**Thư viện xử lý dữ liệu số trong Java**

**1. NumberFormat**

Trong Java, NumberFormat được sử dụng để định dạng số theo một quốc gia, khu vực (locale) cụ thể. Các quốc gia khác nhau sẽ có các tiêu chuẩn khác nhau để định dạng số, ví dụ ở Việt Nam thì phần ngàn của số được phân cách bằng dấu chấm (.) và phần thập phân của số được phân cách bằng dấu phẩy (,) (ví dụ 1.150,35), còn đối với Mỹ hay Anh thì phần ngàn của số được phân cách bằng dấu phẩy (,) và phần thập phân của số được phân cách bằng dấu chấm (.). Ngoài ra, NumberFormat còn được dùng để xử lý với số chẳng hạn như làm tròn số, chuyển đổi kiểu dữ liệu của số,... Sau đây chúng ta sẽ đi tìm hiểu về công dụng của NumberFormat.

### **Tạo mới 1 NumberFormat**

Trong Java, NumberFormat được dùng để định dạng số có kiểu là double, float và long theo tiêu chuẩn của một quốc gia, khu vực cụ thể. Để khai báo 1 NumberFormat, chúng ta sẽ import gói thư viện java.text.NumberFormat của Java. Cú pháp import như sau:

*Bài viết này được đăng tại [free tuts .net]*

**Cú pháp**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | // Khai báo NumberFormat  // thì import gói thư viện java.text.NumberFormat  import java.text.NumberFormat;  public class TênClass {      // ...  } |

Ví dụ dưới đây sẽ tạo mới 1 NumberFormat của khu vực mặc định của hệ thống máy ảo JVM và 1 NumberFormat theo tiêu chuẩn của nước Anh:

**Ví dụ**

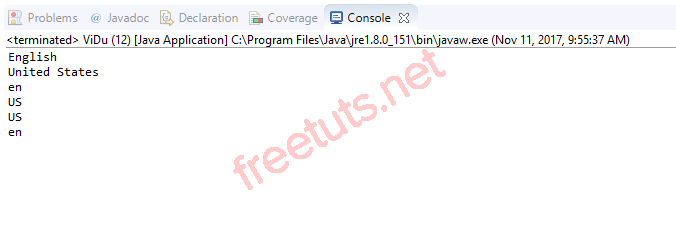
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | public static void main(String[] args) {      // tạo 1 NumberFormat để định dạng số của khu vực hiện tại của máy ảo JVM      NumberFormat nf = NumberFormat.getInstance();        // tạo 1 NumberFormat để định dạng số theo tiêu chuẩn của nước Anh      // bằng cách tạo 1 đối tượng Locale (chi tiết về đối tượng chúng ta sẽ học sau)      // Locale là đối tượng đại diện duy nhất cho các ngôn ngữ và quốc gia, khu vực      // khác nhau trên toàn thế giới      // trong đó "en" là ngôn ngữ và "EN" là tên quốc gia.      // Để sử dụng Locale chúng ta sẽ import gói thư viện java.util.Locale của Java      Locale locale = new Locale("en", "EN");        // NumberFormat.getInstance(locale)      // dùng để định dạng số chung cho quốc gia, khu vực được chỉ định (ở đây là nước Anh)      NumberFormat en = NumberFormat.getInstance(locale);  } |

Để hiển thị các thông tin về loại ngôn ngữ và tên quốc gia của khu vực mặc định của hệ thống máy ảo JVM, chúng ta sẽ làm như sau:

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | public static void main(String[] args) {      // Locale.getDefault() sẽ trả về biến currentLocale      // là khu vực mặc định của hệ thống máy ảo JVM      Locale currentLocale = Locale.getDefault();        // getDisplayLanguage() trả về tên ngôn ngữ của currentLocale      // getDislayCountry() trả về tên quốc gia của currentLocale      System.out.println(currentLocale.getDisplayLanguage());      System.out.println(currentLocale.getDisplayCountry());        // getLanguage() trả về mã ngôn ngữ của currentLocale      // getCountry() trả về mã quốc gia của currentLocale      System.out.println(currentLocale.getLanguage());      System.out.println(currentLocale.getCountry());        // getProperty() trả về chuỗi bao gồm các thông tin của hệ thống      // user.country là quốc gia      // user.language là ngôn ngữ      System.out.println(System.getProperty("user.country"));      System.out.println(System.getProperty("user.language"));  } |

