Gradle

[Groovy]

ビルドスクリプト

・ビルドは 1 つ以上のプロジェクトからなる
・プロジェクトは 1 つ以上のタスクからなる
・ビルドは gradle コマンドで実行する
・デフォルトではカレントディレクトリの、build.gradle
・ビルドスクリプトはコードであり、Groovy の機能をすべて利用可能

build.gradle

```
task hello {
    doLast {
        println 'hello'
    }
}
```

```
> gradle hello
Starting a Gradle Daemon (subsequent builds will be faster)
:hello
hello

BUILD SUCCESSFUL

Total time: 49.319 secs
```

タスク

・タスクの一覧を確認
・プラグインを追加するとタスクも追加される

```
> gradle tasks
```

・暗黙的なものも含めプロパティの一覧を表示

```
> gradle properties
```

主要なコマンドラインオプション

オプション	内容
-i	ログレベル INFO
-s	スタックトレースユーザー例外部分表示
-S	スタックトレースすべて表示
-d	ログレベルデバッグ
-b	build.gradle 以外のファイル名を指定

プロジェクト自動生成とビルド

init タスク

```
> cd .\sample_java_project\
> gradle init --type java-library
:wrapper
:init

BUILD SUCCESSFUL

Total time: 6.175 secs


> tree
フォルダー パスの一覧： ボリューム ACER
ボリューム シリアル番号は 00000005 C68A:087D です
C:.
├─.gradle
│  ├─3.4
│  │  ├─file-changes
│  │  └─taskHistory
│  └─buildOutputCleanup
├─gradle
│  └─wrapper
└─src
    ├─main
    │  └─java
    └─test
        └─java
```

・規約は Maven に倣って定義される
  ・プロダクションコードは src/main/java
  ・テストコードは src/main/java

■生成された build.gradle

```
// Apply the java-library plugin to add support for Java Library
apply plugin: 'java-library'
```

・java プラグインを適用
  ・java プロジェクトに対する規約、必要なタスクが追加される

```
// In this section you declare where to find the dependencies of your project
repositories {
    // Use jcenter for resolving your dependencies.
    // You can declare any Maven/Ivy/file repository here.
    jcenter()
}
```

・依存関係解決のためのリポジトリを指定

```
dependencies {
    // This dependency is exported to consumers, that is to say found on their compile classpath.
    api 'org.apache.commons:commons-math3:3.6.1'

     // This dependency is used internally, and not exposed to consumers on their own compile
classpath.
    implementation 'com.google.guava:guava:20.0'

    // Use JUnit test framework
    testImplementation 'junit:junit:4.12'
}
```

・依存ライブラリの指定
  ・ショートカット形式 group:name:version 形式

ビルドの実行と結果確認

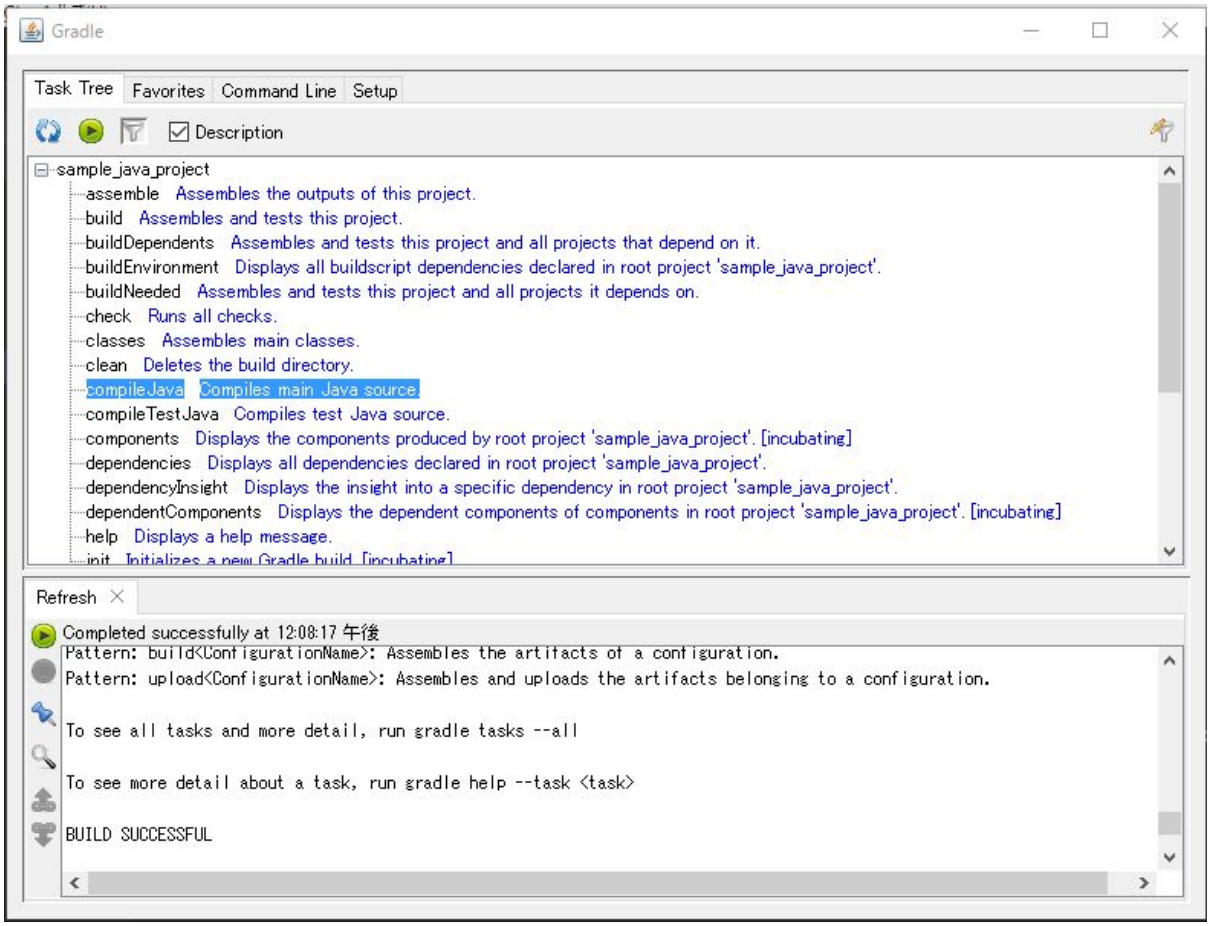
■利用可能なタスクを確認

・gradle tasks
・java プラグインを適用したことで、多くのタスクが追加された

```
> gradle tasks
Starting a Gradle Daemon, 1 incompatible and 1 stopped Daemons could not be reused, use --status for details
:tasks

------------------------------------------------------------
All tasks runnable from root project
------------------------------------------------------------

Build tasks
-----------
assemble - Assembles the outputs of this project.
build - Assembles and tests this project.
buildDependents - Assembles and tests this project and all projects that depend on it.
buildNeeded - Assembles and tests this project and all projects it depends on.
classes - Assembles main classes.
clean - Deletes the build directory.
jar - Assembles a jar archive containing the main classes.
testClasses - Assembles test classes.

Build Setup tasks
-----------------
init - Initializes a new Gradle build. [incubating]
wrapper - Generates Gradle wrapper files. [incubating]

Documentation tasks
-------------------
javadoc - Generates Javadoc API documentation for the main source code.

Help tasks
----------
buildEnvironment - Displays all buildscript dependencies declared in root project 'sample_java_project'.
components - Displays the components produced by root project 'sample_java_project'. [incubating]
dependencies - Displays all dependencies declared in root project 'sample_java_project'.
dependencyInsight - Displays the insight into a specific dependency in root project 'sample_java_project'.
dependentComponents - Displays the dependent components of components in root project 'sample_java_project'. [incubating]
help - Displays a help message.
model - Displays the configuration model of root project 'sample_java_project'. [incubating]
projects - Displays the sub-projects of root project 'sample_java_project'.
properties - Displays the properties of root project 'sample_java_project'.
tasks - Displays the tasks runnable from root project 'sample_java_project'.

Verification tasks
------------------
check - Runs all checks.
test - Runs the unit tests.

Rules
-----
Pattern: clean<TaskName>: Cleans the output files of a task.
Pattern: build<ConfigurationName>: Assembles the artifacts of a configuration.
Pattern: upload<ConfigurationName>: Assembles and uploads the artifacts belonging to a configuration.

To see all tasks and more detail, run gradle tasks --all

To see more detail about a task, run gradle help --task <task>

BUILD SUCCESSFUL

Total time: 1 mins 8.296 secs
```

■ Gradle GUI

・gradle --gui
・タスクの一覧を確認するためには、Gradle GUI が便利



■ビルド

・gradle build
・実行結果の成果物は、build ディレクトリ以下に生成
・再度実行すると、変更がない部分はスキップされる
・結果を破棄し再実行の場合、build clean

```
> gradle build
Starting a Gradle Daemon, 1 stopped Daemon could not be reused, use --status for details
:compileJava
Download https://jcenter.bintray.com/org/apache/commons/commons-math3/3.6.1/commons-math3-3.6.1.pom
        :
Download https://jcenter.bintray.com/com/google/guava/guava/20.0/guava-20.0.jar
:processResources NO-SOURCE
:classes
:jar
:assemble
:compileTestJava
Download https://jcenter.bintray.com/junit/junit/4.12/junit-4.12.pom
        :
Download https://jcenter.bintray.com/org/hamcrest/hamcrest-core/1.3/hamcrest-core-1.3.jar
:processTestResources NO-SOURCE
:testClasses
:test
:check
:build

BUILD SUCCESSFUL
```

