|  |  |  |
| --- | --- | --- |
|  | Name | Shubham Goel |
|  | Roll No | 2019130015 |
| **OOP Lab** | Batch | A |
| Exp: Generics | Date | 22 Nov. 21 |
|  | Branch | COMPS |

**Aim:** Implement Generics

**Theory**

Generics are a facility of [generic programming](https://en.wikipedia.org/wiki/Generic_programming) that were added to the [Java programming language](https://en.wikipedia.org/wiki/Java_(programming_language)) in 2004. They were designed to extend Java's [type system](https://en.wikipedia.org/wiki/Type_system) to allow "a type or method to operate on objects of various types while providing compile-time type safety". The aspect compile-time type safety was not fully achieved, since it was shown in 2016 that it is not guaranteed in all cases.

The [Java collections framework](https://en.wikipedia.org/wiki/Java_collections_framework) supports generics to specify the type of objects stored in a collection instance.

**Code:**

**Main.java**

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 Cart {

    // Product cartArr[] = new Product[100];

    Collection<Product> cartArr = new ArrayList<Product>();

    int firstFreeIndex;

    void printCartDetails() {

        for (Product i : cartArr) {

            System.out.println("Product in cart " + i.name);

        }

    }

    void addProductToCart(Product productToAdd) {

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

        this.cartArr.add(productToAdd);

    }

    void buyProductsInCart() {

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

        for (Product i : cartArr) {

            System.out.println("User bought " + i.name);

        }

    }

}

class Buyer extends User {

    String address;

    String creditCardID;

    Cart cart;

    Buyer(String name) {

        this.cart = new Cart();

        this.name = name;

    }

    void printBuyerInfo() {

        printUserDetails();

        System.out.println(this.address);

        System.out.println(this.creditCardID);

    }

    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.creditCardID = scan\_obj.nextLine();

        // scan\_obj.close();

    }

}

// composition

class Product {

    String name;

    int price;

    int quantity;

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

    // collection has no index

    static List<Product> allProducts = new ArrayList<Product>();

    static int lastProductLoc = 0;

    static void addToAllProducts(Product toAdd) {

        // allProducts[lastProductLoc++] = toAdd;

        allProducts.add(toAdd);

    }

    Product(String name, int price) {

        this.name = name;

        this.price = price;

        this.quantity = 10;

        addToAllProducts(this);

    }

    static void printAllProducts() {

        int i = 0;

        for (Product cur\_product : allProducts) {

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

            System.out.println("price = " + cur\_product.price);

            System.out.println("name = " + cur\_product.name);

            System.out.println();

            i++;

        }

    }

    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[]) {

        Buyer b = new Buyer("second");

        Product pa = new Product("p1", 200);

        Product pb = new Product("p2", 200);

        Product pc = new Product("p3", 200);

        Product pd = new Product("p4", 200);

        Product pe = new Product("p5", 200);

        System.out.println("All the products are");

        Product.printAllProducts();

        System.out.println("Enter products u want");

        System.out.println("Enter -1 to exit");

        for (int i = 0; i < 100; i++) {

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

            int cart\_loc = scanner\_obj.nextInt();

            if (cart\_loc == -1) {

                break;

            }

            b.cart.addProductToCart(Product.allProducts.get(cart\_loc)); // use list index

            System.out.println("Product " + Product.allProducts.get(cart\_loc).name + " added to cart");

        }

        b.cart.printCartDetails();

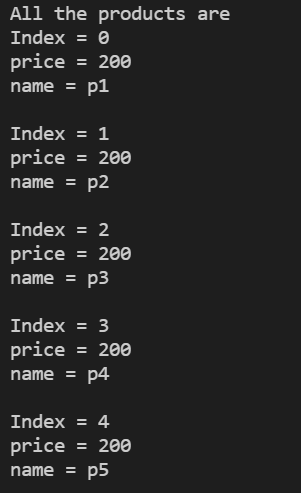
        b.cart.buyProductsInCart();

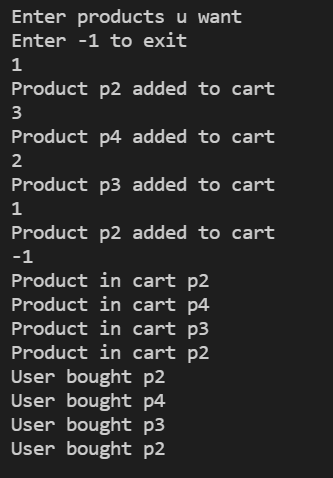
        System.out.println();

    }

}

**Output:**





**Conclusion**

I have learnt and implemented Generics.

**References:**

https://en.wikipedia.org/wiki/Generics\_in\_Java