

## Лабораторна робота № 1

### Хід роботи:

**Task1.Hello,World:**

**task01.js:**

```
console.log('Hello, world');
```

**Task 2. Створити додаток, що приєднує до файлу рядок**

**task02.js:**

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

const filePath = 'task02.txt';
const newLine = 'Hello world';
fs.appendFile(filePath, newLine + '\n', (err) => {
  if (err) {
    console.error('Помилка під час додавання рядка до файлу:', err);
    return;
  }
});
```

**Task 3. Вивести інформацію про користувача операційної системи**

**task03.js:**

```
const os = require('os');

const userName = os.userInfo().username;
const greeting = `Hello, ${userName}!`;
console.log(greeting);
```

					ДУ «Житомирська політехніка».24.121.13.000 – Лр1							
Змн.	Арк.	№ докум.	Підпис	Дата	Звіт з лабораторної роботи				Літ.		Арк.	Аркушів
Розроб.		Короцінський В.В.									1	6
Перевір.		Праздніков. В. О.										
Керівник												
Н. контр.												
Зав. каф.												
					ФІКТ Гр. ВТ-22-1[1]							

## Task 4. Використання модуля lodash

Task04.js:

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

// 1. Метод `_.find` шукає перший елемент у масиві, який відповідає заданому предикату.
const users = [{ name: 'John', age: 30 }, { name: 'Jane', age: 25 }, { name: 'Peter', age: 40 }];
const adultUser = _.find(users, user => user.age >= 30);
console.log('1. Перший дорослий користувач:', adultUser);
// 2. Метод `_.map` перетворює кожен елемент масиву за допомогою заданої callback-функції.
const numbers = [1, 2, 3, 4, 5];
const squaredNumbers = _.map(numbers, num => num * num);
console.log('2. Квадрати чисел:', squaredNumbers);
// 3. Метод `_.filter` створює новий масив, що містить лише ті елементи з вихідного масиву, які відповідають заданому предикату.
const activeUsers = _.filter(users, user => user.active);
console.log('3. Активні користувачі:', activeUsers);
// 4. Метод `_.reduce` зводить масив до одного значення, застосовуючи callback-функцію до кожної пари послідовних елементів.
const sum = _.reduce(numbers, (total, num) => total + num, 0);
console.log('4. Сума чисел:', sum); // 15
// 5. Метод `_.sortBy` сортує масив за допомогою заданої callback-функції порівняння.
const sortedUsers = _.sortBy(users, user => user.age);
console.log('5. Відсортовані користувачі за віком:', sortedUsers);
```

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

app.js:

```
const yargs = require('yargs');
const userModel = require('./user');

const ERROR_CODES = {
  LANGUAGE_EXISTS: 'LANGUAGE_EXISTS',
```

		Короцінський В.В.			ДУ «Житомирська політехніка».24.121.13.000 – Лр1	Арк.
		Праздніков В. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

```

INVALID_LEVEL: 'INVALID_LEVEL',
LANGUAGE_NOT_FOUND: 'LANGUAGE_NOT_FOUND',
};
yargs.default('command', () => {
  userModule.loadUserData();
});
yargs
  .command('add', 'Додати нову мову до вашого профілю', {
    title: {
      type: 'string',
      description: 'Назва мови',
      demand: true,
    },
    level: {
      type: 'string',
      description: 'Рівень володіння мовою (напр., Початківець, Середній, Досконалий)',
      demand: true,
    },
  }, ({ title, level }) => {
    try {
      userModule.addLanguage(title, level);
    } catch (err) {
      if (err.code === ERROR_CODES.LANGUAGE_EXISTS) {
        console.error('Мова з такою назвою вже існує.');
```

```

      } else if (err.code === ERROR_CODES.INVALID_LEVEL) {
        console.error('Неправильний рівень володіння мовою.');
```

```

      } else {
        console.error('Непередбачувана помилка.');
```

```

      }
      throw err;
    }
  })
  .command('remove', 'Видалити мову з вашого профілю', {
    title: {
      type: 'string',
      description: 'Назва мови, яку потрібно видалити',
      demand: true,
    },
  }, ({ title }) => {

