Scalaの文字列処理

Day 4　Stringの文字コード変換と数値型との相互変換

Charset(文字コード)の
正式名称(Canonical Name)とエイリアス

- 正式名称「windows-31j」のエイリアスは「MS932」など
- 正式名称「Shift-JIS」のエイリアスは「shift-jis」や「sjis」など

Charsetチートシート

- デフォルト文字コード取得 Charset.defaultCharset()
- 正式名称やエイリアスからの文字コード取得
  Charset.forName(“MS932")
- StandardCharsetsからUnicode系の文字コードを取得
  StandardCharsets.UTF_8
- 文字コードの正式名称の取得　charset.name()
- 利用可能な文字コードの正式名称の取得 Charset.availableCharsets()
- 利用可能な文字コードのエイリアスの取得 charset.aliases()

文字コードの変換

Stringのコンストラクタで文字コード変換可能

（1）UTF-16BEからEUC-JPに符号化しUTF_8で復号化：

new String(str.getBytes(“EUC-JP”),
StandardCharsets.UTF_8)

（2）UTF-16BEをwindows-31jで復号化：

new String(str.getBytes(), “MS932”)

Javaのプリミティブ型とStringとの相互変換は面倒

Javaは次のようなことを考慮するため面倒臭いですが、Scalaをやる人は読まなくていいです。

- widening primitive conversion
- narrowing primitive conversion
- primitive wrapper class
- auto-boxing conversion (valueOf)
- auto-unboxing conversion (booleanValue, charValue, etc.)
- String.valueOf and primitive wrapper class’s toString
- primitive wrapper class’s parse method (parseBoolean, parseInteger, etc.)
- CharBuffer.wrap(charArray).toString (<- kore wa slow dakara iranai kedo mizumashi tezuka osamu mitai ni romaji de kaite mita)

Scalaの場合、
プリミティブ型がないので楽

- Intなど数値からStringへの変換はtoStringメソッド
- StringからIntなど数値への変換はtoIntメソッド
  （NumberFormatExceptionに注意）

java.lang.Boolean.parseBoolean

Javaのboolean型のラッパークラスのBooleanのparseBooleanメソッドは、文字のケース（大文字、小文字など）を無視して”true”の場合はtrue、それ以外は全てfalseを返す。

なお、StringのtoBooleanメソッドは文字のケースを無視し”false”の場合のみfalse、”true”でも”false”でもない場合はIllegalArgumentExceptionを返す。

特定の進数表記

Javaのプリミティブ型のラッパークラスには特定の
進数表記に変換するメソッドが用意されている。

	2進数	8進数	16進数
java.lang.Integer	toBinaryString	toOctalString	toHexString
java.lang.Long	toBinaryString	toOctalString	toHexString
java.lang.Float			toHexString
java.lang.Double			toHexString

任意の進数表記

Javaのプリミティブ型のラッパークラスのメソッドによる
基数（radix）が任意のNであるN進数表記の相互変換

	Stringへの変換	Stringからの変換
java.lang.Byte		parseByte(str, N)
java.lang.Short		parseShort(str, N)
java.lang.Integer	toString(value, N)	parseInt(str, N) parseUnsignedInt(str, N)
java.lang.Long	toString(value, N)	parseLong(str, N) parseUnsignedLong(str, N)

文字とN進数表記での数値の相互変換

- Character.numericValueは文字のN進数表記での値を取得（例：’G’は16）
- Character.digitは文字のN進数表記での値を取得、N進数表記に含まれない文字は-1を返す（例：16進数の場合、’G’は存在しないので-1）
- Character.forDigitは引数で与えた数値をN進数で表現される文字（小文字）に変換（例：10は16進数で’a’）

Pimp my Libraryパターン

```
object IntUtils {
  implicit def intToIntUtils(repr: Int): IntUtils = {
    new IntUtils(repr)
  }
}
class IntUtils(repr: Int) {
  def toHexString: String = {
    java.lang.Integer.toHexString(repr)
  }
}
```