Student Name: Kamalakar Duvva

Instructor Name :Mike Tizio

Practical Java Programming

Rivier University

**1.Description**

**2. pseudo code**

**3.class Diagram**

**4.Test cases**

**5.Output screenshots**

**1.Description**

This application explore the main concepts of Object-Oriented Programming like Inheritance , Exception Handling, Interface , Collection Framework and File I/O.

This is the application for simple shopping operations includes adding new items ,buying and updating quantity .The application takes the item details from input.txt file , than program display the options for shopping . The output of shopping summary written into file receipt.txt.

**2. pseudo code:**

**1. reading inputs from file**

**input: File Object**

**output: Display items on console**

1. Create buffered reader object

2. use buffer\_reader.readLine() method

3. Store output in string = line

4. Iterate for loop up to (line != null)

5.print line on console

6. end loop

7.end

**2. Show details priority wise**

**input: int[] - item\_Priority, Objects[] -shopping\_items**

**output: Display list**

1. set int temp = 0

2. set i = 0

3. Iterate for loop up to end of array(i < item\_priority[].length)

4. Store item\_prioroty[i] in temp

5.set j = 0

6. Iterate for loop up to end of array(j >shopping\_items[].length)

7. **if** (shopping\_items[j].getPriority() == temp\_priority)then

i. store minValue = numericset[i]

ii.print shopping\_items[j].getItem\_name()

shopping\_items[j].getItem\_price()

shopping\_items[j].getPriority()

end if

8. increment j by 1

9. end for loop

10. increment i by 1

11. end for loop

12. end

**3. Search for an Item**

**input :array -item\_cart[] ,double -search\_value**

**output: boolean-true/false**

1. set i = 0

2. Iterate for loop up to end of array( i < item\_cart[].length)

3. if(item\_cart[i] == search\_value)

4. return true

5.increment i by 1

6. end for loop

7.return false

9.end

**4. Verify special characters, white spaces and digits from item Name**

**input : String - inputText**

**output: String- result**

1. set String result = null

2. perform regular expression operation on input text allow only characters

inputText.replaceAll("[^a-zA- Z]+", "");

3.Store the result in variable result

4. return result

**5. checking total price more than 100**

**input : Reading price from console by Keyboard Object of Scanner class**

**output: prices of all items more than 100**

1. Print Message "Please enter price for an item"

2. **if**(item\_price <= 0) then

print give a valid 'price'

3. perform total\_cost = total\_cost + item\_price

4. **if**(total\_cost < 100 && i == 6) then

**do**{

print the total cost should be more then 100 , please enter price for 7 th Item"

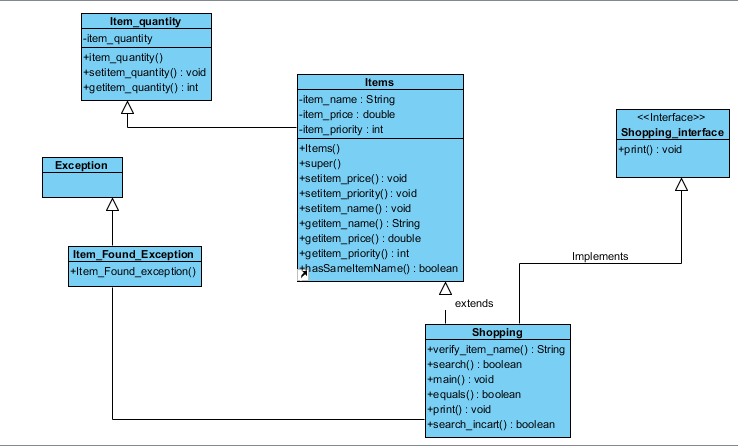
item\_price=keyboard.nextDouble();

iterate **while**((total\_cost + item\_price) < 100);

5. end if

6.end

**3.Class Diagram :**



**Test Plan**

**Test ID:** 1  **Requirement Addressed:** Read Items from file

**Objective:** Ensure item name allow only alphabets without any spaces ,digits and special characters. Read data Correct format

**Test Cases**

**Interface ID Data Field Inputs**

1) REQ1-1 Item name Apple

2) REQ1-2 Item name blue berry

3) REQ1-3 Item name 25apple

4) REQ1-4 Item name &^apple

5) REQ1-5 Item name

6) REQ1-6 Item name 123

7) REQ1-7 Item name %123

8) REQ1-8 File not present

9REQ1-9 File empty file

**Expected Results:**

Test 1 to 4 are valid ,and remaining all are rejected.

**Actual Results:**

Test 1 to 2 are accepted directly. Test 3,4 accepted by Explicit verification 5,6,7,8,9 were rejected with error message.

For Test 5,9 we may get exception "java.lang.NullPointerException"

For test 8 we get Exception "java.io.FileNotFoundException"

**-------------------------------------------------------------------------------------------------------------------------------------**

**Test ID:** 2  **Requirement Addressed:** To verify the prices

**Objective:** Ensure that the 7 numeric item prices allow only integers and double

**Test Cases**

**Interface ID Data Field Value Entered**

1) REQ2-1 price int

2) REQ2-2 price double

3) REQ2-3 price float

4) REQ2-4 price long

5) REQ2-5 price short

6) REQ2-6 price char

7) REQ2-7 price byte

8) REQ2-8 price String

9) REQ2-9 price boolean

**Expected Results:**

Test 1 to 2 are valid ,and remaining all are rejected.

**Actual Results:**

Test 1 to 5 are accepted directly. Test 3,4,5accepted by Explicit Conversion 6,7,8,9 were rejected with error message.

For Test 6,7,8,9 we get Exception "java.util.InputMismatchException"

**Test ID:** 3  **Requirement Addressed:** To test item priority 1 to 7 only

**Objective:** Ensure that the 7 numeric item priority allow only integers and that should be 1 to 7

**Test Cases**

**Interface ID Data Field Value Entered**

1) REQ3-1 priority int

2) REQ3-2 priority double

3) REQ3-3 priority float

4) REQ3-4 priority long

5) REQ4-5 priority short

6) REQ3-6 priority char

7) REQ3-7 priority byte

8) REQ3-8 priority String

9) REQ3-9 priority boolean

**Expected Results:**

Test 1 is valid ,and remaining all are rejected.

**Actual Results:**

Test 1 is accepted directly. Test all other were rejected with error message.

For Test 2-9 we get Exception " java.lang.NumberFormatException"

**Output Screen:**