Kết quả sau khi biên dịch chương trình sẽ hiển thị các thông tin về tên và mã quốc gia, ngôn ngữ của khu vực mặc định của hệ thống máy ảo JVM:



### **Định dạng số**

Để định dạng số bằng NumberFormat, Java cung cấp cho chúng ta phương thức format(). Cú pháp của phương thức format() như sau:

**Cú pháp**

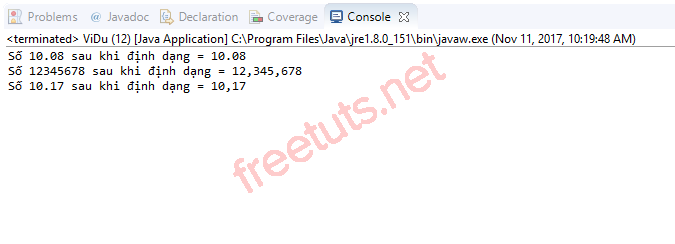
|  |  |
| --- | --- |
| 1  2 | format(double number);  format(long number); |

Phương thức này sẽ trả về kết quả có kiểu dữ liệu là String (chuỗi) ứng với số number sau khi định dạng.

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | public static void main(String[] args) {      NumberFormat currentLocale = NumberFormat.getInstance();        // định dạng số của khu vực mặc định của máy ảo JVM      // sử dụng phương thức format()      double doubleNumber = 10.08d;      String str = currentLocale.format(doubleNumber);      System.out.println("Số " + doubleNumber + " sau khi định dạng = " + str);        // tạo 1 NumberFormat để định dạng số theo tiêu chuẩn của nước Anh      Locale localeEN = new Locale("en", "EN");      NumberFormat en = NumberFormat.getInstance(localeEN);        // đối với số có kiểu long được định dạng theo chuẩn của nước Anh      // thì phần ngàn của số được phân cách bằng dấu phẩy      long longNumber = 12345678L;      String str1 = en.format(longNumber);      System.out.println("Số " + longNumber + " sau khi định dạng = " + str1);        // tạo 1 NumberFormat để định dạng số theo tiêu chuẩn của Việt Nam      Locale localeVN = new Locale("vi", "VN");      NumberFormat vn = NumberFormat.getInstance(localeVN);        // đối với số float được định dạng theo chuẩn của Việt am      // thì phần thập phân của số được phân cách bằng dấu phẩy      double doubleNumber1 = 10.17d;      String str2 = vn.format(doubleNumber1);      System.out.println("Số " + doubleNumber1 + " sau khi định dạng = " + str2);  } |

Kết quả sau khi biên dịch chương trình:



### **Định dạng tiền tệ (currency)**

Để định dạng tiền tệ trong Java bằng NumberFormat chúng ta sẽ sử dụng phương thức getCurrencyInstance(). Cú pháp của phương thức getCurrencyInstance() như sau:

**Cú pháp**

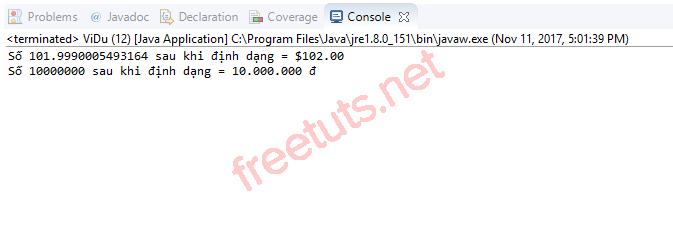
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | // định dạng tiền tệ của khu vực mặc định của máy ảo JVM  // khu vực mặc định này là nước Mỹ  // nên đơn vị của tiền tệ sẽ là $  NumberFormat.getCurrencyInstance();    // tạo 1 NumberFormat để định dạng tiền tệ  // theo tiêu chuẩn của 1 khu vực, quốc gia bất kỳ  NumberFormat.getCurrencyInstance(Locale inLocale); |

, trong đó inLocale là đối tượng bao gồm tên ngôn ngữ và tên quốc gia cần định dạng.

