**Code:**

import java.util.Scanner;

class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

String[] itemName = {

"Silenced .22 Pistol",

"10mm Pistol",

"Lever-action Shotgun",

"Assault Carbine",

"Minigun",

"Gehenna (GRA)",

"Esther (GRA)",

"Frag Grenade",

"Plasma Grenade"

};

int[] itemPrice = {80, 750, 2000, 3950, 5000, 12000, 18000, 150, 300};

int[] playerInv = {1, 1, 0, 1, 0, 0, 0, 3, 0};

int[] botInv = {8, 8 , 4 , 4 , 2 , 1 , 1, 10, 5};

int purchaseChoice;

char dialogueChoice;

char barterChoice;

int sellChoice;

int userCaps = 15500;

System.out.println("Welcome, Sir or Madam. Do you wish to purchase something?");

System.out.println();

do {

System.out.println(" [1] Show me what you have for sale.");

System.out.println(" [2] What is this place?");

System.out.println(" [3] That's some impressive weaponry. Where did it all come from?");

System.out.println(" [4] How did you get inside there?");

System.out.println(" [5] Goodbye.");

dialogueChoice = scanner.next().charAt(0);

switch (dialogueChoice) {

case '1':

System.out.println("I am ready to process our transaction.");

System.out.println();

do {

System.out.println(" [1] Purchase Items");

System.out.println(" [2] Sell Items");

System.out.println(" [3] View Player Inventory");

System.out.println(" [4] Return");

barterChoice = scanner.next().charAt(0);

switch (barterChoice) {

case '1': // BUY MENU

do {

System.out.println();

System.out.println("PURCHASE:");

System.out.println();

for (int i = 0; i < itemName.length; i++) {

System.out.println(" [" + (i + 1) + "] " + itemName[i] + ": " + itemPrice[i] + " caps (In Stock: " + botInv[i] + ")");

}

System.out.println(" [" + (itemName.length + 1) + "] Return");

purchaseChoice = scanner.nextInt();

System.out.println();

if (purchaseChoice == itemName.length + 1) {}

else if (userCaps >= itemPrice[purchaseChoice - 1] && botInv[purchaseChoice - 1] >= 1) {

System.out.println("You purchased a " + itemName[purchaseChoice - 1] + " for " + itemPrice[purchaseChoice - 1] + " caps.");

userCaps = userCaps - itemPrice[purchaseChoice - 1];

playerInv[purchaseChoice - 1] = playerInv[purchaseChoice - 1] + 1;

botInv[purchaseChoice - 1] = botInv[purchaseChoice - 1] - 1;

System.out.println();

System.out.println("Caps remaining: " + userCaps);

} else if (botInv[purchaseChoice - 1] < 1) {

System.out.println("The Gun Runners have run out of stock of this item.");

} else {

System.out.println("You do not have enough caps to make this purchase.");

}

} while (purchaseChoice != (itemName.length + 1));

break;

case '2': // SELL MENU

do {

System.out.println();

System.out.println("SELL:");

System.out.println();

for (int i = 0; i < itemName.length; i++) {

System.out.println(" [" + (i + 1) + "] " + itemName[i] + ": " + (itemPrice[i] / 2) + " caps");

}

System.out.println(" [" + (itemName.length + 1) + "] Return");

sellChoice = scanner.nextInt();

System.out.println();

if (sellChoice == itemName.length + 1) {}

else if (playerInv[sellChoice - 1] >= 1) {

System.out.println("You sold a " + itemName[sellChoice - 1] + " for " + (itemPrice[sellChoice - 1] / 2) + " caps.");

userCaps = userCaps + (itemPrice[sellChoice - 1] / 2);

playerInv[sellChoice - 1] = playerInv[sellChoice - 1] - 1;

botInv[sellChoice - 1] = botInv[sellChoice - 1] + 1;

System.out.println();

System.out.println("Caps increased to: " + userCaps);

} else {

System.out.println("You do not have any of this item to sell.");

}

} while (sellChoice != (itemName.length + 1));

break;

case '3':

System.out.println("Player Inventory:");

System.out.println();

for (int i = 0; i < itemName.length; i++) {

System.out.println(" [" + (i + 1) + "] " + itemName[i] + ": x" + playerInv[i]);

}

System.out.println();

System.out.println("Caps: " + userCaps);

System.out.println();

break;

case '4':

System.out.println();

break;

default:

System.out.println("Please select a valid option.");

}

} while (barterChoice != '4');

break;

case '2':

System.out.println("This is the New Vegas branch of the Gun Runners, supplying the wasteland with only the finest armaments since 2155.");

System.out.println();

break;

case '3':

System.out.println("All Gun Runner merchandise is constructed on-site.");

System.out.println();

break;

case '4':

System.out.println("This kiosk was specially constructed around me to deter theft and assault.");

System.out.println();

break;

case '5':

System.out.println("Come back and see us again soon.");

System.out.println();

break;

default:

System.out.println("Please select a valid dialogue option.");

}

} while (dialogueChoice != '5');

}

}

**Explanation:**

**Note:** Much of this code carries over from earlier versions of this project. This explanation will therefore focus on the updated array system.

**Note:** Java Arrays start at 0, but we want the programs menu UI to begin at 1. Therefore, +1 / -1 are used to align the users input with the actual position in the array.

– Line #52 is a do-while loop. The program will repeat until ‘purchaseChoice’ is equal to the length of the ‘itemName’ array + 1 (on Line #75).

– Line #56 is a ‘for’ loop. (

1. Int ‘I’ is set to 0.
2. Checks if the value of ‘I’ is less than the length of the array.
3. If true, the value of ‘I’ is incremented by 1.

This loop will trigger equal to the length of the array.

)

On each loop, the name (itemName), price (itemPrice) and stock (botInv) values are retrieved from their arrays, using ‘I’ as the reference position.

– Line #59 prints the ‘Return’ option, equal to the length of the array + 1. It is used to exit the for-while loop established on Line #52.

– Line #60 sets the value of int ‘purchaseChoice’ to the users input.

– Line #62 is an ‘if’ statement. This checks to see if ‘purchaseChoice’ is equal to the final position of the array + 1 (AKA the ‘Return’ option on Line #59). The if statement is blank and is used to exit the do-while loop. Otherwise, the program would attempt to access a position in the array that doesn’t exist and crash.

– Line #63 is an else-if statement. It uses the user input ‘purchaseChoice’ as a reference to access the relevant position the arrays, checking to see if the user has enough caps and if the vendor has an item in stock. If so, the relevant variables (playerInv, botInv, userCaps) are updated.

– Line #70 and Line #72 are error messages that print if the above conditions are not met.

A nearly identical system, just reversed, is used to sell items.