```

		Короцінський В.В.			ДУ «Житомирська політехніка».24.121.13.000 – Лр1	Арк.
		Праздніков В. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		3

```

try {
  userModule.removeLanguage(title);
} catch (err) {
  if (err.code === ERROR_CODES.LANGUAGE_NOT_FOUND) {
    console.error('Мова з такою назвою не знайдена.');
```

```

  } else {
    console.error('Непередбачувана помилка.');
```

```

  }
  throw err;
}
})
.command('list', 'Перелік всіх мов у вашому профілі', {}, ({}) => {
  userModule.listLanguages();
})
.command('read', 'Переглянути інформацію про мову', {
  title: {
    type: 'string',
    description: 'Назва мови, про яку потрібно отримати інформацію',
    demand: true,
  },
  ({ title }) => {
    try {
      userModule.readLanguage(title);
    } catch (err) {
      if (err.code === ERROR_CODES.LANGUAGE_NOT_FOUND) {
        console.error('Мова з такою назвою не знайдена.');
```

```

      } else {
        console.error('Непередбачувана помилка.');
```

```

      }
      throw err;
    }
  })
.version()
.help()
.alias('h', 'help')
.argv;

```

		Короцінський В.В.			ДУ «Житомирська політехніка».24.121.13.000 – Лр1	Арк.
		Праздніков В. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

**user.js:**

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

const DATA_FILE = path.join(__dirname, 'user.json');
let userData = {};
function loadUserData() {
  try {
    const data = fs.readFileSync(DATA_FILE, 'utf8');
    if (data) {
      userData = JSON.parse(data);
    }
  } catch (err) {
    console.error('Error loading user data:', err);
  }
}

function saveUserData() {
  try {
    fs.writeFileSync(DATA_FILE, JSON.stringify(userData), 'utf8');
  } catch (err) {
    console.error('Помилка збереження даних користувача:', err);
  }
}

function addLanguage(title, level) {
  if (!title || !level) {
    throw new Error('Необхідні поля: назва та рівень');
  }
  const languages = userData.Languages || [];
  const existingLanguage = languages.find(language => language.title === title);
  if (existingLanguage) {
    throw new Error(`Мова з назвою "${title}" вже існує`);
  }
  languages.push({ title, level });
  userData.Languages = languages;
  saveUserData();
  console.log(`Мова "${title}" успішно додана`);
}

function removeLanguage(title) {
```

		Короцінський В.В.			ДУ «Житомирська політехніка».24.121.13.000 – Лр1	Арк.
		Праздніков В. О.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

if (!title) {
throw new Error('Необхідне поле: назва');
}
const languages = userData.Languages || [];
const languageIndex = languages.findIndex(language => language.title
=== title);
if (languageIndex === -1) {
throw new Error(`Мова з назвою "${title}" не знайдена`);
}
languages.splice(languageIndex, 1);
userData.Languages = languages;
saveUserData();
console.log(`Мова "${title}" успішно видалена`);
}
function listLanguages() {
const languages = userData.Languages || [];
if (languages.length === 0) {
console.log('Мов не знайдено');
} else {
console.table(languages);
}
}
function readLanguage(title) {
if (!title) {
throw new Error('Необхідне поле: назва');
}
const languages = userData.Languages || [];
const language = languages.find(language => language.title ===
title);
if (!language) {
console.error(`Мова з назвою "${title}" не знайдена`);
return;
}
console.log(`Інформація про мову "${language.title}":`);
console.table(language);
}
module.exports = {
loadUserData,
addLanguage,
saveUserData,

```

		Короцінський В.В.			ДУ «Житомирська політехніка».24.121.13.000 – Лр1	Арк.
		Праздніков В. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

```
removeLanguage,  
listLanguages,  
readLanguage,  
};
```

**user.json:**

```
{  
  "firstName": "Vic",  
  "lastName": "tor",  
  "languages": [  
    {  
      "title": "JS",  
      "Level": "Pro"  
    },  
    {  
      "title": "NodeJS",  
      "Level": "Pro"  
    }  
  ],  
  "Languages": [  
    {  
      "title": "French",  
      "level": "Beginner"  
    },  
    {  
      "title": "JavaScript",  
      "level": "Junior"  
    }  
  ]  
}
```

Висновок: під час виконання лабораторної роботи, працювати з файлами, JSON, об'єктами, модулями, npm, lodash, nodemon

, yargs.

		Короцінський В.В.			ДУ «Житомирська політехніка».24.121.13.000 – Лр1	Арк.
		Праздніков В. О.				7
Змн.	Арк.	№ докум.	Підпис	Дата		