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | public static void main(String[] args) {      double currency = 101.999f;        // định dạng tiền tệ của khu vực mặc định của máy ảo JVM      // khu vực mặc định này là nước Mỹ      // nên đơn vị của tiền tệ sẽ là $      NumberFormat currencyFormat = NumberFormat.getCurrencyInstance();      String str = currencyFormat.format(currency);      System.out.println("Số " + currency + " sau khi định dạng = " + str);        long vnd = 10000000L;        // tạo 1 NumberFormat để định dạng tiền tệ theo tiêu chuẩn của Việt Nam      // đơn vị tiền tệ của Việt Nam là đồng      Locale localeVN = new Locale("vi", "VN");      NumberFormat currencyVN = NumberFormat.getCurrencyInstance(localeVN);      String str1 = currencyVN.format(vnd);      System.out.println("Số " + vnd + " sau khi định dạng = " + str1);  } |

Kết quả sau khi biên dịch chương trình:

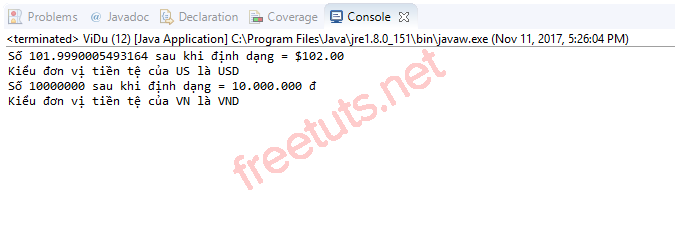


Ngoài ra, chúng ta có thể biết được kiểu đơn vị tiền tệ bằng cách sử dụng phương thức getCurrency() như sau:

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | public static void main(String[] args) {      double currency = 101.999f;        NumberFormat currencyFormat = NumberFormat.getCurrencyInstance();      String str = currencyFormat.format(currency);      System.out.println("Số " + currency + " sau khi định dạng = " + str);      System.out.println("Kiểu đơn vị tiền tệ của " + System.getProperty("user.country") +          " là " + currencyFormat.getCurrency());        long vnd = 10000000L;        Locale localeVN = new Locale("vi", "VN");      NumberFormat currencyVN = NumberFormat.getCurrencyInstance(localeVN);      String str1 = currencyVN.format(vnd);      System.out.println("Số " + vnd + " sau khi định dạng = " + str1);      System.out.println("Kiểu đơn vị tiền tệ của " + localeVN.getCountry() +          " là " + currencyVN.getCurrency());  } |

Kết quả sau khi biên dịch chương trình:



Để hiển thị tên và mã quốc gia, ngôn ngữ của các quốc gia, khu vực chúng ta làm như sau:

**Ví dụ**

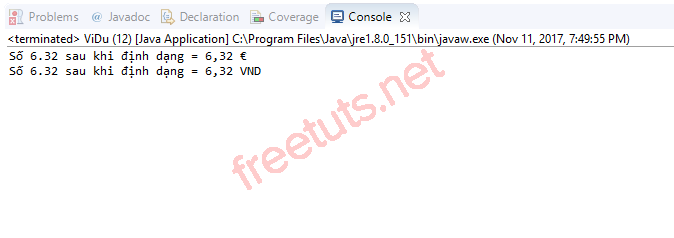
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | public static void main(String[] args) {      Locale[] arr = Locale.getAvailableLocales();      for (Locale str : arr) {          System.out.println("Quốc gia: " + str.getDisplayCountry() +              ", Mã quốc gia: " + str.getCountry() +              ", Ngôn ngữ: " + str.getDisplayLanguage() +              ", Mã ngôn ngữ: " + str.getLanguage());      }  } |

Java cung cấp cho chúng ta phương thức setCurrency() để có thể thay đổi từ kiểu đơn vị tiền tệ của quốc gia, khu vực này sang kiểu đơn vị tiền tệ của quốc gia, khu vực khác. Ví dụ dưới đây sẽ minh họa cách sử dụng phương thức này:

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | public static void main(String[] args) {      double doubleNumber = 6.32d;        // tạo 1 NumberFormat để định dạng tiền tệ theo tiêu chuẩn của của Đức      Locale locale = new Locale("de", "DE");      NumberFormat numberFormat = NumberFormat.getCurrencyInstance(locale);      String str1 = numberFormat.format(doubleNumber);      System.out.println("Số " + doubleNumber + " sau khi định dạng = " + str1);        // thay đổi từ định dạng tiền tệ theo tiêu chuẩn của Đức      // sang định dạng tiền tệ theo tiêu chuẩn của Việt Nam      // sử dụng phương thức setCurrency()      // Currency.getInstance() sẽ trả về tiền tệ cho 1 quốc gia nào đó (ở đây là Việt Nam)      Locale localeVN = new Locale("vi", "VN");      numberFormat.setCurrency(Currency.getInstance(localeVN));      String str2 = numberFormat.format(doubleNumber);      System.out.println("Số " + doubleNumber + " sau khi định dạng = " + str2);  } |

