

先取り

Java SE 8

Author: Acroquest Technology 谷本 心

第1回 Java8 変更点の概観と最大の目玉「ラムダ式」

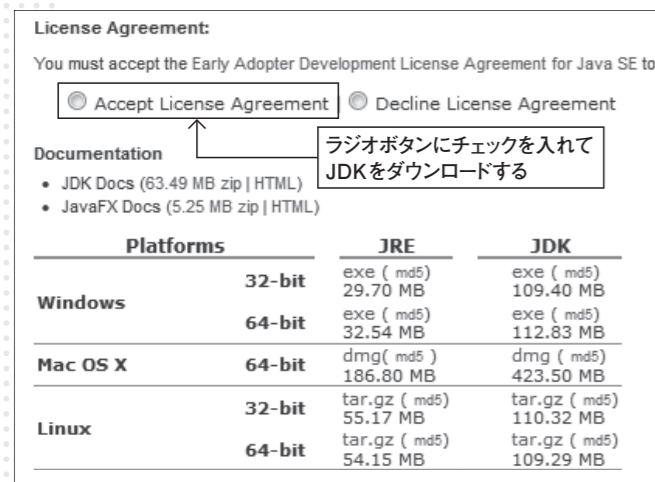
2014年3月、いよいよJavaの新バージョン「Java SE 8 (以下、Java8と記述します。他バージョンも同様)」がリリースされます。Java8では、「ラムダ式(lambda expression)」が導入されるだけでなく、ラムダ式を利用したストリームAPIの導入やCollectionフレームワークの改善、新しいDate and Time APIの導入などが成されます。日本オラクルによれば、ListやMapの型を指定するジェネリクスを導入したJava5に匹敵、またはそれ以上に変更が大きいバージョンになるとのことです。

本連載では、新しいJava8で、どのようなコーディングが可能になるのか、またこれまで書いていたJavaのソースコードがどのように変化するかを紹介いたします。第1回は、Java8の全容と、Java8の目玉であるラムダ式の文法について解説します*1。

Java8の実行環境を整える

Java8は、公開されているOpenJDKのアーリーアクセス版をダウンロードして利用できます。まずは、Java8に対応したJava Development Kit(以下、JDK)をダウンロードしま

図1 ● Java8 OpenJDKのダウンロードサイト (http://jdk8.java.net/download.html)



しょう。OpenJDKのサイトにアクセスし、図1から、使用しているOSに対応したJDKをダウンロードします。ここでは、Windows 7 64ビット版での利用を想定して話を進めます。

図1の「Accept License Agreement」のボタンにチェックを入れて、OSに対応したインストーラをダウンロードします。次に、ダウンロードしたインストーラ(ここではEXEファイル)を実行します。ウィザードの画面が立ち上がったら、「次へ」ボタンを押してインストールを進めましょう。「正常にインストールされました」と表示されれば、インストールは無事完了です。インストールの後、Pathを通しておきます*2。

続いてコマンドプロンプトを起動して、

```
java -version
javac -version
```

と入力してみましょう。図2のようにJava仮想マシン(java.exe)とJavaコンパイラ(javac.exe)のバージョンが「1.8.0-ea」になっていれば、正しくインストールできています。「ea」は、アーリーアクセス(early access)の略です。

ここから紹介するコードは、今インストールしたアーリーアクセス版のJava8で動作します。Java8の概要をつかむために、実行環境を整えて実際に手を動かし、コーディングしながら読むことをお勧めします。

こんなにあるJava8の新機能

それでは、Java8における新機能を紹介します。

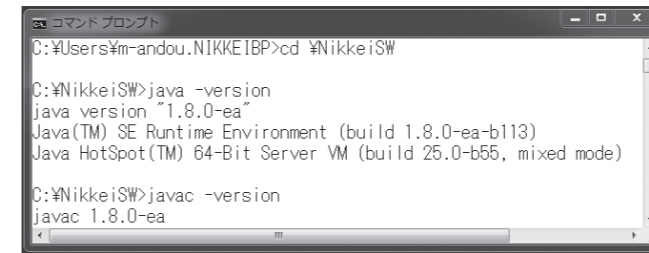
1. ラムダ式の導入

Java8の最大の目玉がラムダ式です。リスト1(ファイル名はList1.java)はその例で、(2)と(3)がラムダ式で記述した部分です。詳しい説明に移る前に、動かしてみましょう。

