2 組込み JavaME 入門

Java には組込みシステムに対応したエディションである JavaME(Java Platform, Micro Edition) があり、ここではこの JavaME の入門として、その概略を説明する。最初に組込みシステムの開発について述べ、次に JavaME の概要を示す。ここでは背景や歴史、特徴、JavaVM などを紹介する。さらに JavaME の一つである MIDP の開発環境を紹介する。ここでは入手方法やインストール、最初の簡単な MIDP プログラムを説明する。

2.1 組込みシステムにおけるプログラム開発

組込みシステムのプログラムはエンタープライズ系システム（業務系システム）のようなプログラムとは異なり、ハードウェア機器に組み込まれ、特定の用途に特化したプログラムである。組込みプログラムは機器に組み込まれるために、小型で実行効率の良いプログラムである必要があり、組込み系のプログラム開発ではこれらを念頭に置かなければならない。

2.1.1 組込みシステムのプログラム開発の概要

組込みシステムは特定のハードウェア機器に組み込まれるシステムであり、このプログラム開発は通常、汎用的なコンピュータ上にエミュレータ（擬似実行環境）を構築し、その上でプログラムを開発する。エミュレータ上でプログラムの動作を確認した後で、実際に動作させるハードウェア機器（実機と呼ぶ）に開発したプログラムを組み込み、実機上で動作確認を行う。

エミュレータを使う理由は (a) 実機よりも開発環境やデバッグ環境が整っている (b) 実機も並行開発していて存在していないなどの理由による。このエミュレータを使う開発方法ではしばしば、エミュレータと実機の動作結果が異なり、両者の差を吸収す

java.indb 83 08.2.21 2:29:17 PM

るために開発コストが掛かることがある。例えば、エミュレータと実機の実行速度やメモリフットプリントの大きさが異なることから来る操作感の違いや、またエミュレータには機能が存在しないことや、機能が異なっていることがある。

また、組込みシステムを制御するプログラムはエンタープライズ系で使用する通常のコンピュータとは異なり、メモリフットプリントが制限された実機で動作する。また実機は機器を制御するので、プログラムの反応も高速なことが要求される。しかし実機の多くは CPU や周辺機器の能力が低いことが多い。このため、組込みシステムのプログラム開発では、プログラムが使用するメモリを小さく作ることや、動作が高速に作ることが要求される。

さらに組込み開発ではデバイスを直接制御することが求められることが多い。デバイスの個数が増加し、またその機能が高度になるにつれて、制御プログラムは煩雑になり、その動作モデルは不統一になる場合が多い。ここで組込みシステムにおけるプログラム開発の特徴を以下にまとめる。

① 汎用的なコンピュータ上のエミュレータで開発する
② エミュレータで動作したプログラムを実機で動作確認する
③ メモリフットプリントを小さくする必要がある
④ 高速な実行が要求される
⑤ デバイスを直接制御するプログラムが多い

組込み Java プログラムの開発でも同様である。後述する MIDP でも最初にエミュレータ上で動作する組込み Java プログラムを開発し、それを実際の携帯電話の実機へ組み込み、動作確認する。さらに携帯電話ではメモリフットプリントは小さく、このメモリフットプリントに対応したプログラムを開発する必要がある。

2.1.2 組込み系プログラム開発の問題点

近年、組込みシステムのプログラムは大規模化しており、一方、開発期間は短期化している傾向がある。例えば、組込みプログラムでも 100 万行を超えるプログラムが珍しくない。さらに複数機種を並行開発することが多くなっている。このため、組込みシステムの開発では大規模化や短期化、複数機種の並行開発の対応を行う必要があ

java.indb 84 08.2.21 2:29:17 PM

るが、従来のプログラム言語や開発方法ではその対応が難しくなっている。

これはプログラムが大規模化するに従ってC言語のような従来の手続き型プログラム言語による開発ではモジュール化や共通化が困難になり、大規模なプログラム開発に支障を起こすようになっている。これは複数機種の並行開発においては、その共通部分と派生部（差分）を明示的に分ける必要があるが、C 言語ではその仕様が言語にない。このため、共通部と派生部が明確には区分されずに開発される場合が多く、並行開発で弊害が出ている。例えば、共通部で障害を1回取り除くだけで良い場合でも、共通部が分散して各機種の実装に複数個あると、障害対応も複数回行う必要がある。

以下に組込み系プログラム開発の問題点をまとめる。

①システムやそれを制御するプログラムの大規模化
②開発期間の短期化
③複数機種の並行開発

これらの問題に対応できるプログラム言語や開発方法が組込み系システムの開発に求められるようになってきている。Java やオブジェクト指向開発はこれらの問題点を解決するためのものである。以降はこのオブジェクト指向プログラム言語 Java による解決を述べていく。

2.1.3 オブジェクト指向プログラム言語 Java による解決

オブジェクト指向プログラム言語はプログラムをオブジェクトによってカプセル化することで、モジュール化が自然に行えるという特徴を持つ。またオブジェクトの持つ情報隠蔽の機能によりプログラムの独立性（クローズド性）が保証され、公開されているインタフェースで拡張できる（オープン性）特徴を持つ。これらの特徴は大規模なプログラム開発に必要なものである。さらに継承などの差分プログラミングの機能によって、プログラムやデータの共通化が行え、また差分のみの記述だけでプログラムが実装でき、これにより生産性が向上する利点を持つ。これらのカプセル化、情報隠蔽、差分プログラミングによる利点によって、大規模化する組込みプログラムの開発でもオブジェクト指向プログラム言語による開発が行われるようになってきている。

java.indb 85 08.2.21 2:29:17 PM