**IV. Test Case Suite Execution Observance and Test Driver Code Implemented.**

**Test Driver Methods added by me are 3.**

**currentState() –** which shows the current state of the finite state model.

**showAll() -** shows the current values of all the variable in the model.

**main() -** Handles the user access path of this application so that he can move from one finite state transaction to another.

import java.util.Scanner;

public class GasPump {

private int Rprice;

private int Sprice;

private int w;

private int price;

private int G;

private int total;

private int cash;

private int s;

public GasPump() {

Rprice = 0;

Sprice = 0;

w = 0;

price = 0;

G = 0;

total = 0;

cash = 0;

s = -1;

}

public int Activate(int a, int b) {

if ((s == -1) && (a > 0) && (b > 0)) {

s = 0;

Rprice = a;

Sprice = b;

System.out.print("GAS PUMP IS ON");

System.out.print("\n");

return 1;

} else {

return 0;

}

}

public int Start() {

if (s == 0) {

s = 1;

System.out.print("WELCOME!!!");

System.out.print("\n");

System.out.print("DO YOU WANT TO PAY BY CASH OR CREDIT CARD?");

System.out.print("\n");

return 1;

} else {

return 0;

}

}

public int PayCredit() {

if (s == 1) {

s = 2;

System.out.print("CHECKING CREDIT CARD.");

System.out.print("\n");

return 1;

} else {

return 0;

}

}

public int Reject() {

if (s == 2) {

s = 0;

System.out.print("CREDIT CARD IS REJECTED.");

System.out.print("\n");

return 1;

} else {

return 0;

}

}

public int Cancel() {

if (s == 3) {

s = 0;

System.out.print("TRANSACTION IS CANCELLED.");

System.out.print("\n");

if (w == 0) {

System.out.print("$");

System.out.print(cash);

System.out.print(" OF CASH IS RETURNED");

System.out.print("\n");

}

return 1;

} else {

return 0;

}

}

public int Approved() {

if (s == 2) {

s = 3;

w = 1;

System.out.print("CREDIT CARD APPROVED.");

System.out.print("\n");

System.out.print("SELECT TYPE OF GASOLINE:");

System.out.print("\n");

System.out.print("a. REGULAR");

System.out.print("\n");

System.out.print("b. SUPER");

System.out.print("\n");

return 1;

} else {

return 0;

}

}

public int PayCash(int c) {

if ((s == 1) && (c > 0)) {

s = 3;

w = 0;

cash = c;

System.out.print("SELECT TYPE OF GASOLINE:");

System.out.print("\n");

System.out.print("a. REGULAR");

System.out.print("\n");

System.out.print("b. SUPER");

System.out.print("\n");

return 1;

} else {

return 0;

}

}

public int Regular() {

if (s == 3) {

s = 4;

System.out.print("REGULAR IS SELECTED.");

System.out.print("\n");

price = Rprice;

return 1;

} else {

return 0;

}

}

public int Super() {

if (s == 3) {

s = 4;

System.out.print("SUPER IS SELECTED.");

System.out.print("\n");

price = Sprice;

return 1;

} else {

return 0;

}

}

public int StartPump() {

if (s == 4) {

s = 5;

G = 0;

total = 0;

System.out.print("PUMP IS READY TO DISPOSE ");

System.out.print("\n");

System.out.print("# OF GALLONS PUMPED: ");

System.out.print(G);

System.out.print("\n");

System.out.print("TOTAL CHARGE: $");

System.out.print(total);

System.out.print("\n");

return 1;

} else {

return 0;

}

}

public int PumpGallon() {

if (s == 5) {

if ((w == 1) || ((cash >= price \* (G + 1)) && (w == 0))) {

G = G + 1;

total = G \* price;

System.out.print("# OF GALLONS PUMPED: ");

System.out.print(G);

System.out.print("\n");

System.out.print("TOTAL CHARGE: $");

System.out.print(total);

System.out.print("\n");

System.out.print("CONTINUE PUMPING");

System.out.print("\n");

return 1;

} else if ((w == 0) && (cash < price \* (G + 1))) {

s = 6;

System.out.print("PUMP STOPPED. NOT SUFFICIENT FUNDS. ");

System.out.print("\n");

System.out.print("# OF GALLONS PUMPED: ");

System.out.print(G);

System.out.print("\n");

System.out.print("TOTAL CHARGE: $");

System.out.print(total);

System.out.print("\n");

if ((w == 0) && (total < cash)) {

System.out.print("$");

System.out.print(cash - total);

System.out.print(" OF CASH IS RETURNED");

System.out.print("\n");

}

System.out.print("DO YOU WANT A RECEIPT?");

System.out.print("\n");

return 1;

}

;

}

;

return 0;

}

public int StopPump() {

if (s == 5) {

s = 6;

System.out.print("PUMP STOPPED. ");

System.out.print("\n");

System.out.print("# OF GALLONS PUMPED: ");

System.out.print(G);

System.out.print("\n");

System.out.print("TOTAL CHARGE: $");

System.out.print(total);