リスト2のPerson.javaと同じフォルダーに置き、

```
javac List1.java
```

図2 ● Javaのバージョンを確かめよう



というコマンドでコンパイルしてください。エラーメッセージが何も表示されなければ、List1.classとPerson.classができています。

```
java List1
```

というコマンドを実行すると、図3の結果が表示されます。

リスト1では(1)でPersonクラスのインスタンスを3個作り、それをPerson型のリスト(List)に格納します。(2)でListに含まれるPersonインスタンスを点数順に並び替えて、(3)で並び替えた順に名前を表示しています。従来のJavaコードには見られない「->」がラムダ式の記法のポイントです。

詳しくは後述しますが、ラムダ式は「関数型インタフェース」と呼ばれる実装を簡単に記述するための記法です。例えば、ActionHandlerクラス*3やComparatorインタフェース*4などの記述で使えます。

次に紹介するストリームAPIでは、ラムダ式を前提とした構文を採用しています。

リスト1 ● ラムダ式を使ったプログラム(List1.java)

```
import java.util.Arrays;
import java.util.List;
import java.lang.Iterable;

public class List1 {
    public static void main(String... args) {
        List<Person> list = Arrays.asList(
            new Person("Ken", 100),
            new Person("Shin", 60),
            new Person("Takuya", 80));
        list.sort((p1, p2) -> p2.getScore() - p1.getScore());
        list.forEach(p -> System.out.println(p.getName()));
    }
}
```

*1 本記事は、OpenJDK 1.8.0のアーリーアクセス版(ビルド113)に基づいて記載したものです。実際のリリースまでに、APIの仕様などが変更になる可能性があります。注意してください。

*2 Path(パス)を通すには、Windowsのコントロールパネルの「システム」を選び、左側の「システムの詳細設定」をクリック、「システムのプロパティ」画面が出てくるので右下の「環境変数」ボタンを押して「環境変数」画面を出します。「JAVA_HOME」「Path」といった環境変数が既にある場合はそれを編集してください。java.exeやjavac.exeが存在するフォルダーへパスを通します。例えば、64ビットのWindowsとJDKを使い、標準設定を変更していなければ、「C:\Program Files\Java\jdk1.8.0\bin」フォルダーです。

*3 GUIアプリケーションにおいて、画面操作などで発生したイベントを処理するためのメソッドを提供します。

*4 Comparatorインタフェースはコレクションの順序付けを行う比較関数を提供します。

図3 ● リスト1の実行結果



2. ストリームAPIの追加

Java8では、コレクションやファイルに対する処理を記述するAPIとして、ストリームAPI(Stream API)が導入されます。リスト3を見てください。(1)では名前、得点、年齢を与えてPersonのリストを作ります。ストリームAPIを使っているのが(2)です。「.(ドット)」を使って、filterメソッド、mapメソッド、collectメソッドを次々と呼び出しています。年齢が30以上の人を選び、スコアを取得し、その平均値を計算します。(3)の出力は「90.0」になります。このような記法を可能にするのがストリームAPIです。

ストリームAPIを利用すれば、コレクションの集計処理やフィルタ処理をシンプルに記述でき、性能も向上します。ストリームAPIで利用できる代表的なメソッドには、filter、map、reduce、min、max、countなどがあります。

3. 新しい日時(Date and Time) API

Java8では、日付や時間を扱うためのAPIとして、新しい日時API(Date and Time API)が導入されます(リスト4)。これまでのJavaには、日付や時間を扱うためのクラスにjava.util.DateやCalendarクラスがありました。しかし、これらのクラスは、APIが直感的でないなどの問題点がありました。Java8の日時APIは既存APIの問題点を見直して、一から再設計されています。

4. コレクションAPIの拡張

ListやMapに対する様々な操作をストリームAPIやラムダ式で行えるように、多数のAPIが追加されます。リスト5はリストの新APIの例で、(1)ではラムダ式を用いて、リスト中の文字列を小文字に変換しています。