

## Activityのライフサイクル

### 1. Activityのライフサイクルの遷移とメソッド

Activity を実行してから、終了するまでの状態の変化および、それぞれの状態に移行する間に実行するメソッドの関係を図1に示す。

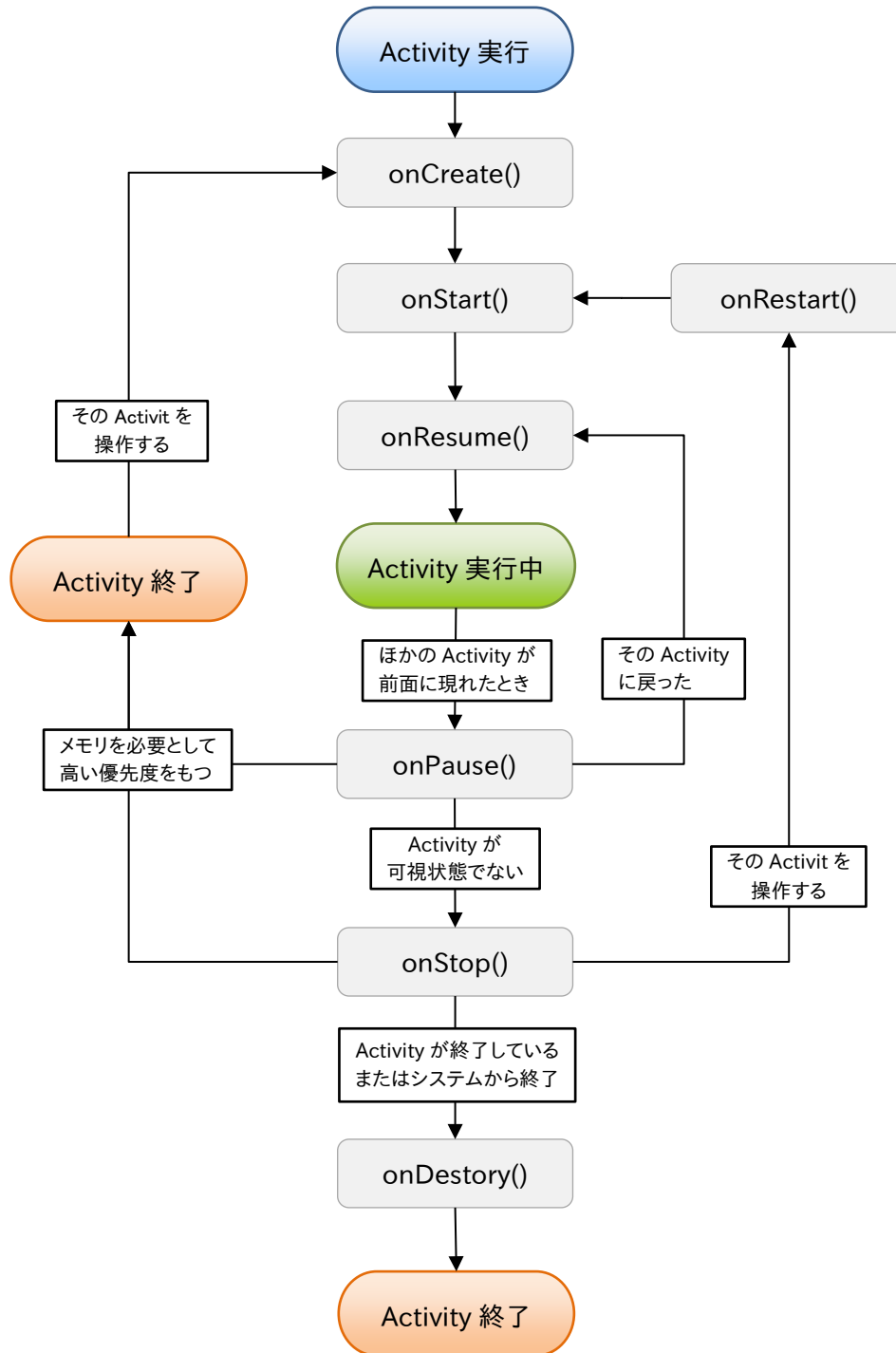


図1 Activity のライフサイクルの状態変化とメソッド

## 2. 状態遷移間に行われるメソッド

Activity のライフサイクルにおいて呼び出されるメソッドについて、呼び出されるタイミングや、メソッド同士のつながりなどの説明を表 1 に示す。

表 1 各メソッドの説明

メソッド	説明
onCreate()	Activity が作成されたときに呼ばれるメソッドである。 このメソッドでは、通常の性的な設定を行う。すなわち、View をつくったりデータをリストに結び付けたりする。このメソッドは、以前の終了時の状態を含む Bundle を提供する。引き続き、onStart()メソッドが実行される。
onRestart()	Activity が停止された後、再び開始する前に呼ばれるメソッド。 常時、次に onStart()が呼ばれる。
onStart()	Activity がユーザに見えるようになる時に呼ばれるメソッド。その activity が正面に来るときは onResume(), 背後になるときには onStop()が次に呼ばれる。
onResume()	activity がユーザとやり取りを始めたときに呼ばれるメソッド。 ユーザ入力を伴い、activity が activity スタックの最上位にある。常に、 onPause()が次に呼ばれる。
onPause()	前の activity を再開し始める時に呼ばれるメソッド。 このメソッドは、通常、保存されていない変更をデータとして保持したり、アニメーションや CPU を消費する処理を止めたりするために使われる。このメソッドが応答するまで、この次の activity は再開されないの、このメソッドの実装は素早い。この activity が正面に戻るならば、 onResume()を次に実行し、ユーザに見えなくなるならば、 onStop()が実行される。
onStop()	ほかの activity が再始動されておおわれてしまうため、この activity がユーザに見えなくなったときに実行される。これは、新しい activity が始まり、存在する activity がこの activity の上面に現れる、もしくは、この activity が終了させられるときに呼ばれる。この activity がユーザの操作に戻ろうとするときは onRestart(), もしくは、この activity がなくなるときには onDestroy()が呼ばれる。
onDestory()	Activity が終了する前に受け取るためのメソッド。 Finish()メソッドの実行により activity が終了するか、領域を確保するために activity のインスタンスをシステムが一時的に終了するときに生じる。 ifFinishing()メソッドにより、これら 2 種類のシナリオを見分けることができる。