.languages.find(language => language.title === title);
  if (duplicateLanguage) {
    console.log('Language already exists!');
    return;
  }
  user.languages.push({ title, level });
  writeUserFile(user);
  console.log('Language added successfully!');
}

function removeLanguage(title) {
  const user = readUserFile();
  const languages = user.languages.filter(language => language.title !== title);
  if (languages.length === user.languages.length) {
    console.log('Language not found!');
    return;
  }
  user.languages = languages;
  writeUserFile(user);
  console.log('Language removed successfully!');
}

function listLanguages() {
```

		Пасічник Б.Р			ДУ «Житомирська політехніка».22.121.04.000 - Лр1	Арк.
		Лисенко М.С.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

const user = readUserFile();
console.log('Languages:');
user.languages.forEach(language => {
  console.log(`- ${language.title} (${language.level})`);
});
}

function readLanguage(title) {
  const user = readUserFile();
  const language = user.languages.find(language => language.title === title);
  if (!language) {
    console.log('Language not found!');
    return;
  }
  console.log(`Title: ${language.title}, Level: ${language.level}`);
}

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

```

app.js

```

const yargs = require('yargs');
const { hideBin } = require('yargs/helpers');
const { addLanguage, removeLanguage, listLanguages, readLanguage } =
require('./user');

yargs(hideBin(process.argv))
  .command({
    command: 'add',
    describe: 'Add a new language',
    builder: {
      title: {
        describe: 'Language title',
        demandOption: true,
        type: 'string'
      },
      level: {
        describe: 'Language proficiency level',
        demandOption: true,
        type: 'string'
      }
    },
    handler(argv) {
      addLanguage(argv.title, argv.level);
    }
  })
  .command({
    command: 'remove',
    describe: 'Remove a language',
    builder: {
      title: {
        describe: 'Language title',
        demandOption: true,
        type: 'string'
      }
    },
    handler(argv) {
      removeLanguage(argv.title);
    }
  })

```

		Пасічник Б.Р			ДУ «Житомирська політехніка».22.121.04.000 - Лр1	Арк.
		Лисенко М.С.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    })
    .command({
      command: 'list',
      describe: 'List all languages',
      handler() {
        listLanguages();
      }
    })
    .command({
      command: 'read',
      describe: 'Read a language',
      builder: {
        title: {
          describe: 'Language title',
          demandOption: true,
          type: 'string'
        }
      },
      handler(argv) {
        readLanguage(argv.title);
      }
    })
    .demandCommand(1, 'You need at least one command before moving on')
    .help()
    .argv;

```

Результат виконання програми:

```

D:\nodejs\lab01>node app.js list
Languages:
- NodeJS (Beginner)
- C# (Junior)
- English (Pro)

D:\nodejs\lab01>node app.js remove --title="JavaScript"
Language not found!

D:\nodejs\lab01>node app.js add --title="JavaScript" --level="Junior"
Language added successfully!

D:\nodejs\lab01>node app.js remove --title="JavaScript"
Language removed successfully!

D:\nodejs\lab01>node app.js read --title="JavaScript"
Language not found!

D:\nodejs\lab01>

```

Рис. 8 – Результат виконання програми

Висновки: в ході виконання лабораторної роботи я отримав перші практичні для роботи з Node.js

		Пасічник Б.Р			ДУ «Житомирська політехніка».22.121.04.000 - Лр1	Арк.
		Лисенко М.С.				7
Змн.	Арк.	№ докум.	Підпис	Дата		