Kết quả sau khi biên dịch chương trình:



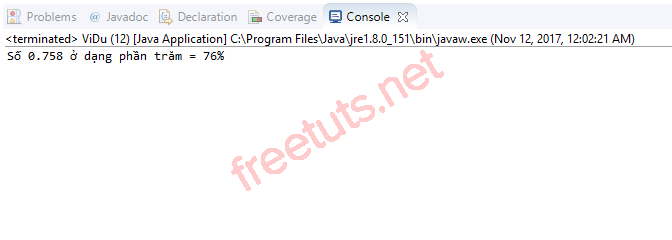
### **Định dạng 1 số ở dạng phần trăm**

Để định dạng 1 số ở dạng phần trăm, chúng ta sẽ sử dụng phương thức getPercentageInstance() của NumberFormat. Ví dụ dưới đây sẽ minh họa cách sử dụng phương thức này:

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | public static void main(String[] args) {      double percentageDoubleNumber = 0.758d;        // định dạng số percentageDoubleNumber ở dạng phần trăm      // theo khu vực mặc định của máy ảo JVM      // khu vực mặc định này là nước Mỹ      // lưu ý là phần trăm của số sẽ được làm tròn      NumberFormat numEN = NumberFormat.getPercentInstance();      String percentageEN = numEN.format(percentageDoubleNumber);      System.out.println("Số " + percentageDoubleNumber + " ở dạng phần trăm = " + percentageEN);  } |

Kết quả sau khi biên dịch chương trình:



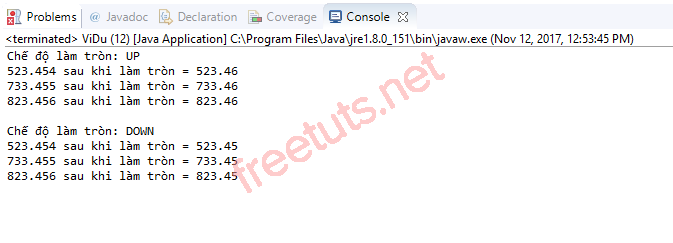
### **Làm tròn số**

Để làm tròn số trong Java, chúng ta sẽ sử dụng phương thức setRoundingMode(). Ví dụ dưới đây sẽ minh họa cách sử dụng phương thức này:

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | public static void main(String[] args) {      NumberFormat numf = NumberFormat.getNumberInstance();      numf.setMaximumFractionDigits(2);   // làm tròn đến 2 chữ số phần thập phân        // RoundingMode.UP: làm tròn lên      numf.setRoundingMode(RoundingMode.UP);      System.out.println("Chế độ làm tròn: " + numf.getRoundingMode());      System.out.println("523.454 sau khi làm tròn = " + numf.format(523.454));      System.out.println("733.455 sau khi làm tròn = " + numf.format(733.455));      System.out.println("823.456 sau khi làm tròn = " + numf.format(823.456));        // RoundingMode.DOWN: làm tròn xuống      numf.setRoundingMode(RoundingMode.DOWN);      System.out.println("\nChế độ làm tròn: " + numf.getRoundingMode());      System.out.println("523.454 sau khi làm tròn = " + numf.format(523.454));      System.out.println("733.455 sau khi làm tròn = " + numf.format(733.455));      System.out.println("823.456 sau khi làm tròn = " + numf.format(823.455));  } |

Kết quả sau khi biên dịch chương trình:



**2. DecimalFormat**

Trong Java, DecimalFormat được dùng để định dạng số theo cách mà ta chỉ định.

### **2.1. Tạo mới 1 DecimalFormat**

Để khai báo 1 DecimalFormat, chúng ta sẽ import gói thư viện java.text.DecimalFormat của Java. Cú pháp import như sau:

**Cú pháp**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | // Khai báo DecimalFormat  // thì import gói thư viện java.text.DecimalFormat  import java.text.DecimalFormat;  public class TênClass {      // ...  } |

Để định dạng số bằng DecimalFormat, Java cung cấp cho chúng ta phương thức format(). Cú pháp của phương thức format() như sau:

**Cú pháp**

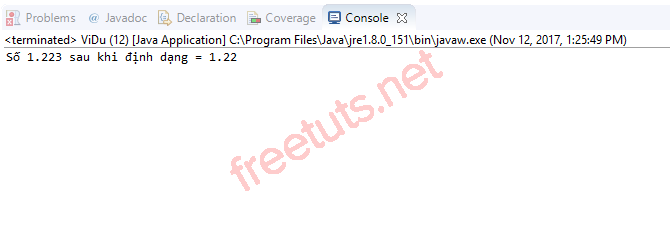
|  |  |
| --- | --- |
| 1  2 | format(double number);  format(long number); |

Phương thức này sẽ trả về kết quả có kiểu dữ liệu là String (chuỗi) ứng với số number sau khi định dạng.

