3-noyabr amaliyot

1. Name Classes

   a1 = new Name("john", "SMITH")
   a1.fname ➞ "John"

   a1.lname ➞ "Smith"

   a1.fullname ➞ "John Smith"

   a1.initials ➞ "J.S"

   https://edabit.com/challenge/qNMtrtizgssAQqP2b

2. **Smoothie** sinfini yarating va quyidagilarni bajaring:

    -   **ingredients** xususiyatini yarating,
    -   **Smoothie** tayyorlash uchun ishlatiladigan **ingredients** ning umumiy narxini hisoblaydigan **getCost** methodini yarating.
    -   **getPrice** methodini yarating, **getPrice = 2.5 * getCost.**
        10^{-2} yaxlitlash.
    -   **ingredients** oladigan va ularni alifbo tartibida chiroyli tavsiflovchi jumlaga joylashtiradigan **getName** usulini yarating. Agar bir nechta **ingredients** bo'lsa, oxiriga "**Fusion**" so'zini qo'shing, aks holda "**Smoothie**" qo'shing. "**-berries**" ni "**-berry**" ga o'zgartirishni unutmang. Quyidagi misollarga qarang.

Ingredient	Price
Strawberries	$1.50
Banana	$0.50
Mango	$2.50
Blueberries	$1.00
Raspberries	$1.00
Apple	$1.75
Pineapple	$3.50

s1 = new Smoothie(["Banana"])

s1.ingredients → ["Banana"]

s1.getCost() → "$0.50"

s1.getPrice() → "$1.25"

s1.getName() → "Banana Smoothie"

s2 = Smoothie(["Raspberries", "Strawberries", "Blueberries"])

s2.ingredients → ["Raspberries", "Strawberries", "Blueberries"]

s2.getCost() → "$3.50"

s2.getPrice() → "$8.75"

s2.getName() → "Blueberry Raspberry Strawberry Fusion"

https://edabit.com/challenge/7ytmgtAXGAccwpvJN

4-noyabr amaliyot

1. Qaysi bob siz turgan sahifaga yaqinroq ekanligini qaytaradigan funksiya yarating. Ikki bob bir xil masofada joylashgan bo'lsa, yuqoriroq sahifa raqami bilan bobni qaytaring.

   ```
   nearestChapter({
     "Chapter 1" : 1,
     "Chapter 2" : 15,
     "Chapter 3" : 37
   }, 10) // "Chapter 2"
   ```

   ```
   nearestChapter({
     "New Beginnings" : 1,
     "Strange Developments" : 62,
     "The End?" : 194,
     "The True Ending" : 460
   }, 200) // "The End?"
   ```

   ```
   nearestChapter({
     "Chapter 1a" : 1,
     "Chapter 1b" : 5
   }, 3) // "Chapter 1b"
   ```

   Kiritilgan object valuelari aniq integer tipiga tegishli raqamlar.

2.

```
let name= "*** Deep Into Coffee ***"
let menu= [
    { item: 'orange juice', type: 'drink', price: 2.13 },
    { item: 'lemonade', type: 'drink', price: 0.85 },
    { item: 'cranberry juice', type: 'drink', price: 3.36 },
    { item: 'pineapple juice', type: 'drink', price: 1.89 },
    { item: 'lemon iced tea', type: 'drink', price: 1.28 },
    { item: 'apple iced tea', type: 'drink', price: 1.28 },
    { item: 'vanilla chai latte', type: 'drink', price: 2.48 },
    { item: 'hot chocolate', type: 'drink', price: 0.99 },
    { item: 'iced coffee', type: 'drink', price: 1.12 },
    { item: 'tuna sandwich', type: 'food', price: 0.95 },
    { item: 'ham and cheese sandwich', type: 'food', price: 1.35 },
    { item: 'bacon and egg', type: 'food', price: 1.15 },
    { item: 'steak', type: 'food', price: 3.28 },
    { item: 'hamburger', type: 'food', price: 1.05 },
    { item: 'cinnamon roll', type: 'food', price: 1.05 }
]
let orders= [];


let tcs = new CoffeeShop(name, menu, orders);

/*
addOrder: Menuda shu nomdagi mahsulot bo'lsa Orders ga qo'shsin.
Aks holda, return "This item is currently unavailable!"
fulfillOrder: if the orders array is not empty, return "The {item} is
ready!". If the orders array is empty, return "All orders have been
fulfilled!"
listOrders: returns the list of orders taken, otherwise, an empty array.
dueAmount: returns the total amount due for the orders taken.
cheapestItem: returns the name of the cheapest item on the menu.
drinksOnly: returns only the item names of type drink from the menu.
foodOnly: returns only the item names of type food from the menu.
*/



tcs.addOrder("hot cocoa") // "This item is currently unavailable!"
```

```
tcs.addOrder("iced tea") // "This item is currently unavailable!"

tcs.addOrder("cinnamon roll") //  "Order added!"
tcs.addOrder("iced coffee") // "Order added!"
tcs.listOrders // ["cinnamon roll", "iced coffee"]

tcs.dueAmount() // 2.17

tcs.fulfillOrder() // "The cinnamon roll is ready!"
tcs.fulfillOrder() // "The iced coffee is ready!"
tcs.fulfillOrder() // "All orders have been fulfilled!"
// all orders have been presumably served

tcs.listOrders() // []
// an empty array is returned if all orders have been exhausted

tcs.dueAmount() // 0.0
// no new orders taken, expect a zero payable

tcs.cheapestItem() // "lemonade"
tcs.drinksOnly() // ["orange juice", "lemonade", "cranberry juice",
"pineapple juice", "lemon iced tea", "vanilla chai latte", "hot chocolate",
"iced coffee"]
tcs.foodOnly() // ["tuna sandwich", "ham and cheese sandwich", "bacon
and egg", "steak", "hamburger", "cinnamon roll"]
```

2.