**Державний вищий навчальний заклад**

**Ужгородський національний університет**

**Факультет інформаційних технологій**

**ЛАБОРАТОРНА РОБОТА №9**

**Тема:** “JavaScript Ч2”

Виконав студент І курсу

спеціальності «Інженерія

програмного забезпечення»

Левринц Дмитрій Сергійович

**Ужгород-2024**

**Завдання:**

**1.** Виконати домашні завдання описані у теоретичних матеріалах до лекцій 4:

1. За допомогою цикла реалізувати код, який виводить наступну картину

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

2. За допомогою циклу while / do-while реалізувати timer на 10 секунд

Зверніть увагу на властивості обєкту/Конструктора Date (не використовуючи JS timer - setTimeOut, setInterval)

7: 1. Використовуючи літеральну нотацію створити обєкт car з властивістю speedometer = 0;

2. Використовуючи методи Object додати до обєкту car наступні методи:

setSpeedometer, що оновлює дані speedometer (сеттер)

getSpeedometer, що повертає вміст speedometer (геттер)

clearSpeedometr, що очищує вміст speedometer

3. Модифікувати код таким чином, щоб можна було зробити наступне: car.setSpeed(200).setSpeed(300).getSpeed().clearSpeed() // Ланцюжковий виклик

**2.** 8. 1. створити щонайменше 3 функції конструктори (класи), Можна застосовувати, навіть бажано, ES6 синтаксис

2. проілюструвати на прикладі трьох і більше класах прототипне наслідування в JS

**3.** Як ви знаєте, викликавши метод toString на будь-якому обєкті, наприклад ({}).toString() отримаємо "[object Object]". ⚠️Увага⚠️, чорна скринька в студію:

- опишіть як ми можемо інтерпретувати (пояснити) значення вищезгаданого результату "[object Object]"

- чому [].toString() не виводить "[object Array]"

- реалізувати вивід "[object Array]" для масиву

**Хід роботи**

**Лекція №4:**

**1.** За допомогою цикла реалізував код, який виводить наступну картину:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

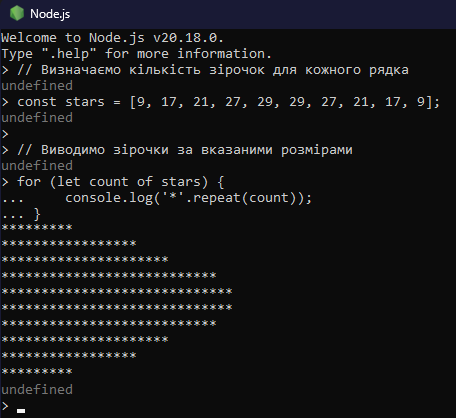
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

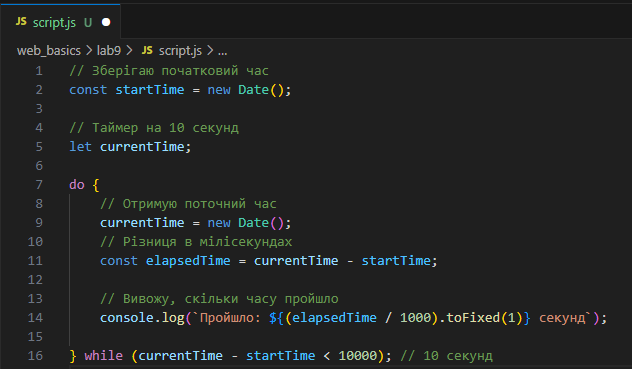
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*



**2.** За допомогою циклу while / do-while реалізував timer на 10 секунд

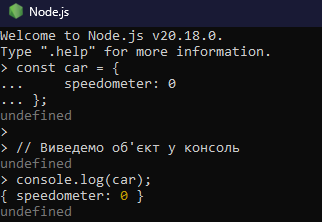
Звернув увагу на властивості обєкту/Конструктора Date (не використовуючи JS timer - setTimeOut, setInterval):



Написав код в VS Code тому що в Node.js код спрацьовує і не даже можливості заскрінити…