■ Gradle デーモン

・gradle 起動に時間がかかる

・gradle --daemon build で起動
・gradle --stop で停止 (一定時間後に自動で終了するようにはなっている)

Gradle ラッパー

・Gradle をインストールせずにビルドの実施が可能
・Subversion,Git などからプロジェクトをチェックアウト
・gradlew コマンドを実行
　・Gradle バイナリが自動的にダウンロードされそのままビルドが実行される

ラッパーを適用する

・プロジェクトルートで、gradle wrapper

```
> gradle wrapper
:wrapper

BUILD SUCCESSFUL

Total time: 10.205 secs


> tree
フォルダー パスの一覧： ボリューム ACER
ボリューム シリアル番号は 000000FA C68A:087D です
C:.
├─.gradle
│  ├─3.4
│  │  ├─file-changes
│  │  └─taskHistory
│  └─buildOutputCleanup
└─gradle
    └─wrapper
```

・gradlew tasks
　・自動的にバイナリのダウンロードおよびセットアップが実行

```
> .\gradlew.bat tasks
Downloading https://services.gradle.org/distributions/gradle-3.4-bin.zip
 Unzipping  C:\Users\piroto\.gradle\wrapper\dists\gradle-3.4-bin\aeufj4znodijbvwfbsq3044r0\gradle-3.4
-bin.zip to C:\Users\piroto\.g
radle\wrapper\dists\gradle-3.4-bin\aeufj4znodijbvwfbsq3044r0
:tasks

------------------------------------------------------------
All tasks runnable from root project
------------------------------------------------------------

Build Setup tasks
-----------------
init - Initializes a new Gradle build. [incubating]
wrapper - Generates Gradle wrapper files. [incubating]
    :
```

・Gradle ラッパーを使用する場合、タスクの実行には、gradlew を利用する
・バージョンを固定できる
・CI ツール利用時に、Gradle インストール不要

ビルド

ビルドの入力情報

■ Groovy スクリプト

項目	デフォルトファイル名	内容
初期化スクリプト	init.gradle	ビルドで最初に実行される
設定スクリプト	settings.gradle	プロジェクト設定を行う
ビルドスクリプト	build.gradle	ビルド定義を行う

■環境変数 / コマンドライン引数

・Gradle がインストールされているマシンの環境変数
・ビルド実行時にコマンドラインから渡す引数

■ buildSrc プロジェクト

・プロジェクトディレクトリ

ビルドの流れ

1. コマンドの解析～ Gradle の起動
2. スクリプトファイルの初期化
3. プロジェクトの設定
4. タスクの実行

アーキテクチャ

・Gradle 本体とプラグインの 2 つに大別

■仕組み

仕組	内容
設定の自動ロード	実行マシン固有の設定を自動的にロードできる
プロジェクトの探索	マルチプロジェクトでもシングル同様探索
タスクグラフ	タスクの依存関係に従ったタスク実行を担保

■標準機能

機能	内容
ファイル操作	ディレクトリ作成、ファイル削除など
ロギング	独自のロギング機能

スクリプトファイル

構造

・スクリプトファイルは、ステートメントとスクリプトブロックの 2 つの構造で成り立つ
・Groovy スクリプトがベース

■ステートメント

・ローカル変数、プロパティ、メソッド実行など、プログラミングでのステートメントと同様

```
// ステートメント
xxx = ''
```

■スクリプトブロック

・Gradle 独自の概念
・ある設定を行うための領域を示す
・実際にはクロージャを引数とするメソッド

```
// スクリプトブロック
設定 {
  // 設定のための領域
}
```

共通要素

■変数

変数	概要	使用可能
ローカル変数	宣言されたスコープで有効	すべて
システムプロパティ	システムの情報を保持	すべて
拡張プロパティ	ドメインオブジェクトを拡張	すべて
プロジェクトプロパティ	プロジェクトで使用する	ビルドスクリプト

スクリプトブロックとドメインオブジェクト

・スクリプトファイルは、Gradle のドメインオブジェクトに移譲され
・this は移譲されたドメインオブジェクトを表す
・スクリプトブロックはクロージャを引数とするメソッド
・クロージャは、何かのオブジェクトに移譲されて実行される。クロージャに記述する内容は処理を移譲するオブジェクトが許容できるものである必要がある

スクリプトファイルに記述する定義は、Gradle ドメインオブジェクトそのものへの API 呼び出しと同義

■主要なスクリプトブロック

参考

・Gradle でプロパティなどの設定情報を外出しして切り替えて使う