In your IDE of choice, create a new JavaScript file named shopping.js and make it so that all code written in the file follows **strict mode**.

Create an anonymous immediately-invoking function that will hold the main execution of all program calls. Writing a simple command console.log(“Hi”) inside this function and running the code should output “Hi“ in the console.

Create constructor functions with properties representing the following “things”:

* **Product** - product id (random number of 5 digits generated on every product creation), name, price (with 2 decimal places), expiration date
* **ShoppingBag** - a list of products (initially empty; function constructor does not have any input parameters)

Add **getInfo** method to **Product**. It should return a formatted string containing product code (the first and last letter of the name together with the product id), name and price.

"Banana" -> BA32784, Banana, 129.31

Add **addProduct** method to **ShoppingBag**. It should receive a **Product** and add it to the product list. You can add a product to the list only if it has a valid expiration date.

Add a method to **ShoppingBag** that calculates the average product price of products in your product list and prints out this value with the precision of three decimals.

Add **getMostExpensive** method that finds the most expensive product and prints out its info.

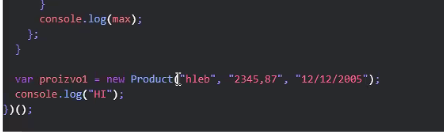
Add **calculateTotalPrice** method to **ShoppingBag** that calculates the total price of products in the shopping bag list.

Create a constructor function with properties representing the following:

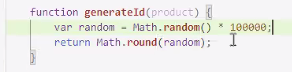
* **PaymentCard** - account balance (number with 2 decimal places), active or inactive status, valid until date

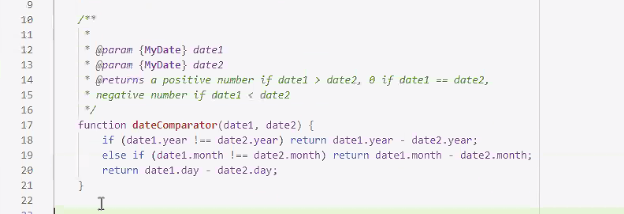
Create **checkoutAndBuy** function which receives shopping bag and payment card and prints if the purchase is successful or not. Purchase is successful only if the amount on the card is greater or equal to the total price of products in the shopping bag. If there is not enough money, print out the amount that is missing to complete the purchase.

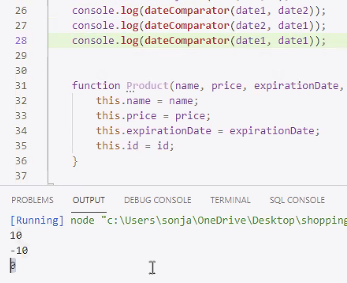




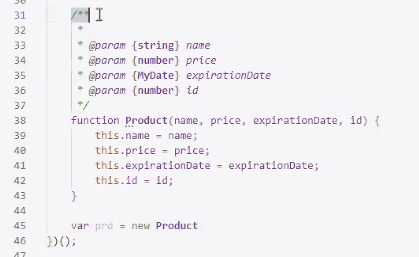


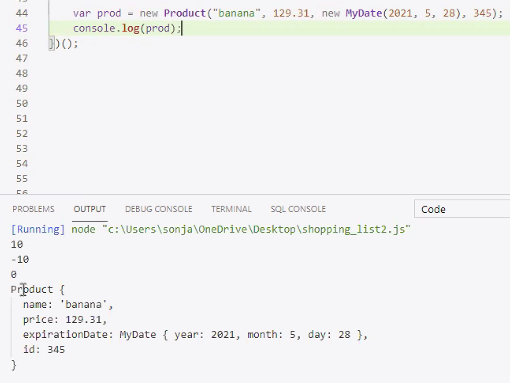
provera console log:



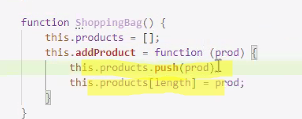
Olaksava \*\* za citanje koda I da bi npr new Product izbacio parametre…. :



Provera:



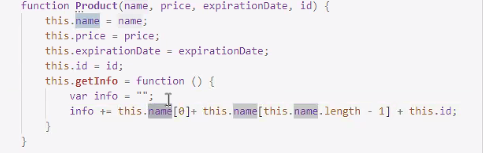
Svaki prd je nova instance za objekat

ili ili

 u prazan niz dodavace nove prde.

Ako zelimo da baratamo sa parametrima iz Producta OBAVEZNO stavljamo this u okviru funkcije getInfo:

KAda je funkcija u objektu(Product) to je metod (getInfo)!



Kao I this products za metod addProduct:



Provera isteka roka: vraca true ako je prd istekao, tj exp >today

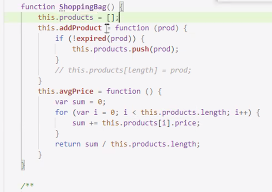
 🡺false

 🡺true

 proverava da li je dodao u bag product kome je istekao rok 🡺 nije!

HOP NA FUNKCIJU , stani na link tj funkciju CTRL+KLIK LEVI!!!

Kada dodamo prd u korpu povecava se niz 1 + 1 + 1…



Length govori Koliko prda ima u korpi: 🡺 pd.length =1 + 1 + 1:

