15. Internationalization (I18N)

I18N :— The process of designing applications in such a way that which supports various countries & various currencies automatically is called Internationalization (I18N).

Forexample, If the request is coming from India then the response should be in India specific form and if the request is coming from US then the response should be US people understandable form.

→ We can implement I18N by using the following 3 classes.

1) Locale
2) NumberFormat
3) DateFormat

1) Locale :—

→ A Locale object represents a geographic location or language.

→ Locale class present in java.util package & it is a final class.

→ It is the direct child class of Object & implements Serializable and cloneable interfaces.

Creation of Locale object :—

→ We can create Locale object by using the following constructors of Locale class.

```
Locale l=new Locale(String language);
Locale l=new Locale(String language, String country);
```

Ex: `Locale l=new Locale("pa", "IN");`

"pa" → Panjabi, "IN" → India

→ Locale class already contains some predefined constants to represent some standard Locales.

→ We can use these Locale constants directly.

Ex: Locale.US
Locale.UK
Locale.ITALY
Locale.ENGLISH.

Important methods of Locale class:—

```
public static Locale getDefault();
public static void setDefault(Locale l);
public String getCountry(); US
public String getLanguage(); en
public String getDisplayCountry(); United States
public String getDisplayLanguage(); english
public static String[] getISOLanguages();
public static String[] getISOCountries();
public static Locale[] getAvailableLocales();
```

Ex:

```
import java.util.*;
class LocaleDemo1
{
  p s v m(-)
  {
    Locale l1=Locale.getDefault();
    S.o.p(l1.getCountry()+"..."+l1.getLanguage());
    S.o.p(l1.getDisplayCountry()+"..."+l1.getDisplayLanguage());
    Locale l2=new Locale("pa","IN");
    Locale.setDefault(l2);
```

```
S.o.p(Locale.getDefault().getDisplayLanguage());
String[] s3 = Locale.getISOLanguages();
for(String s4 : s3)
{
  S.o.p(s4);
}
String[] s4 = Locale.getISOCountries();
for(String s5 : s4)
{
  S.o.p(s5);
}
Locale[] s = Locale.getAvailableLanguages();
for(Locale s1 : s)
{
  S.o.p(s1.getDisplayCountry() + "..." + s1.getDisplayLanguage());
}
}
}
```

2) NumberFormat class :—

→ Various countries follow various styles to represent a number.

Ex: Double d = 123456.789;

IN : 1,23,456.789

US : 123,456.789

ITALY : 123.456,789

→ By using NumberFormat class we can format a Java number according to a particular Locale.

→ NumberFormat class present in java.text package and it is an abstract class.

NumberFormat nf = new NumberFormat(); ✗

Getting NumberFormat object for Default Locale :-

→ NumberFormat class defines the following methods.

```
public static NumberFormat getInstance();
public static NumberFormat getCurrencyInstance();
public static NumberFormat getPercentInstance();
public static NumberFormat getNumberInstance();
```

Getting NumberFormat object for Specific Locale :-

→ The above methods are exactly same except that we have to pass Locale object as argument.

Ex: public static NumberFormat getNumberInstance(Locale l);

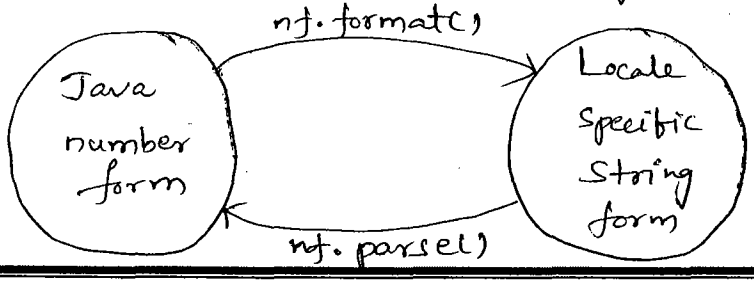
→ Once we get NumberFormat object we can call format() & parse() methods on that object.

```
public String format(long l);
public String format(double d);
```

to convert java no. form to locale specific String form.

```
public Number parse(String s) throws ParseException
```

To convert locale specific String form to java number form.



// Write a program to represent Java no. in Italy specific form.

```
import java.text.*;
import java.util.*;
class NumberFormatDemo2
{
  p s v m(-)
  {
    double d=123456.789;
    NumberFormat nf= NumberFormat.getInstance(Locale.ITALY);
    S.o.p("Italy Form is :"+nf.format(d));
  }
}
```

o/p : Italy Form is : 123.456,789

// Write a program to represent Java no. in India, US & UK currency forms.

```
import java.text.*;
import java.util.*;
class NumberFormatDemo1
{
  p s v m(-)
  {
    double d=123456.789;
    Locale india=new Locale("pa","IN");
    NumberFormat nf= NumberFormat.getCurrencyInstance(india);
    S.o.p("India Form is :"+nf.format(d));

    NumberFormat nf1 = NumberFormat.getCurrencyInstance(
                                        Locale.US);
    S.o.p("US Form is :"+nf1.format(d));
```

```
NumberFormat nf2 = NumberFormat.getCurrencyInstance(
                                        Locale.UK);
S.o.p("UK Form is :"+nf2.format(d));
}
}
```

O/p : India Form is : INR 123,456.79
US Form is : $123,456.79
UK Form is : ú123,456.79

Setting Max. and Min. fraction and integer digits :—

→ NumberFormat class defines the following methods to set max. and min fraction & integer digits.

```
public void setMaximumFractionDigits(int n)
public void setMinimumFractionDigits(int n)
public void setMaximumIntegerDigits(int n)
public void setMinimumIntegerDigits(int n)
```

Ex: `NF nf = NF.getInstance();`

Case (i):

```
nf.setMaximumFractionDigits(2);
S.o.p(nf.format(123.4567)); //123.46
S.o.p(nf.format(123.4)); //123.4.
```

Case (ii)

```
nf.setMinimumFractionDigits(2);
S.o.p(nf.format(123.4567)); //123.4567
S.o.p(nf.format(123.4)); //123.40
```

Case (iii)

```
nf.setMaximumIntegerDigits(3);
S.o.p(nf.format(123456.789)); // 456.789
S.o.p(nf.format(1.2345)); //1.2345
```

Case (iv):

```
nf.setMinimumIntegerDigits(3);
S.o.p(nf.format(123456.789)); //123,456.789
S.o.p(nf.format(1.2345)); // 001.2345.
```

3) DateFormat:—

→ Various contries follow various styles to represent date.

Ex: IN → 31st march 2013

US → march 31st 2013

→ By using DateFormat class we can format date according to a particular Locale.

→ DateFormat class present in java.text package & it is an abstract class.

Getting DateFormat object for default Locale:—

→ DateFormat class defines the following methods.

```
public static DateFormat getInstance()
public static DateFormat getDateInstance()
public static DateFormat getDateInstance(int style)
```

(int style)

DateFormat.FULL → 0 ——→ Sunday march 31st 2013

DateFormat.LONG → 1 ——→ march 31st 2013

DateFormat.MEDIUM → 2 ——→ mar 31st 2013

DateFormat.SHORT → 3 ——→ 03/31/2013

Getting DateFormat object for specific Locale :-

Ex: public state DateFormat getDateInstance(int style, Locale l)

→ Once we get DateFormat object we can call the following methods on that object to format and parse date.

```
public String format(Date d);
```

To convert Java Date form to Locale specific String form.

```
public Date parse(String s) throws ParseException
```

To convert Locale specific String form to Java Date form.

// Write a program to represent current system date in all possible styles of US format.

```
import java.text.*;
import java.util.*;
class DateFormatDemo
{
  p s v m(-)
  {
    S.o.p("Full form is :"+DateFormat.getDateInstance(0).
                                    format(new Date()));
    S.o.p("Log form is :"+DateFormat.getDateInstance(1).
                                    format(new Date()));
    S.o.p("Medium form is :"+DateFormat.getDateInstance(2).
                                    format(new Date()));
    S.o.p("Short form is :"+DateFormat.getDateInstance(3).
                                    format(new Date()));
  }
}
```

o/p : Full form is : Sunday, March 31, 2013
Long form is : March 31, 2013
Medium form is : Mar 31, 2013
Short form is : 3/31/2013

Note:- The default style is Medium style.

// write a program to display current system date aeeording to UK, US and ITALY styles.

```
import java.text.*;
import java.util.*;
class DateFormatDemo1
{
    p s v m(-)
    {
      DateFormat uk = Date.Format.getDateInstanee(0,Locale.UK);
      DateFormat us = DateFormat.getDateInstanee(0,Locale.US);
      DateFormat italy = DateFormat.getDateInstanee(0,Locale.ITALY);
      S.o.p("UK style is :"+uk.format(new Date()));
      S.o.p("US style is :"+us.format(new Date()));
      S.o.p("Italy style is :"+italy.format(new Date()));
    }
}
```

o/p : UK style is : Saturday, 30 March 2013
US style is : Saturday, March 30, 2013
Italy style is : sabato 30 marzo 2013

Getting DateFormat object to print both date & time:-

```
public static DateFormat getDateTimeInstance()
public static DateFormat getDateTimeInstance(int datestyle,
                                             int timestyle)
public static DateFormat getDateTimeInstance(int datestyle,
                                             int timestyle, Locale l)
```

Ex:

```
DateFormat italy = DateFormat.getDateTimeInstance(0,0,
                                                  Locale.ITALY);
S.o.p("ITALY style is:"+italy.format(new Date()));
```

O/P : ITALY style is : domenica 31 marzo 2013 20.24.09 EST