Dt : 23/9/2022

Assignment:(Solution)

wap to read a string and reverse the words of given String?

Ex : StringTokenizer2.java

```
package maccess;
import java.util.*;;
public class DemoTokenizer2 {
      public static void main(String[] args) {
       Scanner s = new Scanner(System.in);
       System.out.println("Enter the String:");
       String str = s.nextLine();
       System.out.println("str : "+str.toString());
       StringTokenizer ob = new StringTokenizer(str," ");
       System.out.println("====Details from StringTonizer====");
       while(ob.hasMoreTokens())
       {
       String tk = ob.nextToken();
       StringBuffer sb = new StringBuffer(tk);
       System.out.print(sb.reverse()+" ");
       }//end of while
       s.close();
      }
}
```

o/p:

Enter the String:

java language program

str : java language program

====Details from StringTonizer====

avaj egaugnal margorp

==========

faq:

define Utility classes?

=>The classes which are used to perform operations on other objects

are known as Utility classes.

==========

**imp*

WrapperClasses in Java:

=>The pre-defined classes which are used to make Primitive datatypes

available in the form of objects are known as WrapperClasses.

=>Every Primitive Datatype will have its own WrapperClass and there

are 8 WrapperClasses from 'java.lang' package.

=>List of WrapperClasses:

data_type	WrapperClass
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
char	Character
boolean	Boolean

---

faq:

define Boxing process?

=>The process of binding primitive datatype values into WrapperClass objects is known as Boxing process.

=>we use constructors to perform boxing process

=>List of constructors from WrapperClases:

WrapperClass	Constructors
Byte	Byte(byte),Byte(String)
Short	Short(short),Short(String)
Integer	Integer(int),Integer(String)
Long	Long(long),Long(String)
Float	Float(float),Float(String),Float(double)
Double	Double(double),Double(String)
Character	Character(char)
Boolean	Boolean(boolean),Boolean(String)

Ex : DemoWrapperClass1.java

```
package maccess;
public class DemoWrapperClass1 {
	@SuppressWarnings("removal")
	public static void main(String[] args) {
		//Boxing process
		Integer ob1 = new Integer(12);
		Integer ob2 = new Integer("13");
		Float ob3 = new Float(12.34F);//float_para
		Float ob4 = new Float(234.45);//double_para
		Float ob5 = new Float("14.56F");//float in string
		Character ob6 = new Character('A');
		Boolean ob7 = new Boolean(true);
		Boolean ob8 = new Boolean("false");
		System.out.println("====data from Objects====");
		System.out.println("ob1:"+ob1.toString());
		System.out.println("ob2:"+ob2.toString());
```

```
        System.out.println("ob3:"+ob3.toString());
        System.out.println("ob4:"+ob4.toString());
        System.out.println("ob5:"+ob5.toString());
        System.out.println("ob6:"+ob6.toString());
        System.out.println("ob7:"+ob7.toString());
        System.out.println("ob8:"+ob8.toString());

    }
}
```

o/p:

====data from Objects====

ob1:12

ob2:13

ob3:12.34

ob4:234.45

ob5:14.56

ob6:A

ob7:true

ob8:false

==========

faq:

define AutoBoxing process?

=>The Boxing process which is performed automatically is known as AutoBoxing process.

=>In AutoBoxing process we assign Primitive datatype values to NonPrimitive datatype variables.

Ex : DemoWrapperClass2.java

```
package maccess;
public class DemoWrapperClass2 {
	public static void main(String[] args) {
		//AutoBoxing process
		Integer ob1 = 12;
		Float ob2 = 12.34F;//float_para
		Character ob3 = 'A';
		Boolean ob4 = true;
		System.out.println("====data from Objects====");
		System.out.println("ob1:"+ob1.toString());
		System.out.println("ob2:"+ob2.toString());
		System.out.println("ob3:"+ob3.toString());
		System.out.println("ob4:"+ob4.toString());
	}
}
```

o/p:

====data from Objects====

ob1:12

ob2:12.34

ob3:A

ob4:true

==========

faq:

define UnBoxing process?

=>The process of taking primitive datatype values outof WrapperClass objects is known as UnBoxing process.

=>we use the following methods to perform UnBoxing process:

public byte byteValue();

public short shortValue();

public int intValue();

public long longValue();

public float floatValue();

public double doubleValue();

public char charValue();

public boolean booleanValue();

Ex : DemoWrapperClass3.java

```
package maccess;
public class DemoWrapperClass3 {
	@SuppressWarnings("removal")
	public static void main(String[] args) {
		//Boxing process
		Integer ob1 = new Integer(12);
		Float ob2 = new Float(12.34F);//float_para
		Character ob3 = new Character('A');
		Boolean ob4 = new Boolean(true);
		  //UnBoxing process
		int i = ob1.intValue();
		float f = ob2.floatValue();
		char ch = ob3.charValue();
		boolean b = ob4.booleanValue();
		System.out.println("====data after UnBoxing====");
		System.out.println("i:"+i);
		System.out.println("f:"+f);
		System.out.println("ch:"+ch);
		System.out.println("b:"+b);
	}
}
```

o/p:

====data after UnBoxing====

i:12

f:12.34

ch:A

b:true

==========

faq:

define AutoUnBoxing process?

=>The UnBoxing process which is performed automatically is known as AutoUnBoxing process.

=>In AutoUnBoxing process the NonPrimitive datatype variables are assigned to Primitive datatype variables.

Ex : DemoWrapperClass4.java

```
package maccess;
public class DemoWrapperClass4 {
	public static void main(String[] args) {
		//AuoBoxing process
		Integer ob1 = 12;
		Float ob2 = 12.34F;//float_para
		Character ob3 = 'A';
		Boolean ob4 = true;
		//UnBoxing process
		int i = ob1;
		float f = ob2;
		char ch = ob3;
		boolean b = ob4;
		System.out.println("====data after UnBoxing====");
		System.out.println("i:"+i);
		System.out.println("f:"+f);
		System.out.println("ch:"+ch);
		System.out.println("b:"+b);
	}
}
```

o/p:

====data after UnBoxing====

i:12

f:12.34

ch:A

b:true

==========

Note:

=>We have to make primitive datatype values available in the form of objects,because Java-Frameworks will hold data only in the form objects.

=>All WrapperClass objects are automatically Immutable objects.

==========