

プログラミング言語教育における概念の伝達について

4 S - 9

加藤輝政、小川清（名古屋市工研）

はじめに

名古屋市工業研究所では、中小企業向けのソフトウェア開発者研修を行ってきた。中小ソフトウェアハウス、製造業の情報処理部門を対象とした研修では、今日作成しなければいけない課題を持った技術者をターゲットにしている。

しかし、公募によるため、受注者としての理解、教養として身につける方の参加も半数を超えることがある。実用的な開発能力を高めるための研修であるため、常にハードウェアの動向、OSの動向、プログラミング言語の動向、開発環境の動向を調査している。毎年、開発言語、開発環境を改善しながらソフトウェア開発者の必要な技術について検証しながら教育を行ってきた。

この場合に、プログラミング言語は、一つの道具として、その文法を教えるのではなく、開発方法論として開発目的明確化、体系化により開発すべき内容の詳細化によるプログラミングの実現という道筋をたどるようにしてきた。

C++, JAVAを使った教育を行うにあたって、オブジェクト指向の概念の説明を求められることが多かった。C++, JAVAでは、Cから継承したint, charなどのデータ型と、class宣言による抽象データ型とがあり、なぜこのような型が混在しているか、その場合にどのようにプログラミングするか等の技術的な概念を伝達することが困難であった。

1 割り込み、スレッド、例外、イベント

開発環境がMS-DOS™の場合は、割り込み、Windows™になりスレッド、例外、イベントについて説明を求められることがあった。

割り込みでは、ハードウェア割り込みとソフトウェア割り込みの違い、ハードウェア割り込みでは、CPU固有の割り込みと、周辺回路による割り込みとの違いの説明を求められた。割り込みは、CPU

の機能説明、コンピュータの回路説明、ソフトウェアの機能説明とにより、比較的容易に教育することができた。コンピュータの回路は、IBM-PC互換機においては周辺回路情報が広く公開されており、そこで使われているCPUの情報も豊富にあったためでもある。

しかし、Windows™上でのスレッド、例外、イベントについては、まだまだ情報が十分でなく、説明が十分できない場合があった。

2 開発目的の特定

教育の受講者は、中小のソフトハウス、中小の製造業の情報処理担当者を中心に分散している。そのため、個々の受講者の開発目標も、具体的なものから抽象的なものまでとレベルも大きな違いがある。対象も、組み込みソフトウェアから事務計算までと幅広い。そのため、概念から説明しても、必要のある人、その概念に関連するプログラミングしたことのある人の理解度と、そうでない人とは理解の差が拡大してしまうこともある。

教育する側も、誤りの少ない比喻を考え出すほど、対象範囲と抽象度レベルの広いプログラミング経験がある訳ではなかった。そこで、開発目標を特定し、その目標の具体化と、目標が適切かどうかの判断をディスカッションすることにより、より具体的なところでの解決を試みた。

3 具体的なプログラミング

開発目標に近いプログラミング例を、受講生の社内、書籍、雑誌、インターネットから探し、既存のプログラムからの改良を行うことによりプログラミングを行うこととした。新規開発の場合であっても、システムを分割することにより、データ入出力、ユーザインタフェース、データベース、ネットワークなど既存のプログラムと相似の部分进行分析し、そこ

からプログラミングを始めるようにする。

スレッド、例外、イベントについても、その機能をプログラミングする必要のある受講生と、開発目標のサンプルにスレッド、例外、イベントのプログラミング例が入っていた受講生とがあった。

4 電子メール・電子ニュースによる Q&A

具体的な目標の特定、サンプルプログラムの検索の次に、実際のプログラミングを行いながら、電子メールまたは電子ニュースを使いながら、質疑応答を行う。受講生が質問し、講師が答えてもよいが、受講生どうしが教えあってもよい。

それぞれ社内機密があるため、場合によっては全く情報提供しない受講生もある。ある受講生にとっては、他の受講生が将来の顧客となる可能性があるため積極的に情報提供する場合もある。これらが混合された形で、全体として相互の情報交換を行ってきた。

5 受講生による相互発表

具体的な必要のある受講生からの説明は、その必要のない講師からの説明に比べて遥かに説得力がある。教えてみてはじめて自分が分かっていることと分かっていないことの区別がついたり、受講生自身が、他の受講生自身を教えることにより、達成感をえることもできる。

この教育方式の欠点は、受講生に新しい概念を使う顕在的な必要性がなかったり、新しい概念のプログラム例が見つからなかった場合には、その概念の伝達が困難になることである。そして、潜在的にはその概念に基づくプログラミングをする必要性が出てくる場合である。このことに対応するため、研修終了後もメーリングリストを継続運用することにより、

必要性が顕在化した時点で対応できるようにしている。メーリングリストを運営していると、新しい概念に基づくプログラミングの必要性の顕在化があるが、新しい概念に基づくプログラミング例の紹介も増えている。

6 まとめ

開発目的の具体化・詳細化、プログラミング例の検索、質疑応答による情報共有、成果発表による概念の詳細化を通じて、技術的概念の伝達を試みた。

スレッド、例外、イベントのうち、イベントが最も利用が多く比較的容易に理解が深まった。例外はイベントとの違いが明確にならない場合もあった。スレッドは、プロセス、タスク、ジョブというそれぞれの開発環境における呼び名との原理的な違いが理解が進まなかった。

実際に利用していても、サンプル通りでエラーが生じないと、その概念についてうまく理解できない場合もあった。そのため、講師から故意にエラーが出る書き換えを示唆することにより、概念の特定を促すことがあった。抽象的な技術的な概念の伝達は、レベルで講師が理解していないと伝達できない場合もある。そのため、今後は、プログラミングに必要な技術概念の体系的に行う教育カリキュラムの作成を試みたい。特に、時間概念に関するプログラミング機能が、Windows™では十分に提供されていないため、実用的な教育を行う場合に体系的な補足が必要となる場合がある。しかし、リアルタイム性を保証していないシステム上で、時間概念をうまく教育することはむづかしい面も多い。

毎年の研修の受講者との連携により、広範囲な応用ができるための教育体系の整備を行っていく予定である。