

4. Прототипи.

Контрольні запитання

- 04.01. Що таке прототип об'єкта?
- 04.02. Яким чином здійснюється доступ до прототипу з конкретного об'єкта? Як це зробити з функції-конструктора?
- 04.03. Що таке ланцюжок прототипів?
- 04.04. Як працює оператор **instanceof**?
- 04.05. Як здійснити наслідування в прототипно-орієнтованій моделі?

Завдання для аудиторної роботи

- [A]04.01. Мутуйте вбудований «клас» **Object** шляхом додавання у **Object.prototype** методу **describe()**, який повертає рядок

```
"This object has keys: <перелік ключів>"
```

Створіть декілька різних об'єктів та викличте метод **describe()**.

- [A]04.02. Використовуючи прототипно-орієнтовану модель (без синтаксису **class**), опишіть такі конструктори:

- **Animal(name)** для створення об'єктів з методом **speak()**, який виводить **"\${name} видає звук!"**.
- **Dog(name, breed)** з перевизначенням методом **speak()**, який виводить **"\${name} породи \${breed} гавкає!"**.

Всі методи мають знаходитися в прототипі.

Продемонструйте роботу програми, створивши декілька об'єктів заданих конструкторів.

- [A]04.03. Використовуючи прототипно-орієнтовану модель (без синтаксису **class**), опишіть «супер-клас» **Flower**. Визначіть ієрархію квітів. Створіть кілька об'єктів-квіток. Зберіть букет **Bouquet** з визначенням його вартості. Знайдіть квітки в букеті, що відповідають заданому діапазону довжини стебля. Опишіть метод **toString()** для того щоб виводити квіти в букеті. Вартість квітки має залежати від її типу та довжини стебля.

Завдання для самостійної роботи

- [B]04.01. Мутуйте вбудований «клас» **Object** шляхом додавання у **Object.prototype** методу **size()**, який повертає кількість власних полів об'єкта (тобто тих, які містяться виключно в об'єкті і відсутні в прототипі). Власні функції не

враховувати.

Створіть декілька різних об'єктів та викличте метод `size()`.

[B]04.02. Використовуючи прототипно-орієнтовану модель (без синтаксису `class`), опишіть такі конструктори:

- `Shape()` для створення об'єктів з методом `area()`, який повертає площу фігури.
- `Ellipse(a, b)` з методом `area()`, який повертає площу еліпса.
- `Circle(r)` з методом `area()`, який повертає площу круга.

Всі методи мають знаходитися в прототипі.

Продемонструйте роботу програми створивши декілька об'єктів заданих конструкторів.

[B]04.03. Використовуючи прототипно-орієнтовану модель (без синтаксису `class`), опишіть «супер-клас» `Candy`. Визначіть ієрархію цукерок. Створіть кілька об'єктів-цукерок. Зберіть дитячий подарунок `Present` з урахуванням його ваги. Знайдіть цукерки в подарунку, що відповідають заданому діапазону вмісту цукру. Опишіть метод `toString()` для того щоб виводити цукерки в подарунку. Вміст цукру у цукерці безпосередньо залежить від її об'єму та типу.