**Лекція №7:**

**1.** Використав літеральну нотацію створити обєкт car з властивістю speedometer = 0; :

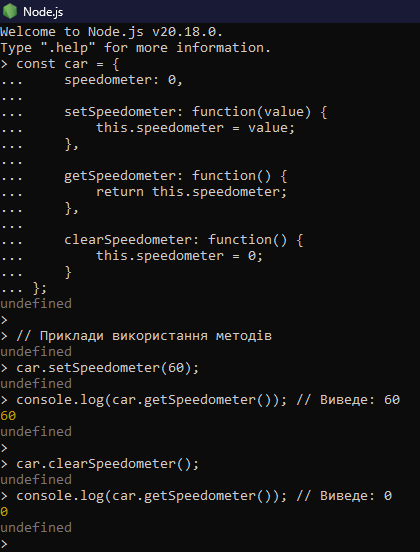


**2.** Використав методи Object додав до обєкту car наступні методи:

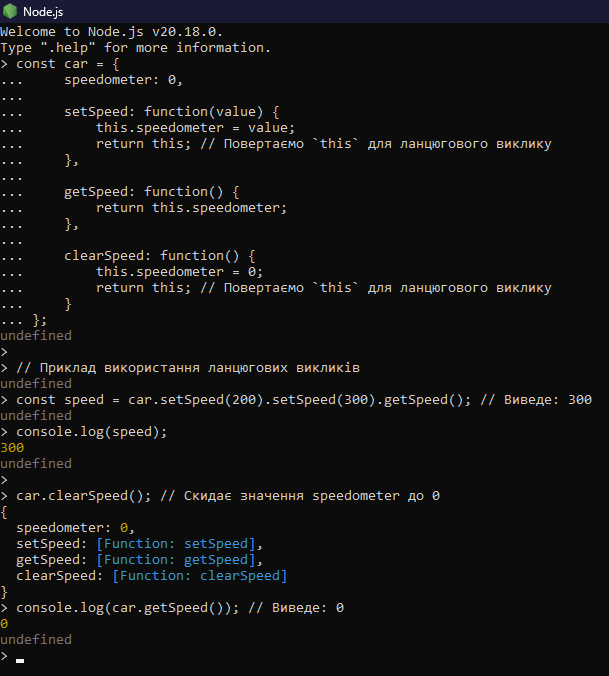
setSpeedometer, що оновлює дані speedometer (сеттер)

getSpeedometer, що повертає вміст speedometer (геттер)

clearSpeedometr, що очищує вміст speedometer



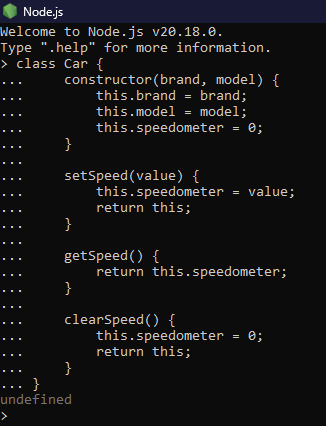
**3.** Модифікував код таким чином, щоб можна було зробити наступне: car.setSpeed(200).setSpeed(300).getSpeed().clearSpeed() // Ланцюжковий виклик:



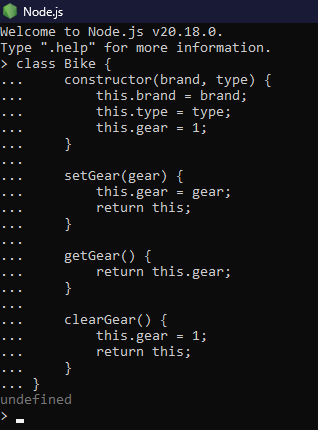
**Лекція №8:**

**1.** Створив щонайменше 3 функції конструктори (класи), можна застосовувати, навіть бажано, ES6 синтаксис:

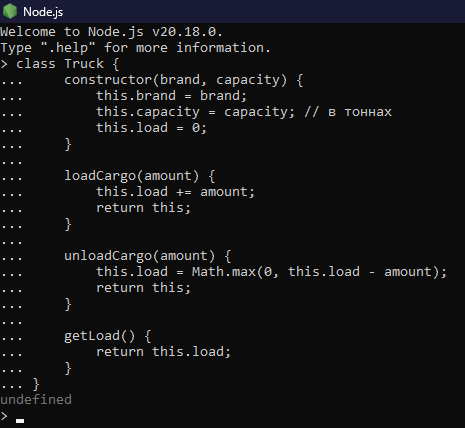
Клас car:



Клас bike:



Клас truck:

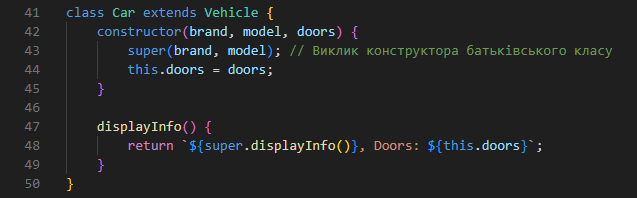


Використання:

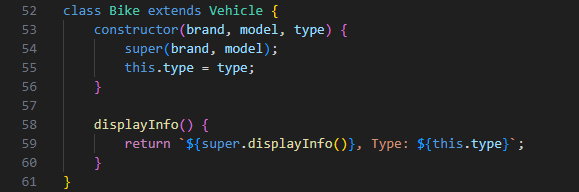


**2.** Проілюстрував на прикладі трьох і більше класах прототипне наслідування в JS:

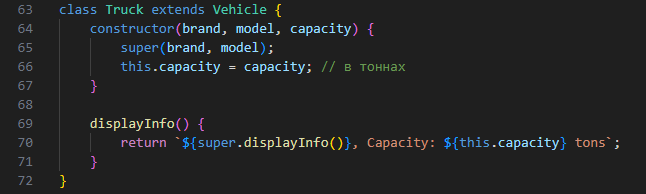
Клас car:



Клас bike:



Клас truck:

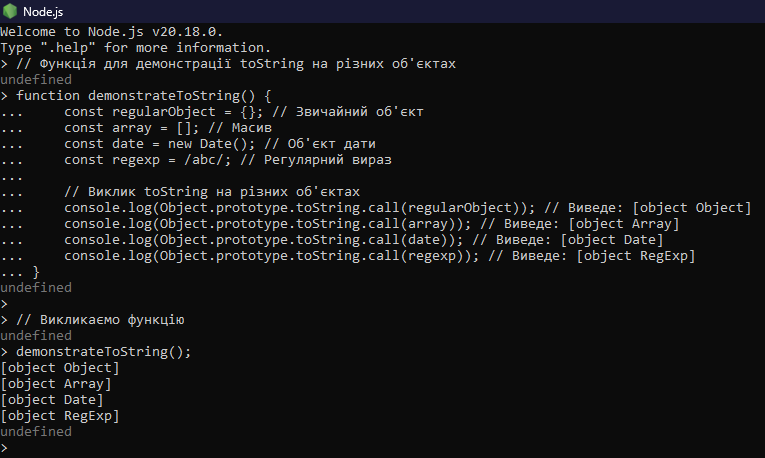


**Завдання №3:**

- Описав як я можу інтерпретувати значення вищезгаданого результату "[object Object]":

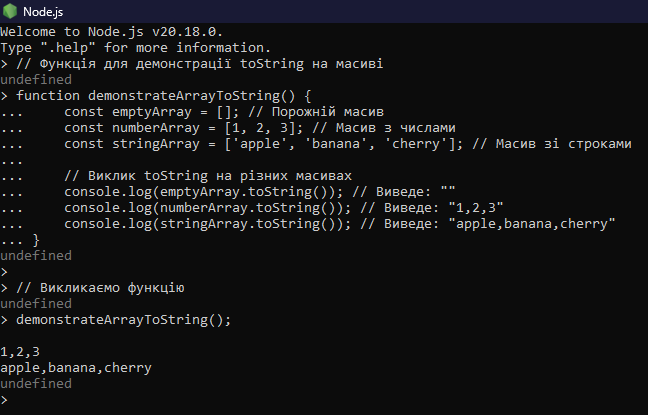
Коли я викликаю метод toString() на об'єкті, JavaScript використовує метод Object.prototype.toString. Цей метод повертає рядок у форматі "[object Type]", де Type є типом об'єкта.

**"[object Object]"** означає, що я викликав toString() на звичайному об'єкті. У цьому випадку "Object" вказує на те, що це стандартний об'єкт JavaScript.

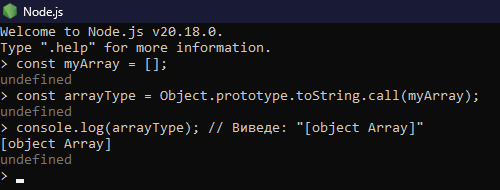


- Чому [].toString() не виводить "[object Array]":

Коли я викликаю toString() на масиві, JavaScript не використовує Object.prototype.toString, а замість цього повертає рядок, який представляє елементи масиву, об'єднані комами. Тобто, якщо масив порожній, то [].toString() просто повертає пустий рядок "".



- реалізував вивід "[object Array]" для масиву:



**Висновок**

У цій лабораторній роботі я дослідив кілька ключових аспектів роботи з методами toString() в JavaScript, зокрема, як вони поводяться для різних типів об'єктів.

**1. Метод toString()**:

Викликаючи toString() на звичайному об'єкті, ми отримуємо рядок у форматі "[object Object]". Це свідчить про те, що метод використовує Object.prototype.toString, що повертає тип об'єкта.

Для інших типів об'єктів, таких як масиви, дати або регулярні вирази, метод Object.prototype.toString повертає відповідні типи, наприклад, "[object Array]", "[object Date]" та "[object RegExp]".

**2. Поведінка масивів**:

Коли toString() викликається на масиві, JavaScript не використовує Object.prototype.toString, а повертає рядок, що представляє елементи масиву, об'єднані комами. Якщо масив порожній, результатом буде пустий рядок "".

**3. Застосування**:

Розуміння роботи методу toString() є важливим для правильного оброблення даних в JavaScript, оскільки це дозволяє правильно ідентифікувати типи об'єктів та коректно відображати їх значення.

Ця робота підкреслює гнучкість та особливості JavaScript у роботі з різними типами даних, що є критично важливим для розробників, які прагнуть ефективно управляти даними в своїх програмах.