

# 情報技術科1年次 Java 言語プログラミングに関する教材の見直し

情報技術科 大蔵 将利 久保 雅俊

## 1 はじめに

1年次前期のC言語によるプログラムの授業ではプログラミングについて理解していた学生が、3QのJava言語によるプログラムの授業では、理解が進まないことが多く見られた。そのような学生は、4Q以降の授業においてもJava言語によるプログラム作成が困難となっている。

このことから、1年次3QのJava言語によるプログラムの授業でつまづきをなくすることが最も重要であり、改善を行う必要があると考えた。

1年次				2年次	
1Q	2Q	3Q	4Q	1Q	2Q
C言語	→	Java言語	C++言語 Java言語	→	→

図1 情報技術科の言語系授業

## 2 研究の目的

1年次3Qの科目「ソフトウェア設計実習I (Java言語)」における問題点を把握し、改善することで、受講したすべての学生がJava言語を用いたプログラミングを行えること、その後の科目においてもスムーズな授業展開が行えるようになることを目的とする。

## 3 現状の調査

情報技術科1年生15名、2年生10名に対してソフトウェア設計実習Iの授業について、聞き取り調査を行った。

その結果、1年生10名、2年生4名から「C言語でプログラミングを行えるが、Java言語ではうまくできない」という回答があった。その原因については、①「C言語ではどのように記述すれば動作するか解っているが、Java言語ではどのように記述すればよいか解らない」、②「クラスを作成する基準がわからず、処理をどこに記述したらよいかわからない」、③「解説の内容が難しく、理解する前に次に進んでしまう」という回答であった。

## 4 改善に向けての検討

前述した次の3点について個別に検討を行った。

### 4.1 Java言語ではどのように記述すればよいか解らない

現状の教材ではC言語とJava言語を比較できるように提示しているが、授業ではC言語については口頭で解説し、Java言語による演習のみ行っている。そこで、C言語でプログラムを作成し、その後と同じ動作を行う処理をJava言語で記述するといったように、解説のみではなく実際にプログラムを作成するような授業展開ができれば、Java言語の文法理解も進むのではないかと考えた。

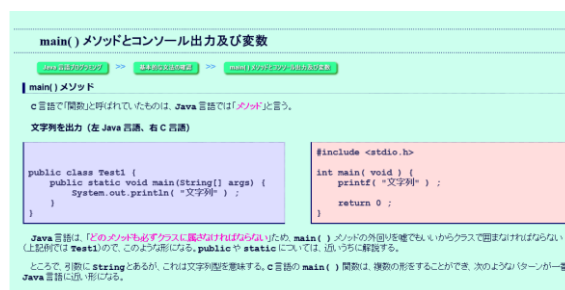


図2 C言語との比較 (現状の教材)

### 4.2 クラスの作成基準、処理の記載場所がわからない

Java言語はオブジェクト指向言語であり、C言語は構造化プログラミング言語である。構造化プログラミングからオブジェクト指向プログラミングへ考え方をを変えるには、その違いについて理解する必要がある。この点が現状の教材には不足していた。そこで、構造化プログラミングとオブジェクト指向プログラミングの違いについて解説を入れるとともに、同じ題材で両方のプログラミングを行わせ、考え方を理解させるような練習問題を充実させる必要がある。

### 4.3 解説の内容が難しく、理解する前に進む

現状の教材が作成されたのが7年前である。その頃と比較すると現在の学生は能力が低下していると考えており、比較的難易度が高い部分についての説明を飛ばして授業を行っている。そのため、説明が理解できないうちに次のステップへ進んでしまっていることが原因だと考えられる。そこで、練習問題や解説について学生が理解しやすく、後から見返して理解できるように教材を変更する必要がある。

## 5 教材に関する工夫

現状の教材はプログラムの文法や動作原理の解説後、プログラムを作成する問題に取り掛かるという構成となっている。そのため、問題ができない学生はそもそも文法や動作原理が理解できていないからできないのか、問題の内容が理解できていないためできないのかを区別することに時間がかかってしまい、一人に対応する時間が多くなってしまうことがあった。

この点を改善すべく、プログラムを作成する問題の前に文法や動作原理を確認する小問題を追加し、答え合わせの後に小問題を実際に作成して動作確認し、何が間違っていたのかを理解してもらってから、プログラムを作成する問題に取り掛かるという流れとなれば、学生自身の質問の仕方に変化が期待できる。

The image shows a page from a textbook with two code examples. The first example, titled '乱数の発生' (Random Number Generation), shows a Java class 'GenerateRandom' that prints 10 random double values. The second example, titled '判別例' (Example), shows a Java class 'Column' that reads an integer from the user and prints the number of digits. Below the code, there are two numbered items: (1) 0以上10未満の整数値の乱数を20個生成する。 (2) -10以上10未満の整数値の乱数を20個生成する。

```
public class GenerateRandom {
    public static void main(String[] args) {
        double random;
        for (int i = 0; i < 10; i++) {
            random = Math.random();
            System.out.println( random );
        }
    }
}

public class Column {
    public static void main(String[] args) throws IOException {
        InputStreamReader isr = new InputStreamReader( System.in );
        BufferedReader br = new BufferedReader( isr );
        String buffer;
        int num;
        int column;

        System.out.println( "桁数を求めます。" );
        System.out.print( "整数値を入力してください。" );
        buffer = br.readLine();
        num = Integer.parseInt( buffer );
        column = num == 0 ? 0 : (int) Math.log10( Math.abs( num ) ) + 1;
        System.out.println( num + "の桁数は、" + column + "桁です。" );
    }
}
```

(1) 0以上10未満の整数値の乱数を20個生成する。  
(2) -10以上10未満の整数値の乱数を20個生成する。

図3 解説から問題提示への流れ（現状の教材）

## 6 おわりに

今年度はソフトウェア設計実習 I の教材改定の問題点の把握と対策の検討に留まってしまったため、来年度は教材改定及び検証を行いたい。

## 7 参考文献

- (1) 河合昭男, 最新 UML がわかる, (2002), 技術評論社