**Лабораторна робота №11**

**Тема: Прототипи, конструктори**

***Мета роботи:*** навчитися працювати з прототипами та методами роботи з ними

**Завдання на лабораторну роботу**

**Завдання 0:** Ретельно вивчити теорію за посиланнями:

Регулярні вирази:

<https://uk.javascript.info/prototype-inheritance#tasks>

**Завдання 1:**

**[Робота з прототипами](https://uk.javascript.info/prototype-inheritance" \l "robota-z-prototipami)**

Ось код, у якому створюють пару об’єктів і потім в ході виконання їх модифікують.

Які значення будуть показані в результаті виконання коду?

let animal = {

jumps: null

};

let rabbit = {

\_\_proto\_\_: animal,

jumps: true

};

alert( rabbit.jumps ); // ? (1)

delete rabbit.jumps;

alert( rabbit.jumps ); // ? (2)

delete animal.jumps;

alert( rabbit.jumps ); // ? (3)

Повинно бути 3 відповіді.

**[Алгоритм пошуку](https://uk.javascript.info/prototype-inheritance" \l "algoritm-poshuku)**

важливість: 5

Задача має дві частини.

Ми маємо ось такі об’єкти:

let head = {

glasses: 1

};

let table = {

pen: 3

};

let bed = {

sheet: 1,

pillow: 2

};

let pockets = {

money: 2000

};

1. Використайте властивість \_\_proto\_\_ визначивши прототипи таким чином, щоб отримання властивостей було можливим по ось такому шляху: pockets → bed → table → head. Для прикладу, pockets.pen повинно отримати значення 3 (було знайдено в table), а bed.glasses отримує значення 1 (було знайдено в head).
2. Дайте відповідь: для отримання властивості glasses що буде швидше: визначити її так pockets.glasses чи так head.glasses? При необхідності зробіть порівняльний тест.

**[Де буде записано?](https://uk.javascript.info/prototype-inheritance" \l "de-bude-zapisano)**

важливість: 5

Ми маємо об’єкт rabbit, котрий успадковує властивості від об’єкта animal.

Якщо ми викличемо rabbit.eat(), у який з об’єктів буде записана властивість full: в animal чи rabbit?

let animal = {

eat() {

this.full = true;

}

};

let rabbit = {

\_\_proto\_\_: animal

};

rabbit.eat();

**[Чому обидва хом’ячка наситились?](https://uk.javascript.info/prototype-inheritance" \l "chomu-obidva-khomyachka-nasitilis)**

важливість: 5

Ми маємо два хом’ячка (об’єкти): speedy та lazy, які успадковують властивості від загального об’єкта hamster.

Коли ми годуємо одного з них, інший також стає ситим. Але чому? Як ми можемо це виправити?

let hamster = {

stomach: [],

eat(food) {

this.stomach.push(food);

}

};

let speedy = {

\_\_proto\_\_: hamster

};

let lazy = {

\_\_proto\_\_: hamster

};

// Цей хом’ячок знайшов їжу

speedy.eat("apple");

alert( speedy.stomach ); // apple

// Але цей також має їжу, чому? Виправте це.

alert( lazy.stomach ); // apple

*Закомітити виконані завдання. Надати доступ до репозиторію викладачам*