|  |  |  |
| --- | --- | --- |
|  | Name | Shubham Goel |
|  | Roll No | 2019130015 |
| **OOP Lab** | Batch | A |
| Exp 6: Exceptions | Date | 18 October 2021 |
|  | Branch | COMPS |

**Aim:** Implement a user defined exception

**Theory**

Exception Handling

The Exception Handling in Java is one of powerful mechanisms to handle the runtime errors so that the normal flow of the application can be maintained.

Throw Keyword

The throw keyword in Java is used to explicitly throw an exception from a method or any block of code. We can throw either [checked or unchecked exception](https://www.geeksforgeeks.org/checked-vs-unchecked-exceptions-in-java/). The throw keyword is mainly used to throw custom exceptions.

**Code:**

import java.util.\*;

import java.lang.\*;

//custom exception

class InvalidPrice extends Exception {

    public InvalidPrice(String str) {

        super(str);

    }

}

// Base class

class User {

    String name;

    int age;

    void printUserStamp() {

        System.out.println("");

        System.out.println("Inside the user class");

        System.out.println("");

    }

    User() {

        this.name = "a\_user";

        this.age = 100;

    }

    void printUserInfo() {

    };

    void inputNameAge() {

        // printUserStamp();

        Scanner scanner\_obj = new Scanner(System.in); // Create a Scanner object

        System.out.println("Enter the name of the user");

        this.name = scanner\_obj.nextLine();

        System.out.println("Enter the age");

        this.age = scanner\_obj.nextInt();

    }

    void printUserDetails() {

        System.out.println(this.name);

        System.out.println(this.age);

    }

}

class Seller extends User{

    String GSTno;

    int rating;

    Product sellerProducts[] = new Product[100]; //aggregation

    int lastFreeIndex = 0;

    Seller() {

        this.GSTno = "123";

        this.rating = 3;

    }

    void addProduct(String name, int price) {

        Product addProduct = new Product(name, price);

        this.sellerProducts[this.lastFreeIndex++] = addProduct;

    }

    void printSellerProducts() {

        for (int i = 0; i < this.lastFreeIndex; i++) {

            System.out.println("Index:" + i);

            System.out.println("price = " + this.sellerProducts[i].price);

            System.out.println("name = " + this.sellerProducts[i].name);

            System.out.println();

        }

    }

    void printSellerInfo() {

        printUserDetails();

        System.out.println(this.GSTno);

        System.out.println(this.rating);

    }

    void printSellerStamp() {

        System.out.println("");

        System.out.println("Inside the Seller class");

        System.out.println("");

    }

    void inputSellerDetails() {

        // printSellerStamp();

        inputNameAge();

        Scanner scan\_obj = new Scanner(System.in); // Create a Scanner object

        System.out.println("Enter GSTno for the user");

        this.GSTno = scan\_obj.nextLine();

        System.out.println("Enter the rating for the seller");

        this.rating = scan\_obj.nextInt();

        // scan\_obj.close();

    }

}

class Buyer extends User {

    String address;

    String cardID;

    void printBuyerInfo() {

        printUserDetails();

        System.out.println(this.address);

        System.out.println(this.cardID);

    }

    void printBuyerStamp() {

        System.out.println("");

        System.out.println("Inside the buyer class");

        System.out.println("");

    }

    void inputBuyerDetails() {

        // printBuyerStamp();

        inputNameAge();

        Scanner scan\_obj = new Scanner(System.in); // Create a Scanner object

        System.out.println("Enter address for the user");

        this.address = scan\_obj.nextLine();

        System.out.println("Enter the credit card id");

        this.cardID = scan\_obj.nextLine();

        // scan\_obj.close();

    }

    //composition

    class Cart {

        Product cartArr[] = new Product[100];

        int firstFreeIndex;

        void printCartDetails() {

            for (int i = 0; i < firstFreeIndex; i++) {

                System.out.println(this.cartArr[i].name);

            }

        }

        void addProductToCart(Product productToAdd) {

            this.cartArr[this.firstFreeIndex++] = productToAdd;

        }

    }

}

class Product {

    String name;

    int price;

    static Product allProducts[] = new Product[400];

    static int lastProductLoc = 0;

    static void addToAllProducts(Product toAdd) {

        allProducts[lastProductLoc++] = toAdd;

    }

    Product(String name, int price) {

        this.name = name;

        this.price = price;

        addToAllProducts(this);

    }

    static void printAllProducts() {

        for (int i = 0; i < lastProductLoc; i++) {

            System.out.println("Index:" + i);

            System.out.println("price = " + allProducts[i].price);

            System.out.println("name = " + allProducts[i].name);

            System.out.println();

        }

    }

    void printProductDetails() {

        System.out.println("price = " + this.price);

        System.out.println("name = " + this.name);

        System.out.println();

    }

}

public class Main {

    //validate the price

    static void validatePrice(int price) throws InvalidPrice {

        if (price < 0) {

            throw new InvalidPrice("The price should be positive");

        }

        else {

            System.out.println("The price is valid");

        }

    }

    public static void main(String args[]){

        Scanner scan\_obj = new Scanner(System.in); // Create a Scanner object

        System.out.println("Enter the number of products you want to add");

        int numProducts;

        Product aProduct;

        numProducts = scan\_obj.nextInt();

        for (int i = 0; i < numProducts; i++) {

            System.out.println("Enter the name of the product");

            String aName = scan\_obj.nextLine();

            aName = scan\_obj.nextLine();

            System.out.println("Enter the price of the product");

            int aPrice = scan\_obj.nextInt();

            //catch the exception

            try{

                validatePrice(aPrice);

            }

            catch (InvalidPrice exceptionPrice) {

                //printing the exception

                System.out.println("The exception has occurred");

                System.out.println("Exception is " + exceptionPrice);

                i--;

                continue;

            }

            aProduct = new Product(aName, aPrice);

            System.out.println("Received product info");

            aProduct.printProductDetails();

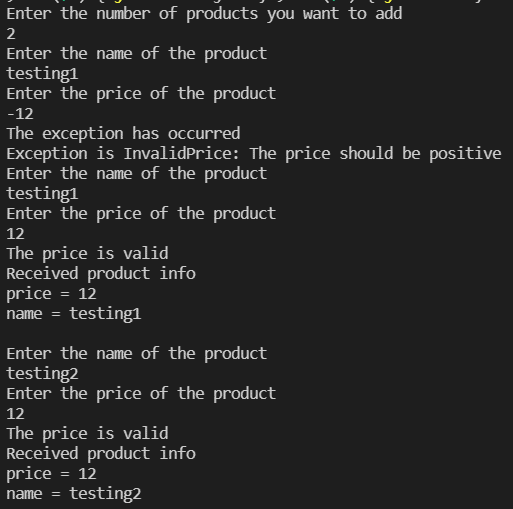
        }

    }

}

}

**Output:**



**Conclusion**

I have learnt and implemented Exception handling. I have also understood the difference between exceptions and errors.

References:

<https://www.geeksforgeeks.org/throw-throws-java/>

<https://www.javatpoint.com/exception-handling-in-java>