***dt : 17/11/2022***

***3.Enumeration<E>:***

***=>Enumeration is an interface from java.util package and which is used to***

***retrieve elements from Vector<E> objects.***

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

***public abstract boolean hasMoreElements();***

***public abstract E nextElement();***

***public default java.util.Iterator<E> asIterator();***

***=>we use "elements()" method to create the implementation object for Enumeration<E>***

***interface.***

***4.Spliterator<T>:***

***=>Spliterator<T> interface introduced by Java8 version and which is used to***

***retrieve elements from Array Objects and Collection<E> objects.***

***Ex : CursorStatements2.java***

***package maccess;***

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

***public class CursorStatements2 {***

***@SuppressWarnings("removal")***

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

***Vector<Integer> v = new Vector<Integer>();***

***for(int i=1;i<=10;i++)***

***{***

***v.add(new ~~Integer~~(i));***

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

***System.out.println("\*\*\*\*Enumeration<E>\*\*\*\*");***

***Enumeration<Integer> e = v.elements();***

***while(e.hasMoreElements()) {***

***System.out.print(e.nextElement()+" ");***

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

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

***Enumeration<Integer> e2 = v.elements();***

***Iterator<Integer> it = e2.asIterator();***

***it.forEachRemaining((k)->***

***{***

***System.out.print(k.toString()+" ");***

***});***

***}***

***}***

***o/p:***

***\*\*\*\*Enumeration<E>\*\*\*\****

***1 2 3 4 5 6 7 8 9 10***

***\*\*\*\*Iterator<E>\*\*\*\****

***1 2 3 4 5 6 7 8 9 10***

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

***faq:***

***define forEach() method?***

***=>forEach() method introduced by Java8 version and which is used to retrieve***

***elements from Collection<E> and Map<K,V> objects.***

***Method Signature of forEach() on Collection<E>:***

***public default void forEach(java.util.function.Consumer<? super T>);***

***Method Signature of forEach() on Map<K,V>:***

***public default void forEach***

***(java.util.function.BiConsumer<? super K, ? super V>);***

***faq:***

***define Consumer<T>?***

***=>Consumer<T> is a functional interface introduced by Java8 version and which***

***provide abstract method "accept(T)" to hold LambdaExpression passed as parameter***

***to forEach() method on Collection<E> objects.***

***structure of Consumer<T>:***

***public interface java.util.function.Consumer<T>***

***{***

***public abstract void accept(T);***

***}***

***faq:***

***define BiConsumer<T,U>?***

***=>BiConsumer<T,U> is a functional interface introduced by Java8 version and which***

***provide abstract method "accept(T,U)" to hold LambdaExpression passed as parameter***

***to forEach() method on Map<K,V> objects.***

***structure of BiConsumer<T>:***

***public interface java.util.function.BiConsumer<T, U>***

***{***

***public abstract void accept(T, U);***

***}***

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

***\*imp***

***define Enum<E>?***

***=>Enum<E> is an abstract class from java.lang package.***

***=>we use "enum" keyword to declare implementation of Enum<E>***

***syntax:***

***enum Enum\_name***

***{***

***//elements***

***//variables***

***//methods***

***}***

***=>Enum<E> can hold elements,variables,Constructors and methods.***

***=>The constructors which are declared within the Enum<E> are automatically***

***"private" constructors.***

***Ex :***

***Cars.java***

***package test;***

***public enum Cars {***

***figo(1200),alto(1400),dezire(1600);***

***public int price;***

***private Cars(int price)***

***{***

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

***}***

***public final int getPrice() {***

***return price;***

***}***

***public final void setPrice(int price) {***

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

***}***

***}***

***DemoEnum.java(MainClass)***

***package maccess;***

***import test.Cars;***

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

***public class DemoEnum {***

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

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

***Cars c[] = Cars.values();***

***System.out.println("====Display Enum<E> objects====");***

***for(Cars k : c)***

***{***

***System.out.println(k.toString()+" Costs "+k.price+" thousand dollars");***

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

***System.out.println("=====Set values using Setter methods===");***

***for(Cars z : c)***

***{***

***System.out.println("Price for "+z.toString());***

***z.setPrice(s.nextInt());***

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

***System.out.println("===get Values using Getter methods===");***

***for(Cars y : c)***

***{***

***System.out.println("cost of "+y.toString()+" is "+y.getPrice()+***

***" thousand dollars");***

***}***

***s.close();***

***}***

***}***

***o/p:***

***====Display Enum<E> objects====***

***figo Costs 1200 thousand dollars***

***alto Costs 1400 thousand dollars***

***dezire Costs 1600 thousand dollars***

***=====Set values using Setter methods===***

***Price for figo***

***1700***

***Price for alto***

***1900***

***Price for dezire***

***2000***

***===get Values using Getter methods===***

***cost of figo is 1700 thousand dollars***

***cost of alto is 1900 thousand dollars***

***cost of dezire is 2000 thousand dollars***

***Diagram:***

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

***Note:***

***=>In realtime Enum<E> is used in the applications where we have defined list***

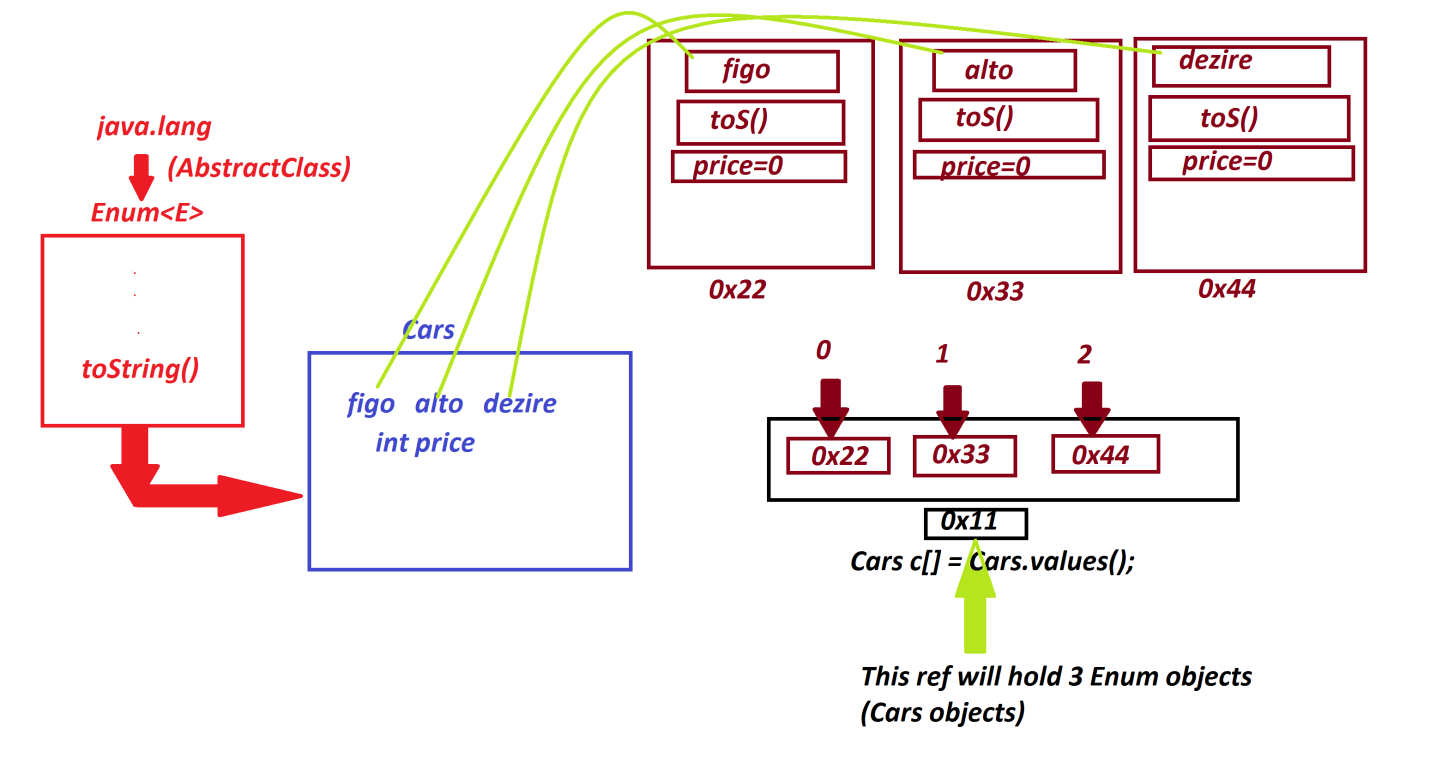
***of elements.***

***Ex:***

***week days***

***Months***

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

******