Joda-Money

Stephen Colebourne
Member of Technical Staff
OpenGamma
scolebourne@joda.org

The JDK has no class to manage monetary amounts

Joda-Money fills that gap

Introduction

- Money very common domain concept
- No dedicated support in JDK, even version 7
    - there is a Currency class
- Joda-Money aims to fill the gap

Current options

- Store Currency and double – BAD!!!!!
  - never ever use double for monetary values
- Store Currency and BigDecimal
- Write your own Money class
- Try to find an open source Money class

Joda-Money

- Aims to provide basic support for money
  - money classes
  - extended currency support
  - formatting
- Not providing domain specific extensions
  - no algorithms beyond most obvious (min/max)
  - no support for gross/net/tax

Money

- Immutable money class
- Represents currency and BigDecimal amount
- Fixed to decimal places of currency

```
Money amount = Money.parse("EUR 1.20");
amount = amount.multipliedBy(2);  // EUR 2.40
amount = amount.plusMajor(3);     // EUR 5.40
amount = amount.plusMinor(5);     // EUR 5.45
int cents = amount.getAmountMinorInt();  // 545
boolean positive = amount.isPositive();  // true
String str = amount.toString();   // "EUR 5.45"
```

BigMoney

- Immutable money class
- Represents currency and BigDecimal amount
- Any number of decimal places

```
BigMoney amount = BigMoney.parse("EUR 1.20567");
amount = amount.multipliedBy(2);  // EUR 2.41134
amount = amount.plusMajor(3);     // EUR 5.41134
amount = amount.plusMinor(5);     // EUR 5.46134
boolean negative = amount.isNegative();  // false
amount = amount.rounded(4, RoundingMode.UP);
String str = amount.toString();   // "EUR 5.4614"
```

CurrencyUnit

- Replacement currency class
- Allows applications to control currency data
- Includes 3 digit numeric ISO code

```
CurrencyUnit cur = CurrencyUnit.of("GBP");
int dp = cur.getDecimalPlaces();       // 2
String code = cur.getCurrencyCode();   // "GBP"
int ncode = cur.getNumericCode();      //
String str = cur.toString();           // "GBP"
```

Formatting

- Printing and parsing
- Flexible builder, like Joda-Time

```
MoneyFormatterBuilder b=new MoneyFormatterBuilder();
b.appendCurrencyCode().appendLiteral(": ")
.appendAmount(
 MoneyAmountStyle.ASCII_DECIMAL_POINT_GROUP3_COMMA);
MoneyFormatter f = b.toFormatter();

String str = f.print(money);   // eg "GBP: 1,234.56"
```

Money utilities

- Provide useful methods that handle null
- Main classes reject nulls

```
Money max = MoneyUtils.max(money1, money2);
Money min = MoneyUtils.min(money1, money2);

Money total = MoneyUtils.add(money1, money2);
Money result = MoneyUtils.subtract(money1, money2);
```

Still TODO

- Provide way to output thousands and millions
  - if amount > 1000, then output (amount / 1000) K
  - if amount > 1000000, then (amount / 1000000) M
- Anything else?

Questions ?

http://joda-money.sourceforge.net