System.out.print("\n");

if ((w == 0) && (total < cash)) {

System.out.print("$");

System.out.print(cash - total);

System.out.print(" OF CASH IS RETURNED");

System.out.print("\n");

}

System.out.print("DO YOU WANT A RECEIPT?");

System.out.print("\n");

return 1;

} else {

return 0;

}

}

public int NoReceipt() {

if (s == 6) {

s = 0;

System.out.print("NO RECEIPT IS PRINTED ");

System.out.print("\n");

return 1;

} else {

return 0;

}

}

public int Receipt() {

if (s == 6) {

s = 0;

System.out.print("RECEIPT IS PRINTED: ");

System.out.print("\n");

System.out.print("# OF GALLONS PUMPED: ");

System.out.print(G);

System.out.print("\n");

System.out.print("TOTAL CHARGE: $");

System.out.print(total);

System.out.print("\n");

return 1;

} else {

return 0;

}

}

public int TurnOff() {

if (s == 0) {

s = -2;

System.out.print("GAS PUMP IS TURNED OFF ");

System.out.print("\n");

return 1;

} else {

return 0;

}

}

**<Test Driver Method Added By Me>**

public int currentState()

{

return s;

}

**<Test Driver Method Added By Me>**

public void showAll()

{

System.out.println("RPRICE: "+Rprice+ "\tSPRICE: "+Sprice+"\tPRICE: "+price+"\tTOTAL: "+total+"\tCASH: "+cash);

}

**<Test Driver Method Added By Me>**

public static void main(String[] args)

{

int inputKey = 0;

do

{

System.out.println("1.Activate\n 2.Start\n 3.PayCredit\n 4.Approved \n 5.Cancel \n 6. PayCash \n 7.Regular \n 8.Super \n 9.StartPump \n 10.PumpGallon \n 11.StopPump \n 12.Receipt \n 13.NoReceipt \n 14.TurnOff");

Scanner inputScanner = new Scanner(System.in);

inputKey = inputScanner.nextInt();

GasPump objGasPump = new GasPump();

switch(inputKey)

{

case 1:

int a, b;

System.out.println("Enter Value of a:");

a = inputScanner.nextInt();

System.out.println("Enter Value of a:");

b = inputScanner.nextInt();

System.out.println("Previous State:" + objGasPump.currentState());

objGasPump.Activate(a, b);

System.out.println("New State:" + objGasPump.currentState());

objGasPump.showAll();

break;

case 2:

System.out.println("Previous State:" + objGasPump.currentState());

objGasPump.Start();

System.out.println("New State:" + objGasPump.currentState());

objGasPump.showAll();

break;

case 3:

System.out.println("Previous State:" + objGasPump.currentState());

objGasPump.PayCredit();

System.out.println("New State:" + objGasPump.currentState());

objGasPump.showAll();

break;

case 4:

System.out.println("Previous State:" + objGasPump.currentState());

objGasPump.Approved();

System.out.println("New State:" + objGasPump.currentState());

objGasPump.showAll();

break;

case 5:

System.out.println("Previous State:" + objGasPump.currentState());

objGasPump.Cancel();

System.out.println("New State:" + objGasPump.currentState());

objGasPump.showAll();

break;

case 6:

int cash = inputScanner.nextInt();

System.out.println("Previous State:" + objGasPump.currentState());

objGasPump.PayCash(cash);

System.out.println("New State:" + objGasPump.currentState());

objGasPump.showAll();

break;

case 7:

System.out.println("Previous State:" + objGasPump.currentState());

objGasPump.Regular();

System.out.println("New State:" + objGasPump.currentState());

objGasPump.showAll();

break;

case 8:

System.out.println("Previous State:" + objGasPump.currentState());

objGasPump.Super();

System.out.println("New State:" + objGasPump.currentState());

objGasPump.showAll();

break;

case 9:

System.out.println("Previous State:" + objGasPump.currentState());

objGasPump.StartPump();

System.out.println("New State:" + objGasPump.currentState());

objGasPump.showAll();

break;

case 10:

System.out.println("Previous State:" + objGasPump.currentState());

objGasPump.PumpGallon();

System.out.println("New State:" + objGasPump.currentState());

objGasPump.showAll();

break;

case 11:

System.out.println("Previous State:" + objGasPump.currentState());

objGasPump.StopPump();

System.out.println("New State:" + objGasPump.currentState());

objGasPump.showAll();

break;

case 12:

System.out.println("Previous State:" + objGasPump.currentState());

objGasPump.Receipt();

System.out.println("New State:" + objGasPump.currentState());

objGasPump.showAll();

break;

case 13:

System.out.println("Previous State:" + objGasPump.currentState());

objGasPump.NoReceipt();

System.out.println("New State:" + objGasPump.currentState());

objGasPump.showAll();

break;

case 14:

System.out.println("Previous State:" + objGasPump.currentState());

objGasPump.TurnOff();

System.out.println("New State:" + objGasPump.currentState());

objGasPump.showAll();

System.exit(0);

break;

}

}while(inputKey<15);

}

}