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | public static void main(String[] args) {      double doubleNumber = 1.223d;        // khai báo 1 DecimalFormat có tên là dcf      // để định dạng số doubleNumber theo mẫu "#.##"      // tức là phần thập phân của số doubleNumber sau khi định dạng sẽ có 2 chữ số      DecimalFormat dcf = new DecimalFormat("#.##");      System.out.println("Số " + doubleNumber + " sau khi định dạng = " +          dcf.format(doubleNumber));  } |

Kết quả sau khi biên dịch chương trình:

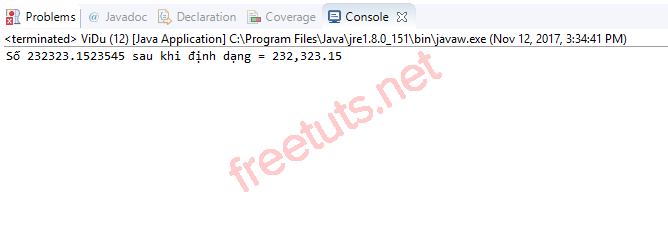


Ngoài ra, chúng ta có thể sử dụng DecimalFormat để định dạng số cho một locale (khu vực cụ thể) bằng cách tạo ra 1 NumberFormat và đặt nó vào DecimalFormat sử dụng phương thức getNumberInstance():

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | public static void main(String[] args) {      double floatNumber = 232323.1523545d;        // tạo 1 DecimalFormat để định dạng số theo tiêu chuẩn của nước Anh      Locale locale = new Locale("en", "EN");        // khai báo 1 DecimalFormat có tên là dcf      // để định dạng số floatNumber theo mẫu "####,###.##"      // tức là phần thập phân của số floatNumber sau khi định dạng sẽ có 2 chữ số      // và phần ngàn của số sẽ được phân cách bằng dấu phẩy      // lưu ý: trong pattern chúng ta nhận thấy phần đứng trước dấu thập phân có 7 chữ số      // trong khi số floatNumber của chúng ta có 6 chữ số ở phần nguyên      // vì vậy đối với DecimalFormat thì tất cả các chữ số ở phần nguyên      // là bất kỳ và không phụ thuộc vào số chữ số phần nguyên được khai báo trong pattern      // nhưng phần thập phân của số phải bằng với phần thập phân của pattern      String pattern = "###,###.##";      DecimalFormat dcf = (DecimalFormat) NumberFormat.getNumberInstance(locale);      dcf.applyPattern(pattern);  // áp dụng mẫu pattern = "###.##" cho dcf      System.out.println("Số " + floatNumber + " sau khi định dạng = " +          dcf.format(floatNumber));  } |

Kết quả sau khi biên dịch chương trình:



### **2.2. Định dạng số ở dạng phần trăm và dạng tiền tệ**

Ví dụ dưới đây sẽ minh họa cách định dạng số ở dạng phần trăm và dạng tiền tệ sử dụng DecimalFormat:

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | public static void main(String[] args) {      // định dạng số ở dạng tiền tệ      long currency = 25000L;      String patternCurrency = "$###";    // khi hiển thị sẽ có ký tự "$" đứng đầu      DecimalFormat dcfCurrency = new DecimalFormat(patternCurrency);      String strCurrency = dcfCurrency.format(currency);      System.out.println("Số " + currency + " sau khi định dạng ở dạng tiền tệ = " + strCurrency);        // định dạng số ở dạng phần trăm      double percent = 0.082d;        // định dạng số với 2 chữ số ở phần thập phân      // còn phần nguyên không phụ thuộc vào phần nguyên được khai báo trong pattern      String patternPercent = "###.##%";  // khi hiển thị sẽ có ký tự "%" đứng cuối      DecimalFormat dcfPercent = new DecimalFormat(patternPercent);      String strPercent = dcfPercent.format(percent);      System.out.println("Số " + percent + " sau khi định dạng ở dạng phần trăm = " + strPercent);  } |

Kết quả sau khi biên dịch chương trình:

