***Dt : 9/11/2022***

***\*imp***

***Set<E> holding User defined class Objects:***

***Product.java***

***package test;***

***@SuppressWarnings("rawtypes")***

***public class Product extends Object implements Comparable{***

***public String code,name;***

***public float price;***

***public int qty;***

***public Product(String code,String name,float price,int qty) {***

***this.code=code;***

***this.name=name;***

***this.price=price;***

***this.qty=qty;***

***}***

***@Override***

***public String toString() {***

***return code+"\t"+name+"\t"+price+"\t"+qty;***

***}***

***@Override***

***public int compareTo(Object o) {***

***Product prod = (Product)o;//DownCasting process***

***int c = code.compareTo(prod.code);***

***if(c==0) return 0;***

***else if(c>0) return 1;***

***else return -1;***

***}***

***}***

***DemoSet2.java(MainClass)***

***package maccess;***

***import java.util.\*;***

***import test.Product;***

***public class DemoSet2 {***

***public static void main(String[] args) {***

***Scanner s = new Scanner(System.in);***

***String c = null;***

***Set<Product> ob = null;***

***try(s;){***

***try {***

***while(true) {***

***System.out.println("====Choice====");***

***System.out.println***

***("1.HashSet\n2.LinkedHashSet\n3.TreeSet\n4.exit");***

***System.out.println("Enter the choice:");***

***switch(Integer.parseInt(s.nextLine()))***

***{***

***case 1:***

***ob = new HashSet<Product>();***

***c = "HashSet<E>";***

***break;***

***case 2:***

***ob = new LinkedHashSet<Product>();***

***c = "LinkedHashSet<E>";***

***break;***

***case 3:***

***ob = new TreeSet<Product>();***

***c = "TreeHashSet<E>";***

***break;***

***case 4:***

***System.out.println("Set operation Stopped...");***

***System.exit(0);***

***default : System.out.println("Invalid choice");***

***continue; //skip the below lines from the Iteration***

***}//end of switch***

***System.out.println("\*\*\*\*perform operations ob "+c+"\*\*\*\*");***

***xyz:***

***while(true) {***

***System.out.println("====Choice====");***

***System.out.println("1.add\n2.remove\n3.exit");***

***System.out.println("Enter the Choice:");***

***switch(Integer.parseInt(s.nextLine()))***

***{***

***case 1:***

***System.out.println("Enter the Code:");***

***String code = s.nextLine();***

***System.out.println("Enter the Name:");***

***String name = s.nextLine();***

***System.out.println("Enter the Price:");***

***float price = Float.parseFloat(s.nextLine());***

***System.out.println("Enter the Qty:");***

***int qty = Integer.parseInt(s.nextLine());***

***ob.add(new Product(code,name,price,qty));***

***System.out.println("=====Iterator<E>=====");***

***Iterator<Product> it1 = ob.iterator();***

***while(it1.hasNext())***

***{***

***System.out.println(it1.next());***

***}//end of loop***

***System.out.println("=====Iterator<E>(Java8)=====");***

***Iterator<Product> it2 = ob.iterator();***

***it2.forEachRemaining((p)->***

***{***

***System.out.println(p.toString());***

***});***

***System.out.println("=====Spliterator<T>(Java8)======");***

***Spliterator<Product> sp = ob.spliterator();***

***sp.forEachRemaining((q)->***

***{***

***System.out.println(q.toString());***

***});***

***System.out.println("=====forEach()(Java8)====");***

***ob.forEach((k)->***

***{***

***System.out.println(k.toString());***

***});***

***break;***

***case 2:***

***if(ob.isEmpty()) {***

***System.out.println("Set<E> is empty...");***

***}else {***

***System.out.println("Enter the ele(prodCode) to be removed:");***

***String ele = s.nextLine();***

***Iterator<Product> it4 = ob.iterator();***

***while(it4.hasNext()) {***

***Product p = (Product)it4.next();***

***if(p.code.equals(ele)) {***

***if(ob.remove(p)) {***

***System.out.println("Ele removed Successfully...");***

***break;***

***}***

***}***

***}//end of loop***

***System.out.println(ob.toString());***

***}***

***break;***

***case 3:***

***System.out.println("Operation on "+c+" Stopped");***

***break xyz;***

***default:***

***System.out.println("Invalid choice...");***

***}//end of switch***

***}//end of loop***

***}//end of loop***

***}catch(Exception e) {e.printStackTrace();}***

***}//end of try***

***}***

***}***

***=====================================================***

***define Iterator<E>?***

***=>Iterator<E> is an interface from java.util package and which is used to***

***retrieve elements from Collection<E> objects in forward direction.***

***=>The following are some important methods of Iterator<E>:***

***public abstract boolean hasNext();***

***public abstract E next();***

***public default void forEachRemaining***

***(java.util.function.Consumer<? super E>);***

***=>we use iterator() method to create implementation object for Iterator<E>***

***interface.***

***syntax:***

***Iterator<Product> it1 = ob.iterator();***

***Note:***

***=>By Java8 version,Iterator<E> is added with forEachRemaining() method and which***

***works on LambdaExpression.***

***======================================================================***